

Patrick Littell

Non-violent game design patterns

VERSION 1.9

Copyright © 2024 Patrick Littell

All game titles mentioned herein are trademarks of, and all screenshots copyrighted by, their respective publishers.

Contents

<i>Introduction</i>	11
<i>How to use this book</i>	12
<i>What this book is not</i>	13
<i>Motivation</i>	14
<i>Are games really all that violent?</i>	15
<i>So why do we think non-violent games are rare?</i>	16
<i>Fitting it all together</i>	17
<i>The Unemployment Problem</i>	18
<i>Unemployment Hedge 1</i>	20
<i>Unemployment Hedge 2</i>	21

PART I WIN AND LOSS CONDITIONS

<i>Reach a destination</i>	24
<i>Area Control</i>	25
<i>Encirclement</i>	27
<i>Connections</i>	28
<i>Search and capture</i>	29
<i>Collection</i>	30
<i>Conversion</i>	32
<i>Scoring/Victory points</i>	33
<i>Constraint satisfaction/optimization</i>	34
<i>Whodunit</i>	35
<i>Chaining/following</i>	36

<i>Beat the clock</i>	37
<i>Persistence</i>	38
<i>Self-challenge and self-expression</i>	39
<i>Self-sabotage</i>	41

PART II AVOIDANCE

<i>Avoidance</i>	43
<i>Edge avoidance</i>	44
<i>Grazing</i>	45
<i>Ground avoidance</i>	46
<i>Moving safe zone</i>	48
<i>Stealth</i>	49

PART III RELATIONSHIPS AND SERVICE

<i>Relationship building</i>	52
<i>NPC preferences</i>	54
<i>Customer service</i>	56
<i>Cooking</i>	57
<i>Medicine and healing</i>	58
<i>Rescue and emergency response</i>	59
<i>Parenting and education</i>	60
<i>Matchmaking</i>	61
<i>Animal companions</i>	62
<i>Animal husbandry</i>	63
<i>Gardening</i>	64
<i>Ecosystem maintenance</i>	65
<i>Cleaning</i>	66

PART IV TIME-RELATED PATTERNS

<i>Time management</i>	68
<i>Schedule deduction</i>	70

<i>Time travel</i>	71
<i>Time loops</i>	72
<i>Day/night, weather, and seasons</i>	73
<i>Streaks</i>	74
<i>Festivals</i>	75
<i>Resting</i>	76
<i>Diaries</i>	78
<i>News</i>	79

PART V BUILDING PATTERNS

<i>Map building</i>	81
<i>Level editors</i>	83
<i>Base-building and home decoration</i>	84
<i>Showcases</i>	85
<i>Multiplayer visitation</i>	86
<i>Structural engineering</i>	87
<i>Demolition</i>	88

PART VI ART AND MUSIC

<i>Picture revelation</i>	90
<i>Photography</i>	91
<i>Playing musical instruments</i>	92
<i>Music composition and orchestration</i>	93

PART VII MAPS AND TRAVEL

<i>Breadcrumbs</i>	95
<i>Mazes</i>	96
<i>Constrained traversal</i>	98
<i>Parallel worlds</i>	99
<i>Delivery</i>	100
<i>Trading</i>	102

<i>Racing</i>	103
<i>Shortcuts</i>	104
<i>Fast travel</i>	105
<i>Shortcut worlds</i>	106

PART VIII CONTROL AND MOVEMENT

<i>Timed inputs</i>	108
<i>Combos and trick chaining</i>	109
<i>Climbing</i>	110
<i>Flying, swimming, and gliding</i>	111
<i>Digging</i>	112
<i>Vehicles</i>	113
<i>Slalom</i>	114
<i>Wall-running</i>	115
<i>Rails and tracks</i>	116
<i>Trampolines and launchers</i>	117
<i>High inertia/low friction</i>	118
<i>Fumblecore</i>	120
<i>Consensus controls</i>	121
<i>Simultaneous control</i>	122
<i>Blob movement</i>	123
<i>Character abilities</i>	124

PART IX INDIRECT CONTROL

<i>Escort</i>	127
<i>Lemmings</i>	128
<i>Containment</i>	130
<i>Minions</i>	131
<i>Automation</i>	133
<i>Chain reactions</i>	134

<i>Ballistics</i>	135
<i>Ricochet</i>	136
<i>Gravity manipulation</i>	138

PART X POWER-UPS AND STATUS EFFECTS

<i>Power-ups</i>	141
<i>Ability borrowing</i>	143
<i>Variant forms</i>	144
<i>Status effects</i>	146
<i>Stunning, freezing, and lost turns</i>	147
<i>Knockback</i>	148
<i>Pushing and throwing</i>	149
<i>Confusion</i>	150

PART XI SEARCH AND GATHERING

<i>Gathering/foraging</i>	152
<i>Set collection</i>	154
<i>Hidden objects</i>	155
<i>Scanning</i>	157
<i>Radar and triangulation</i>	158
<i>Treasure maps</i>	159
<i>Location deduction</i>	160
<i>Tracking and identification</i>	161

PART XII CRAFTING AND ITEM PATTERNS

<i>Crafting</i>	163
<i>Trait propagation</i>	164
<i>First-time item bonuses</i>	165
<i>Supply-chain management</i>	166
<i>Inventory Tetris</i>	167
<i>Shops</i>	168

<i>Haggling</i>	169
<i>Provisioning</i>	170
<i>Theft and hacking</i>	172
<i>Contraband</i>	173
<i>Collectable game components</i>	174

PART XIII EQUIPMENT

<i>Equipment</i>	176
<i>Pseudo-equipment</i>	177
<i>Equipment-based abilities</i>	178
<i>Equipment sets</i>	179
<i>Cosmetic items and dress-up</i>	180
<i>Disguises</i>	181

PART XIV PUZZLES

<i>Lock and key puzzles</i>	183
<i>Permutation puzzles</i>	185
<i>Permutable levels</i>	186
<i>Ordering puzzles</i>	188
<i>Tiling puzzles</i>	190
<i>Arrangement, packing, and dissection</i>	192
<i>Box pushing</i>	193
<i>Sorting and organization</i>	194
<i>Matching puzzles</i>	196

PART XV DIALOGUE PATTERNS AND LANGUAGE PUZZLES

<i>Dialogue trees</i>	199
<i>NPC chatter</i>	200
<i>Topic-based dialogue</i>	202
<i>Language learning</i>	204
<i>Spellcrafting</i>	206

<i>Queries</i>	208
<i>Riddles</i>	209
<i>Knights and knaves</i>	210
<i>Word puzzles and other wordplay</i>	211

PART XVI HIDDEN INFORMATION AND SPECIAL RULES

<i>Hidden rules</i>	213
<i>Hidden goals and roles</i>	215
<i>Fog of war</i>	216
<i>Hidden passages</i>	217
<i>Clairvoyance</i>	218
<i>Light and darkness</i>	219
<i>Memorization</i>	220
<i>Unidentified items</i>	222
<i>Religion</i>	224
<i>Rule modification</i>	226

PART XVII PROGRESSION AND UPGRADES

<i>Map expansion</i>	228
<i>Expanding hubs</i>	230
<i>Leveling up</i>	232
<i>Leveling by use</i>	234
<i>Research</i>	236
<i>Money</i>	237
<i>Class systems</i>	239
<i>Multiclassing</i>	241
<i>Upgrade boards</i>	242
<i>Upgradable items</i>	243

PART XVIII RANDOMNESS AND GAMBLING PATTERNS

<i>Ability checks</i>	245
-----------------------	-----

<i>Wagering</i>	247
<i>Press your luck</i>	248
<i>Gacha</i>	250
<i>Procedural generation</i>	251
<i>Rando</i>	252

PART XIX REPLAY AND METAGAME PATTERNS

<i>Challenge collectables</i>	255
<i>Second quests</i>	256
<i>NewGame+</i>	257
<i>Prestige/Ascension</i>	258
<i>Meta-progression</i>	259
<i>Leaderboards and histograms</i>	260
<i>Timelapses and highlight reels</i>	261
<i>Alternate endings</i>	262

Introduction

With the growing popularity of non-violent (or at least *less*-violent) videogames, I've noticed game designers increasingly considering the possibility of just leaving combat out of their game... and then, on doing so, hitting a bit of a wall. "I'm worried that my game will seem shallow without combat." "Playtesters say there's not enough to do." "Is it even possible to make a fun game without combat?"

Of course it's *possible*; a huge number of games have no violence at all, especially if you look at marketplaces and venues beyond Steam and consoles. Even in violent games, you do a lot of non-violent activities; sometimes the non-combat portions of a game are the most memorable parts (looking at you, *Shenmue* and *Yakuza* games).

But I still agree it's a challenge, with a genuine risk of a feeling of "shallowness", of the player just doing a grab bag of activities the designer added in lieu of punching goblins.

I think this is a really interesting design challenge, and I started compiling the various activities people do in games. (Granted, many of these, in their source games, *do* support a punching-goblins game loop – I'm not limiting myself to looking at non-violent games here – but there's nothing goblin-specific about them, we could just as easily slot them into a game about chefs or spies or walruses playing alpenhorns.)

I especially took note of some repeating patterns, and how they fed back into other mechanics, crafted particular emotional experiences, or expanded the game's cognitive variety. This book is a compilation of these patterns in one place.



Much of this recent interest is probably due to the success of *Stardew Valley*; it's very frequently mentioned in online discussions of nonviolent game design. While it's not the first game in its genre (it's a spiritual successor to the *Harvest Moon* games), and does include some light dungeon-crawling, the game is very well put-together and I think has convinced many game designers that they can execute their vision without putting combat at the center.



Undertale is likewise inspiring many designers to consider the possibilities of non-violent gameplay.

How to use this book

This book is still a messy grab-bag of ideas, and my hope is that a designer of a non-violent game, upon hitting a wall, might reach in and find a handful of ideas that deepen their game.

Some of the patterns are very abstract like REACH A DESTINATION (p. 24), which forms part of the win conditions of a vast variety of games. Others like SPELLCRAFTING (p. 206) or ANIMAL HUSBANDRY (p. 63) appear only in a handful of games, but I highlight them because I think they might not occur to you as a possible ingredient for your game. Not everything here is a “game mechanic”; some of them are aspects of theming more than mechanics themselves.

However, I want to emphasize that merely *adding more activities* does not necessarily keep a game from feeling shallow. It can even contribute to it: “Why am I doing all this random stuff?” I advise designers to read the section FITTING IT ALL TOGETHER (p. 17), where I try to diagnose what I think is a common problem causing non-violent adventures to feel anemic or purposeless.

Part of the theme of this book is that you *don't* have to pad out your game with more fetch quests or a fishing minigame. You don't even have to add arbitrary “puzzles”; designing good puzzles is a bit of an art and many people aren't great at it, and that's fine.

There are *hundreds* of activities players do in games, and there are almost certainly some that resonate with your game's themes/characters/story. That's why having a catalogue of them is useful. You don't have to reach for the obvious filler activities; over 50 years of game design we've come up with a lot of alternatives.

What this book is not

This book will not provide any sort of Unified Theory of Nonviolent Gaming, laying out any sort of abstract structure that all such games share or need to have. This would actually undermine one of my broader goals, to point out that human play is extraordinarily diverse in structure and content. Even just limiting it to videogames, *Tetris*, *The Sims*, *SpaceChem*, *Diner Dash*, *Hidden Folks*, *Euro Truck Simulator*, *Tokimeki Memorial*, and *Among Us* offer experiences that are barely commensurable, except at a very basic cognitive/neurological level that I think is shared with many other kinds of “play” from knitting to banter to playing musical instruments.

This is also not a tutorial or beginners’ guide. I hope I never sound like I’m talking down to the reader as if they’ve never heard of leveling or victory points or collecting stuff. I assume you know basically everything in this book already, and my goal is only anamnesis, helping you unforget them. This book has about 175 chapters and... I’m not going to count it but probably 500+ example games, and there’s no way you can keep that number of things in your mind. The theme of the book is really that you *already* have everything you need to make a nonviolent game, but you might need to be reminded of it, or play some example games to see familiar ideas in a new light.

In the sections that follow (especially pp. 17-18 and 23) where it looks like I might be offering a theory-of-games or pointing-out-the-obvious, it’s just because I’ve encountered many designers who express similar concerns and whose games had similar problems, and I’m offering my preferred prescription for that problem. It’s not a prescription on *how to make a game*, any more than a doctor’s prescription of a particular medicine is necessary for human life in general.

The language I know second-best, after English, has no word for “game”. There is a word for “play” with a similar semantic range as the English word, and then there’s a nominalization of that, roughly “plaything”, that describes everything from chess to a ball to a doll.

Motivation

Why am I writing this? Well, I only design non-violent games, myself, so it's something I think about quite a bit. Every time I'm playing a game, there's a question whirring in the back of my mind, "Would this game actually work with the combat removed or replaced?"

Don't get me wrong, I don't think violent fantasies are *wrong* or cause us to be violent! Jones (2003) makes a compelling argument about the ubiquity and value of violent fantasy in childhood make-believe, and what role it plays in children's emotional development into confident adults. If we lived in an opposite, puritanical world where *only* nonviolent games existed and children didn't have the freedom to *pew-pew-pew*, I might be writing the exact opposite book. But we don't live in that world, and combat is basically the autopilot of videogame design.

Your mental prototype of "videogame" – not every game, but the prototype, the thing that probably comes to mind when you see the word "videogame" without any context – probably involves a core mechanic where a player is making "attacks" against an enemy that decrement some sort of counter ("HP") and when this counter reaches zero, the enemy is removed from play; meanwhile the enemy is trying to do the same to the player.

Beyond any philosophical concerns, I'm simply getting really *bored* of that. In the astonishing variety of human play across the world and through history, somehow about half of our videogames use this exact same mechanic, endlessly rehashing minor variations to this. And I'm clearly not the only person on this wavelength; "How do I make my game fun without combat?" is one of the most frequent questions on game design forums. People *want* to make games exploring *other* game mechanics, representing other aspects of the human condition. If that's you, this book is for you; I want to play your games, and I don't want them to be boring.

Actually, it's really more honest to say that I don't design games at all. I made a few hobby videogames and set a lot of puzzles for puzzle competitions, and then my career went in a different direction. But I still care about non-violent game design, and especially for designers who are trying to make the games that I never did.

Gerard Jones. 2003. *Killing Monsters: Why Children Need Fantasy, Super Heroes, and Make-Believe Violence*. New York: Basic Books.

Are games really all that violent?

Whenever violence in videogames comes up in discussion, there's a sort of unspoken assumption that there's a lot of it. And there is... but also the gaming entertainment industry considered broadly puts out an absolute *ton* of games every year where the question of violence never even comes up. Just imagine if you were suddenly required to straight-up murder a dude halfway through playing *Madden*, video slots, *Windows Solitaire*, or your mother's favorite hidden object game.

These aren't tiny niches for eccentric hobbyists like me, either. Gamers and game designers often poo-poo these parts of the industry for being full of recycled ideas and shameless cash grabs... but the ideas are recycled because they work, and you can't grab for cash without a lot of people holding up cash.

I tried to do some (very unscientific) quantification of this question, looking at a variety of sources of game titles (publisher websites, Wikipedia, Home of the Underdogs, etc.), and my rough conclusion was about half of games published were nonviolent – not even cartoonish enemy-squishing like *Mario* or the side-activity combat of *Stardew Valley*, but without any hint of violence whatsoever. I went to the homepage of a major publisher (EA) and it happened that *none* of the games featured on the front page that day featured violence. I was honestly quite surprised.

You may try the same and get different numbers, but it's not like violent games are 95% of games, the way you might expect from online discourse about it.



Videogame stores are full of sports titles like *FIFA 23*. “But I don’t really care about those games,” you say. To be honest, nor do I, but their existence is absolutely relevant to the question of whether games are violent overall.



www.homeoftheunderdogs.com is an invaluable resource for this sort of thing, despite the age of the games featured, because it's quite neutral about what kinds of games are worthy of inclusion. Most gaming websites are interested only in particular kinds of games, and consider other games as beneath notice, but the Underdogs sees all.

So why do we think non-violent games are rare?

When we “hardcore” gamers think about the violent/nonviolence continuum, we somewhat myopically concentrate on game genres where that distinction is *live*. Consider this conversation:

A Hey I picked up a new action-adventure game this weekend!

B Do I kill anything in the game?

That’s a reasonable question; the answer is probably yes, but it really could go either way. But now consider:

A Hey I picked up *PGA Tour 2K23* this weekend.

B Do I kill anything in the game?

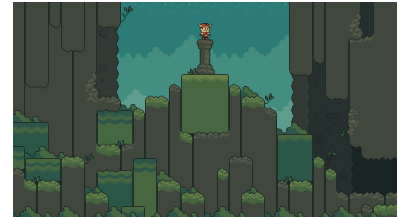
A Dude...

Golf games are probably not the first thing that comes to mind when you think “non-violent videogame”, because there’s not a salient genre of combat-golf games to contrast them with. You would never say “It’s the latest in EA’s line of non-violent golf simulators,” or write a thinkpiece on *Rock Paper Shotgun* with the line “*PGA Tour 2K23* offers a refreshing alternative vision for what a game can be.”

Instead, “non-violent videogame” makes us think of walking simulators like *Dear Esther*, farming games like *Harvest Moon*, or puzzle-platformers without enemies like *FEZ* – games that are broadly similar to violent games in style and compete against them in the marketplace, but have the violent aspects stripped out or replaced by something else. These *are* somewhat rare, although they’re becoming a lot more common.

It’s a pretty limited slice of the gaming-entertainment industry... but it *is* what a lot of game designers are curious about right now, and it’s this kind of game that this book is geared towards.

I feel like I’m committing the same sin of circularity that I’m arguing against in this section – I’m largely going to ignore gambling and “mom games” and even sports titles, and focus myopically on Steam-and-console style games. But that’s what I play! Probably that’s what you play too!



Am I going to kill anything in this game? Could go either way...



Am I going to kill anything in this game? Probably not...

Fitting it all together

This book contains a long list of ingredients, but just throwing together ingredients doesn't necessarily result in a cake. That's really the problem I see designers worried about – not that there's not enough for the player to *do*, but that for some reason the game as a whole still feels shallow or disconnected. If your game feels purposeless, just adding more fetch-quests and a fishing minigame probably isn't going to fix anything.

So let's back up and look what role combat plays in adventure genres like RPGs. Not just as a source of visceral excitement, but the way that combat forms the protagonist's identity and their main connection to their world. In the prototypical combat-based game, the player is often playing a blank-slate sort of "adventurer", sort of an unemployed drifter who's good with a sword. But drifter or not, the protagonist is not really *unemployed* – they're a monster hunter, which is a crucial economic and social role in a world that's strewn with dangerous monsters. They may not collect a formal paycheck, but monster hunting is their profession.

1. The protagonist has skills, and opportunities to practice them. Over time they learn new skills, accumulating a versatile toolbox with which to solve problems.
2. The world offers rewards for using these skills (money, loot, experience points, etc.), which can be spent/equipped/applied to strengthen the protagonist and improve and diversify their skills.
3. Thematically, the protagonist's skills are esteemed by NPCs and treated as valuable.¹ The world is shown to change in some way due to the protagonist using their skills.

This forms a solid mechanical backbone and core fantasy. Even if there were no overarching story, these ingredients let the player author their own mini-stories of competence, overcoming adversity, and feeling valued.

But none of the above three points mention combat itself. Any sort of skillset could work here: blacksmith, doctor, chef, thief, journalist, paleontologist, breakdancer, etc.



Combat systems on their own are not always very exciting, considered in isolation. Nonetheless, people still play and love these games. Combat is rewarding in part because of a game structure, and world theme, in which combat is rewarded.

¹ Or at least, *some* NPCs esteem the protagonist's skills. The protagonist might be an assassin or something that the world despises, but at least the Assassin's Guild thinks these skills are important.

The Unemployment Problem

The problem that I see designers falling into, though, is that they start with the DNA of a combat adventure, remove the combat, and leave that hole empty. Not the hole of *things to do* – many of these games have lots to do – but the hole of how the protagonist and world interact and grow together, how the protagonist gives to their world and the way the world gives back. The protagonist becomes an *actually* unemployed drifter, and I feel that’s game design hard-mode. It’s hard to signal to the player how the various activities fit together, in the absence of some sort of skillset that the player is seeking to improve, and the gameworld needs and acknowledges.

Consider *Stardew Valley*. While you do a complete smörgåsbord of activities in the game, ultimately almost everything ties back mechanically into the growth of the farm, and thematically to the protagonist’s role as a small-town farmer.

Compare this to *Yonder: The Cloud-Catcher Chronicles*, which I think doesn’t hold together as coherently as *Stardew Valley* does. The protagonist is a somewhat generic adventurer, basically Link without a sword. His relationship with the world is... I’m not sure, beyond the fact that presumably he’s going to save it somehow. The hero being a “chosen one” is lore, though, not a mechanical or thematic justification for the smörgåsbord of activities you actually do.

Why is the protagonist quarrying stone and dying wool in *Stardew Valley*? Well, presumably at some point this is going to feed back into my farm, even though I might not know exactly *how*. “Farmer” provides a ready-made mental model for the player to hang all sorts of assumptions on; the player is a co-conspirator in making this game hold together.

Why is the protagonist quarrying stone and dyeing wool in *Yonder*? I’m not actually sure; I’m sure it’ll eventually feed back into winning the game somehow, but I don’t have a good mental model of how. I have to fall back on my understanding of videogames: that if I’m asked to do something, it’s probably necessary for the completion of the game. This feels more shallow to me because I can see the designer working behind the curtain. What am I working towards?



Until writing this, I had actually forgotten that the central goal of *Yonder* is to clear the world of poisonous “murk”. This is a pretty good overarching goal for a non-combat game, as it naturally ties into MAP EXPANSION (p. 228) (and I suppose AREA CONTROL, p. 25).

However, there isn’t a very direct mechanical or thematic connection between your in-game activities and murk removal, making it feel (to me at least) that the real goal of the game is satisfying a checklist for the designer.

Satisfying that designer, I guess?

Let me clarify, I enjoyed my time playing *Yonder* and I think it's an important document in non-violent game design; "Let's make *Zelda* without combat" is a hard problem and the designers' boldness should be congratulated. It's just that my takeaway from this experiment is that the resulting game is less successful than the easier design experiment that *Stardew Valley* represents.

So my first questions to a concerned non-violent designer is always, "What is your protagonist's role in their community?" or even just "What are they good at?" If the answer is, "They're sort of a hero, I guess," or "They're good at helping people," then the feeling of aimlessness or shallowness is probably a symptom of this deeper problem.

The fetch-quest problem

Speaking of "helping people", it's perhaps good to take a moment and address the bane of many players, excessive fetch-quests.

I don't actually think there's something *mechanically* wrong with fetch-quests. A lot of what you do in games can be boiled down to "Go somewhere, do a thing, return for a reward." I think the main problem with fetch quests is actually *theming* – does the request *reinforce* that the player's skills and efforts are valued in the world, or do they *undermine* it?

A request that ties into my character's actual skillset feels good. If I'm role-playing a strapping young warrior, when someone asks "Please save my little sister from the goblin king," that's an appropriate and respectful request, and shows that the time I put into improving my goblin-punching was valued by the community.

But when someone asks of the same character, "Please bring me a chocolate cake," it doesn't matter how politely they ask, it feels slightly disrespectful. You only live two blocks from the bakery! It suggests the character's value to the community is so low that they should be running odd jobs that everyone else can't be arsed to do. You can get away with some of this, it's not wrong in and of itself, but if there are enough of them it can add up to a game where it feels like I'm everyone's submissive doormat.

But now, flip this situation. What if my character is a baker? "Please bring me a chocolate cake!" is a completely appropriate and respectful request to make of a baker. It's an opportunity to show off their hard-won mastery of their field.

Unemployment Hedge 1

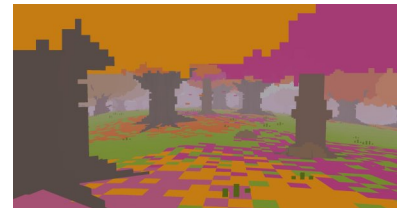
I'm not saying you *can't* make a good adventure out of a smörgåsbord of unrelated activities; I think *A Short Hike* is a masterpiece. I don't even think you need *activities*, I think *Proteus* is a masterpiece too. But as a piece of practical commercial advice, I think we too quickly point to games like this as counterexamples without acknowledging some important caveats:

- With a play length of about 2 hours, *A Short Hike* is very short game. *Journey* is a 2-hour game; *Proteus* is a 1-hour game.
- And their art design and music are great; they would support a compelling few hours of entertainment even with no game whatsoever. It's important to remember that minimalist games as vehicles for content delivery live or die based on the quality of that content. You need to be realistic about how much your non-gameplay content is a reward on its own, and how long of a game that can support. Are you at *Journey* levels of art and music, or at "my first pixel platformer" levels?
- People like to point to *Dear Esther* (2012), *Gone Home* (2013), and similar as examples of how you don't need a lot of gameplay to be a commercial success, but to some extent that's survivorship bias, and it's concerning to me that everyone's favorite examples were released in a five-year period, 2012–2017. Playing my first walking simulators around 2012 was a revelation, but by 2017, I was burnt out on wandering yet another forest path or empty village. I feel that a decade later, audiences are expecting more. *A Short Hike* or *Sky: Children of the Light* are chill and cozy exploration games but they're not mere "walking simulators" in the sense of *Dear Esther*.

Anyway, my argument isn't that you can't succeed making a game about a drifting nobody, with an unclear relationship to the world, who does a hodgepodge of activities. I'm just saying it's hard, I think few games do it well, and the games that I think do it best tend to be short and hit it out of the park on all other aspects.



A Short Hike



Proteus didn't start out as a deliberately minimalist walking simulator, it's just that the designer realized the musical experience was so compelling that he stripped out traditional gameplay elements like equipment and enemies, so as not to distract from it.

Unemployment Hedge 2

The examples I gave were all themed as a “profession” or “employment” but the protagonist↔world relationship need not be a real-world job or themed as being part of a formal economy.

The protagonists and worlds of the Pokémon games are a good illustration of this. The protagonists are children and I hesitate to call Pokémon Trainer a *job*, but the Pokémon are a sort of non-monetary economy of their own, and the role of Pokémon Trainer fills a very profession-like role in the protagonist’s life. Everything in the game feeds back to their growing ability as a Pokémon Trainer, and using these abilities is the main way the protagonist affects their world.

One important thing to observe is that the game takes a frankly rather weird collection of in-game activities and institutionalizes it by giving it a *name*; Ash and company aren’t described as kids who catch and fight Pokémon, but *Pokémon Trainers*.

The whole world accepts that “Pokémon Trainer” is a valid and valued thing to be, and there are worldwide networks of institutions like “gyms” and hospitals that support this role and provide progress MacGuffins. Institutions are also natural generators of plot and conflict: they allow for the creation of institutional factions with conflicting philosophies, and rivals who do things wrongly or for the wrong reasons.

Naming the activities *as if* they’re a profession is a shrewd way to tie together a variety of activities into a coherent whole, providing the player with a mental model for why the protagonist is doing the various things they do. Why does Ash care about getting badges? Because badges are important to Pokémon Trainers and you need them to be a Pokémon Master, that’s why.

This is circular, of course, but it’s not like an uncooperative NPC is going to blow the lid on it. You’re the god of this world and control every NPC; you can put together an airtight conspiracy to pretend that this is a Real Thing that the whole world cares about.



Pokémon Diamond/Pearl



Pokémon Red/Blue

It should go without saying, but if you’re doing this, you need to *commit* to your premise wholly. Beginner writers sometimes try to hedge their commitment by having characters break the fourth wall to comment disparagingly about the work itself, as if the writer is trying to say, “I know how stupid this is, don’t hate me for it, I’m actually cool.”

Fourth-wall breaking is a valid tool in one’s writing toolbox, but it only works when the other three walls are really solid. You can’t use it as a get-out-of-jail-free card to excuse work that’s otherwise sloppy and lacking in commitment.

Part I

Win and loss conditions

Introduction

In the last few chapters, we talked about how removing combat from a game can also accidentally remove the protagonist↔world connection that forms the backbone of many games. Another accidental consequence of removing combat can be removing a frequent and obvious win condition: “Eliminate the enemies”. Games throw regular enemy encounters at the player in part because it’s a *frequent opportunity for winning*, and people like winning.

We should be conscious of the hole left behind, and conscious about what we’re filling it with, and how often. *Must* you fill this hole? Again, I have some favorite games that don’t, like *Proteus*. But don’t leave the hole empty *accidentally* because you forgot to think about it, or replaced it with a kind of win that doesn’t happen frequently enough to keep the player engaged.

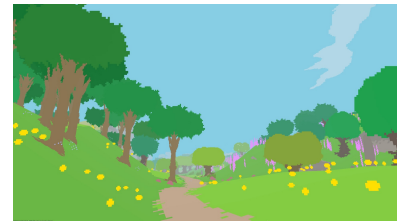
Luckily, games supply an enormous number of win conditions beyond “Eliminate all enemies” – that’s not even a particularly *common* one when we think of games more broadly. What sports have a win condition of “Eliminate all opposing players”? Dodgeball? Freeze tag? It’s not exactly common. Likewise, some boardgames have removing all opposing pieces as their primary win condition – Checkers, for example – but it’s not exactly a *common* one. When enemy elimination is present at all, it’s often a means to an end rather than an end in itself.

Does the player know the win condition?

It’s also important to think about whether the player actually knows their current win condition. If they don’t know what it is, they can’t look forward to winning.

Combat games often don’t telegraph the win condition, in part because you can pretty much assume “eliminate the enemies” will be a winning strategy unless directed otherwise. (Even if you *didn’t* know this, it quickly becomes clear, because if you don’t, the enemies eliminate *you*. That’s clearly not winning!)

But say you’re making an exploration game, and you have it so the player is rewarded with 1000 EXP when they first happen across the Forest of Zorgost. I had no particular jones to find the Forest of Zorgost, and rewarding me for something I couldn’t anticipate doesn’t set up a strong dopamine cycle. I mean, go ahead and give me 1000 EXP, rewarding the player is still valuable; just don’t count that as your *win condition*.



Winning isn’t everything; games (considered broadly) can stimulate us cognitively and emotionally in lots of ways beyond just offering opportunities for winning.

Even so, I really encourage you to think hard about how the player wins and how often. If your game feels pointless, one of my first questions is going to be “What’s your win condition?”

Reach a destination

The player has to bring themselves or an object to a particular position in the level, board, or world.

Examples

This is the win condition of an enormous variety of games. In many platformers, you're often just trying to reach the end of the level. In racing games, it's the finish line. In baseball, it's home plate; in American football, it's the end zone. In *Snakes and Ladders*, you move your piece to a goal space; in *Backgammon* or *Chinese Checkers* you need to move all your pieces into a goal area.

You can also see games like *Lemmings* as reach-a-destination games, even though the thing reaching the destination is not the player's avatar. Or even hitting-a-target games like basketball or archery; there common factor is that there's *something* that the player exerts some control over, and the goal is to get it to a certain place.

Even many videogames that you might think are about defeating enemies are really about reaching destinations; the enemies are just obstacles. Watch speedrunners go through a game avoiding 99% of encounters, and it becomes clear that the primary goal of many (non-boss) scenarios is reaching a destination.

Analysis

The widespread use of reach-a-destination in games is particularly interesting from the point of view of non-violent game design because of the way it provides an alternative win condition in an action or strategy game beyond just "eliminate all the enemies".

Even strategy games like *Final Fantasy Tactics* (1997) and *XCOM: Enemy Unknown* (2012) switch things up in story missions by challenging you to get to particular squares. Similarly, the "geo panel" system of the *Disgaea* series, combined with characters' abilities to pick up and THROW (p. 149) each other, leads to strategies sometimes centered around movement more than directly attacking.



Super Mario Bros.



In videogames, the natural blurring of identity means that this pattern is very general. In *Lemmings*, the player might momentarily identify with a particular lemming, and other times with none at all. In *Pinball*, the player might at some points identify with the paddle, or with the ball. So we don't necessarily need to draw a hard line between reaching-a-destination and hitting-a-target.

Area Control

The player attempts to establish control over portions of a map by encircling them, placing pawns, merely touching them, etc.

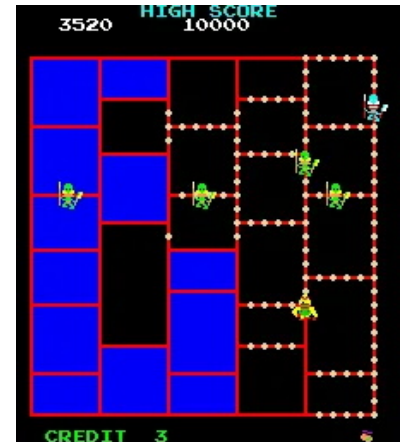
Examples

This is usually a term used for boardgames; there are so many examples I couldn't list them, from classics like *Go* to modern "Eurogames" like *Thurn und Taxis*. Instead of giving an in-depth analysis of how area control plays out in boardgames (which someone else can do better than I), I'll concentrate here on some interesting implementations that you may never have thought of as "area control".

Area control was actually quite prevalent in the 1980s; many of the "maze games" of the era aren't mazes in the sense of MAZES (p. 96) but area control games in which you have to visit every corner of the map to win. *Pac-Man* (1980), for example, is themed as a collectathon in which you are gathering pellets, but since pellets exist on every visitable square of the maze, the real goal is to visit every square at least once. This is even clearer in similar games like *Amidar* (1981) where having visited every square around a block changes that block's color (cf. ENCIRCLEMENT, p. 27).

Some painting and graffiti games like *Jet Set Radio* (2000) and *Splatoon* (2015) take the format of shooter games but with area-control win conditions (e.g. paint some portion of the arena, often against opponents who are trying to do the same). In a mechanically similar but thematically opposite direction, there are "clean-em-up" (p. 66) games like *PowerWash Simulator* (2022) where the player is shooting water and removing grime/paint/goop from the level. These also shade into firefighting games (cf. RESCUE AND EMERGENCY RESPONSE, p. 59), which tend to combine area control mechanics (against the fire) with SEARCH (p. 29) mechanics (for survivors).

When there is only a single area/point to control, and players/enemies are contesting it, area control is sometimes called "King of the Hill", after the children's activity of the same name.



By coloring in rectangles that you've fully circumnavigated, games like *Amidar* make clearer the underlying area-control mechanic of 1980s maze games.



Splatoon

Analysis

As mentioned above, area control offers an alternative win condition beyond “defeat the enemies”. This was extremely common in the arcade era; in the early 1980s publishers were still a bit squeamish about killing (at least of humanoid opponents). Even in 2d side-on platformer-like games like *Wrecking Crew* (1984), *City Connection* (1985), and *Rampage* (1986), the enemies are primarily obstacles keeping you from visiting all the map elements you need to progress.

Another interesting aspect is the way area control can provide a visual indicator of player progress. In *Super Mario Bros.* and the like, it’s obvious how far you’ve come, because you’re just running rightward; however far you’ve run is how far into the level/game you are. But in a more free roaming game, it’s less clear. An area control mechanic lets the player see at a glance their past successes: “Look at how much of the map is blue now!”

Whether or not your progress is ratcheted (i.e., can you ever lose progress by an opponent capturing that area?) greatly influences how area control feels. Ratcheted area control feels particularly reassuring to me and tweaks the “completionist” part of my brain, especially in single-player games where I know I’m slowly working towards 100%. On the other hand, the threat of losing an area is a way to engender anxiety; loss aversion is a very deep-seated human cognitive bias!



In *City Connection* (pictured here in the 1988 NES version), the goal isn’t to defeat the other cars, just to travel over every platform segment.

Encirclement

The player controls a region by surrounding it on all sides.

Examples

This is a sort of AREA CONTROL (p. 25) and many such games make use of it. What counts as encirclement varies – having a piece on every space around a region (as in *Go*), just having pieces on either end of a piece or line (sometimes called “custodial capture” and seen in *Tafl*, *Reversi/Othello*, and *Ninuki-Renju/Pente*), or even just visiting every space as in *Amidar* (1981).

A notable early arcade game based on encirclement was *QIX* (1981), in which the player draws lines behind them and progressively shaves off regions of an arena. *JezzBall* (1994) was a popular early Windows-era game along similar lines.

Zen Bound is an rare 3D encirclement (enspherement?) game in which you’re trying to wrap irregular figures with a length of rope.

Analysis

Another key question for encirclement games is “What do you do with the captured area/pieces?” In *Go* and *QIX*, the enemy pieces are removed. In *Othello*, however, they’re converted to your own color. *Line Knight Fortix* (2014) is a *QIX*-like game on a medieval battlefield, and encircling units like catapults converts them to your side.

Many *QIX* descendants involve PICTURE REVELATION (p. 90), reinforcing the basic gameplay with the player’s curiosity about what the picture might be. There’s a long lineage of erotica games à la *Gals Panic* (1990) where slicing off areas gradually reveals pinup models (or today, anime waifus).

Sometimes more complex things appear. *Bunmei Korokoro Game: Egg* (1998) tasks the player with rolling a gigantic egg around a field, mini-golf style, and when its path encloses an area, civilization develops inside. *Urbanix* (2010) is a more conventional *QIX*-clone with a city-building theme, where captured areas develop into city blocks.



QIX



Gals Panic



Bunmei Korokoro Game: Egg (*Civilization Roly-Poly Game: Egg*)

Connections

The player has to build a route between two or more things.

Analysis

In the 1940s-50s, a large family of connection games were invented, in which you need to use pieces or lines to connect opposing sides of the board while trying to prevent the opposing player from doing so. Games in this family include *Hex* (1942), *Y* (1953), *Twixt* (1957), and *Bridg-It* (late 1950s). Abstractly, many of these are special cases of the *Shannon Connection Game*.

Transit-related games often have connection as one of their primary goals – sometimes a goal in itself, sometimes to move cargo, often to make money. Railroad games like *Empire Builder* (1982), *Railroad Tycoon* (1990), *Ticket to Ride* (2004), and *Mini Metro* (2016) involve routing tracks between networks of stations.

There is also a subgenre of connection games where you need to solve a TILING PUZZLE (p. 190). The first one I can think of is *Pipeline: The Oil Game* (1988), in which players build pipelines between oil wells and tankers. *Saboteur* (2004) has the same basic mechanic but combines it with HIDDEN ROLE (p. 215) social deduction. The HACKING (p. 172) minigame in *Bioshock* (2007) is also a tiling connection game; although it's broadly similar to *Pipe Dream* (1989), its goal is connection rather than PERSISTENCE (p. 38).

Analysis

To some extent this is a variation of AREA CONTROL (p. 25); *Hex*'s closest relatives are probably area-control boardgames like *Go*. There's also an abstract connection to LEMMINGS (p. 128)-style games; we can see the solution to a *Lemmings* puzzle as constructing a lemming-traversable route from the drop point to the exit.

Even if your game isn't primarily about making connections, it's a possible component even in a more adventure-focused game. You could use it to, for example, restore a FAST TRAVEL (p. 105) network.



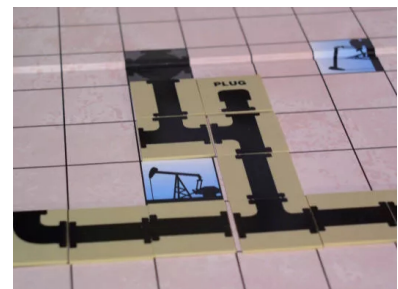
Hex, as played on *Tabletop Simulator*



In the *Empire Builder* series, like *Euro-rails* pictured here, players draw rails on the map using a dry-erase marker.



Mini Metro



Pipeline: The Oil Game

Search and capture

The player wins when they've found and/or caught one or more things. Unlike REACH A DESTINATION (p. 24), the locations of the desired things are typically not known in advance.

Examples

Finding/chasing is an evergreen win condition that shows up in many children's games, like *Hide and Seek* and *Easter Egg Hunts*. *Tag* is the classic children's game about chasing; one player is "it" and attempts to chase any other player; if they can manage to touch the other player, that player becomes "it".

This is the win condition underlying many of the more specific patterns in the SEARCH AND GATHERING section: GATHERING (p. 152), TREASURE MAPS (p. 159), etc. HIDDEN OBJECT (p. 155) puzzles and games are perennially popular, from books like *Where's Waldo* to videogames like *Hidden Folks*. It's also the primary win condition of LOCATION DEDUCTION (p. 160) games including *Hunt the Wumpus*, *Scotland Yard*, and *Where in the World is Carmen Sandiego?*, as well as most RESCUE AND EMERGENCY RESPONSE (p. 59) games.

The *Ape Escape* series is one of the best examples of an action/adventure game centered around chasing; in other ways it's a fairly standard 1990s-style "collectathon" platformer except that your collectables are excitable apes who do not want to be caught.

Chasing frogs in *Proteus* illustrates that finding and chasing can provide mild amusement even if the thing in question cannot be caught. (Well, I found it fun, at least.)



An Easter egg hunt (photo © CBC News)



Ape Escape 2

Collection

The player collects a (typically huge) number of beneficial objects (e.g. coins).

Examples

The previous section on SEARCH (p. 29) concentrated more on games where the challenge was finding special and rare things; this chapter concentrates more on the other end of the spectrum. I hesitate to categorize *Crystal Quest* (1987), *NiGHTS into DREAMS* (1996), *Katamari Damacy* (2004), or *Jetpack Joyride* (2011) as search games: hundreds of things are right there in front of you for the taking. The core of the challenge lies not in finding them – it'd be finding hay in a haystack – but something else: high speed or inertia, an unusual control scheme, a time limit, etc.

Many collectathon games of the 1990s – *Spyro the Dragon* (1998), *Banjo Kazooey* (1998), etc. – combined both ends of this spectrum: both searches for rare and important MacGuffins but also hoovering up abundant little things (e.g. gems), often laid out in BREADCRUMBS (p. 95).

An area that is unusually generous with collectables can also provide some downtime in the midst of more intense challenges, or encourage the player to press the limits of movement. Consider the level Donut Plains 1 in *Super Mario World* (1990). The player has just been given a POWER-UP (p. 141) that enables flight. Rather than give the player some substantial obstacle requiring the cape (the way *Super Metroid* would), the next section simply contains hundreds of coins in the sky. This invites the player to try out flying in a safe environment, and rewards them generously for doing so.

Analysis

Strewing around over-generous rewards is a simple tool, and I think it's oddly underutilized in non-violent gaming. After all, running



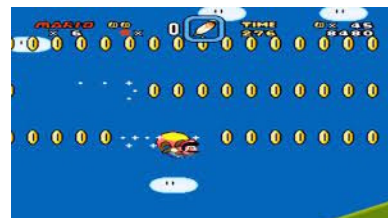
NiGHTS into Dreams



Jetpack Joyride



Spyro the Dragon



Donut Plains 1 in *Super Mario World*

around like a madman, hoovering up money, is a very basic sort of fun.

Of course, you have to be a particular kind of game for this to feel right. In lighthearted series like *Mario*, *Spyro the Dragon*, and *Ratchet & Clank*, scattering minor rewards everywhere incentivizes exuberant movement, and reinforces their fun and freewheeling gameplay and theming.

This might feel rather out of place in, say, *Castlevania* or *Dark Souls*, whose occasional and more stingy rewards reinforce those series' darker and more deliberate feel. But even if you're intending to design the latter kind of game, it's worth considering occasional areas where the player is rewarded over-generously, as an occasional shake-up that gets the player controlling the character more wildly.

Now that I'm thinking about it, actually, it's no surprise that non-violent indie games of the 2010s didn't strew coins everywhere. While they weren't *grimdark*, a lot of them tended towards a more art-house vibe. It'd be weird to position your studio as exploring games-as-art and not-games-as-games and such, and simultaneously strew golden coins around like Wario exploded.

Anyway, I think we're coming out of that era; I think the indie world is a lot less self-serious now than the mid-2010s, and I think we can consider recreating the abandon of *Donut Plains 1* without worrying that people will question our commitment to games as an art form.

Conversion

The player wins former enemies to their side.

Examples

Some traditional boardgames involve the conversion of opponents' pieces, from *Shogi* to *Reversi*.

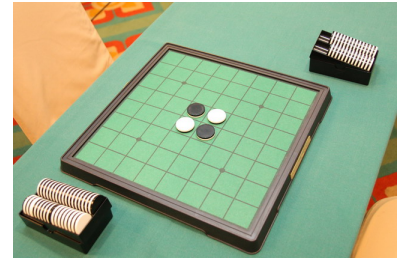
The first videogame example I can think of this is *Cosmic Soldier* (1985), an otherwise-forgettable early Japanese RPG in the vein of *Wizardry*, in which you can talk with enemies and convince them to join your side. A more influential example from the same era would be *Digital Devil Saga: Megami Tensei* (1987), ancestor of the *Persona* games. I should probably also mention the *Pokémon* series here.

In strategy games, *Civilization IV* was the first *Civ* game to give religious conversion a prominent gameplay role. The player can create "missionary" units to convert cities (including opponents' cities) to the player's religion, which gives bonuses towards the Cultural victory condition and bonuses to diplomacy with like-minded opponents.

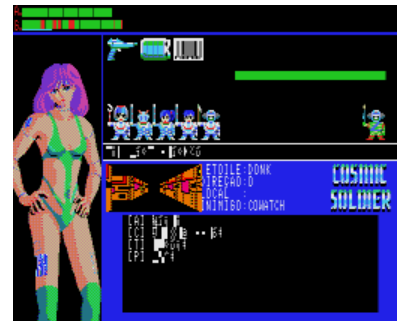
Analysis

Conversion rather than defeat is sometimes used to theme otherwise-standard-combat systems to at least seem less violent, e.g., "The bad guys were just confused and now they're friendly again."

But it's also worth stepping carefully around themes of religious conversion due to its real-world historical connection to coercion and violence. A number of reviews of *Virtual Villagers: New Believers* were uncomfortable with the central theme of converting the heathen "savages" to your religion.



Othello, the best-known modern *Reversi* game.



Cosmic Soldier



Civilization IV

Scoring/Victory points

The player is given points for various reasons, and wins when they hit a certain threshold, or wins by having more than any other player.

These points might be unthemed (e.g. simply called “score” or “points”), themed as collectables (e.g. Mario’s stars), as money (e.g. paying off a debt or being the richest player), or as relationship hearts (e.g. filling up a heart meter in a dating game).

Examples

Victory points are especially associated with German-style board-games (often called “Eurogames”) like *Settlers of Catan* or *Carcassonne*. Rather notoriously so; when people want to parody the genre they describe it along the lines of, “Turning wooden cubes into other colors of wooden cubes to get victory points.”

But some sports also work by victory points. The score in American Football isn’t a count of touchdowns; there are various things you can do to get various numbers of points.

Many “collectathon” games like *Super Mario Galaxy* (2007) effectively work by victory points. It may be themed as the player SEARCHING (p. 29) something down, but stars are effectively victory points. Some of them are found out in the world, but many are granted to you in exchange for doing things (races, scavenger hunts, timed challenges, boss fights, etc.).

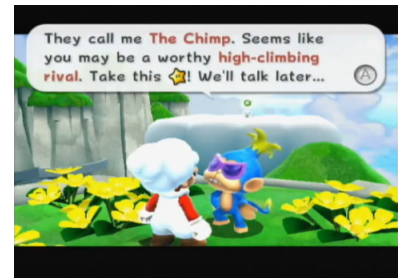
Analysis

Victory points allow the conversion of diverse activities to a common win “currency”, which can be a useful when a lot is going on and it’s unclear how it all fits together towards victory.

On the other hand, this can introduce a whiff of artificiality. This is fine in *Mario* because those games are very “gamey” and don’t pretend otherwise, but in a more seriously-themed game they might break immersion by reminding the player that it’s all a game.



The scoring track in *Carcassonne*



There are an awful lot of NPCs in *Super Mario Galaxy 2* who happen to have a star in their pockets.

Constraint satisfaction/optimization

The player is given constraints and can create/propose a wide variety of solutions (e.g. those emerging from a combinatoric system), and their solution is judged according to how well it meets those constraints.

Examples

Classic logic puzzles like *Zebra puzzles* (sometimes called *Einstein's Puzzle*) are prototypical examples. (E.g., these six people live in these six houses and own these six animals, and the player is given clues like "Neither person living next to Betty owns an elephant.") The solution space is huge, and the player's job is to narrow down their possibilities until they know who (say) owns the zebra.

But more broadly, this is a particularly abstract win condition; constraint satisfaction or optimization encompasses a wide variety of puzzle types, like **TILING PUZZLES** (p. 190) and **KNIGHTS-AND-KNAVES** (p. 210). Even though they look quite different, if you were to express these puzzles in the appropriate format, you could shove them into a constraint solver and get an answer.

Analysis

One thing to consider is whether your constraints are hard (only fully-correct solutions are accepted) or soft (the player tries to meet the constraints as best as possible, and are scored accordingly).

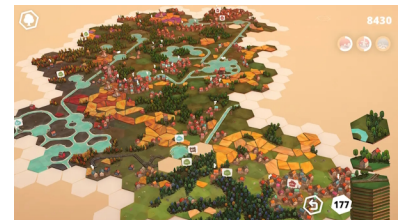
Dorfromantik, a tile-laying world-builder game, has both of these. Like in *Carcassonne*, the player's goal is generally to match tile edges – city-to-city, farm-to-farm, forest-to-forest, etc. The player is rewarded both for matching edges and making large regions of tiles connected by matches, but perfect matching is not obligatory for most kinds of edges. On the other hand, a few kinds of edge, rivers and railroads, *must* match; the player is simply not allowed to place a tile such that a river or railroad abruptly stops at a tile edge.

		LOCATION			JOB			MISSION	
		ENGINEER	INSTEAD	LANDSLIDE	ENGINEER	MANAGER	SCIENTIST	AAM	EAP
EMPLOYEE	Sonia								
	Jorge								
	Nasder								
MISSION	AAM								
	EAP								
	X-59								
JOB	Engineer								
	Manager								
	Scientist								

Zebra puzzles are often accompanied by this familiar diagram, in this case in a puzzle from NASA.gov

$$\begin{array}{r}
 \\
 \\
 + \\
 \hline
 =
 \end{array}$$

Even "Cryptarithm" puzzles like these can be solved by a general constraint solver, it's just a matter of knowing how to express these as the right kind of constraint graph. Example by Henry Dudeney in *Strand Magazine*, July 1924.



Dorfromantik

Whodunit

The player has to narrow down who someone is, typically someone who committed a crime.

Examples

This is, of course, the central win condition of most detective games, or a secondary win condition for detective games more focused on SEARCH AND CAPTURE (p. 29) like *Where in the World is Carmen Sandiego?* (1985). (Non-detective examples are more rare; I can only think of *Guess Who?*)

Sometimes the deduction is just a matter of narrowing down people based on rather direct clues (simple features and traits like blonde hair in *Carmen Sandiego*, or process of elimination in *Clue*); other times you can listen to statements (e.g. in *Her Story*), or interrogate witnesses (e.g. in *The Lost Files of Sherlock Holmes*).

Analysis

Given how enormously popular mystery/crime fiction is, it's a surprise that mystery games aren't more of a major genre in the West. They have been much more popular in Japan, however – something that many Western gamers don't know because so few of them were ever localized.

Part of this longstanding popularity is due to *The Portopia Serial Murder Case* (1983), which had an enormous influence on subsequent Japanese gaming. (The *Famicom* port is effectively the prototype for the “go on an adventure by choosing menu options” Japanese console game that has been with us ever since – the designer's next game was the venerable *Dragon Quest*.)



Clue/Cluedo



Where in the World is Carmen Sandiego?



The Portopia Serial Murder Case. It's interesting that you can't actually lose this game; if you make a false accusation, your assistant Yasu just comes back empty-handed and you continue playing. For a game of that era, that's surprisingly forgiving!

Chaining/following

The player has to respond to the previously-played piece/card/word/etc., matching it by some manner of CONSTRAINT (p. 34), and typically loses or takes a penalty if they cannot.

Examples

Many card games do this, from simple color/number matching in *Uno* to complex team trick-taking games like *Bridge*. *The Crew: The Quest for Planet Nine* is an interesting cooperative take on this.

Many languages have a parlor game in which one player calls out a word (which may have to belong to a particular category), and the next player's word must begin with the ending of that word. (E.g. GERMANY -> YEMEN -> NEPAL.) There is often a time limit for responses, like five seconds or two claps. The Japanese version of this, *Shiritori*, has a unique losing condition – there is one letter that never appears at the beginning of words, and if a player accidentally says a word ending in that letter, they lose the game because the next player could not possibly follow up.

Games in which you build on a tableau – from *Dominoes* games to *Scrabble* to TILING (p. 190) games – can be seen as a sort of chaining game; the difference is that more previous moves remain on the table and so the player has more choice in how to follow.

Aside from tableau games like the above, I've wracked my brain to think of implementations of this pattern in videogames, and the only one I can think of is Insult Swordfighting in *The Secret of Monkey Island* (1990). In this famous DIALOGUE (p. 199) puzzle, pirates block your way on multiple paths, and to get past them you must respond to their insults with clever comebacks. However, comebacks are specific to the insult, and at the beginning you have no comebacks at all. It's only by remembering those insults, and deploying them to learn the corresponding comeback, that you assemble a sufficient repertoire of matching comebacks. As you spar with (and lose to) combatants across the island, your "deck" grows until you can finally counter any insult.



Uno



Chickenfoot dominoes, photo © yoppy



The Secret of Monkey Island

Beat the clock

The player must do something before a countdown reaches zero.

Examples

The number of beat-the-clock challenges in videogames are so many that it's hard to choose particular examples; it's typically an adjunct to another win condition like REACHING A DESTINATION (p. 24) or amassing a certain number of VICTORY POINTS (p. 33).

Race the Sun (2013) has a particularly elegant diegetic timer: the setting sun. The player is racing a solar-powered craft, and so when the sun sets the game is over. Since the player is racing westward, the game has an automatic way to add bonus seconds to the timer – the faster the player goes west, the more time they have.

A temporary POWER-UP (p. 141) can also be an implicit timer. In *Wario Land II* (1998), certain blocks can only be passed if Wario is on fire. This is an implicit timer: you have to get between the fire source and the flammable block before Wario is extinguished.

Or, the “timer” might be an NPC trying to achieve the same thing before you, like the Koopa race in *Super Mario 64* (1996), or Avery's parkour races in *A Short Hike* (2019). They, too, are effectively timers: however long they take is how much time you have.

Analysis

Since these games end when you run out of time, a ticking clock is actually much like a “poison” STATUS EFFECT (p. 146) where HP keeps declining, and “time pick-ups” are effectively health potions.

Some games with ticking clocks use this parallelism to get rid of HP altogether. For example, *Treasures of the Aegean* (2021) is a TIME LOOP (p. 72) platformer where you only have a fixed amount of time to explore before the world explodes. The protagonist has no HP, but when she gets hurt, a minute is subtracted from the clock while she gets to her feet and bandages the injury.



Even looking at this timed garbage-cleanup challenge from *Super Mario Galaxy* still gives me a sense of dread.



Race the Sun



Treasures of the Aegean

Persistence

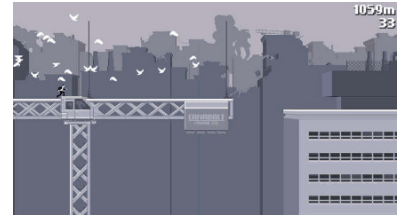
The player's goal is ultimately impossible and they simply try to survive as long as possible.

Examples

Games described as “endless”, like “Endless runner” games, usually have this as their primary goal. In an Endless Runner game like *Canabalt*, the game will eventually go too fast for you to react and you will fall or otherwise die; the goal is just to last as long as you can.

Similarly, various tower defense games like the *Bloons* series have endless modes; increasingly difficult waves of enemies will eventually breach your defenses, and the goal is to survive as many as possible.

Endless games often have increasing speed or difficulty to ensure that the game eventually ends, but entropy alone is your primary opponent in Match-3 games like *Bejeweled* (2001). As achievable matches are found and disappear, and are replaced by random replacements, there eventually will come a point where no matches are possible. Some games task you with collecting a particular number of matches, but *Bejeweled* just lets you play until you've exhausted all possibilities.



Canabalt



Bejeweled

Self-challenge and self-expression

The player is given few if any explicit goals, but has ample opportunity to invent challenges for themselves or express their creativity.

Examples

Game-like entertainments primarily built around self-challenge are often called “toys”; Maxis actually trademarked the term “software toy” to promote *SimCity* (1989). In *SimCity*, the player does not have an explicit goal, beyond avoiding bankruptcy; rather, the player is constantly setting themselves small personal challenges (e.g. laying a new rail line through an existing part of town), or responding to challenges that emerge out of the simulation due to their past actions (e.g., responding to a crime wave due to insufficient policing).

Even in a run-&-jump style adventure game, a sufficiently rich and enjoyable move-set and a complex world can support toy-like engagements. “Can I make it to the top of that thing?” “Can I make it from here to there without jumping?” Self-directed challenge, along with the advent of streaming and YouTube, has given new life to games like *Super Mario 64*. The game has become a staple of streaming and speedrunning not just because it’s a well-loved classic, but because Mario’s complex move-set lets players perform strange feats and put entertaining restrictions on their own play.

Analysis

Games differ greatly in how well they support self-challenge. Some games and toys, like LEGO building blocks, have players naturally coming up with things to build and do, to the point where players only need a gentle nudge. (In fact, giving the player too much direction, or adding too much external challenge, can undermine the player’s intrinsic motivation to explore the system.)

Other games simply don’t; there may not be enough richness, complexity, or unpredictability to support that kind of creative play.



SimCity



Many adults are still fans of *LEGO*, and there’s a rich market of designs and products for them, including this 3rd-party book on building modular urban neighborhoods.

A rich-enough toy can grow with the player, because it can adapt to changes in the player’s imagination. Something like this city excites my adult imagination the way a pirate ship did as a child.

The player will need some more direction as to how to best entertain themselves with this set of objects and rules.

Narrative richness is also important in supporting toy-like play. Dollhouses and action figures aren't all that complex or unpredictable, but they naturally suggest the kinds of narratives that children can work with – narratives from their lives or from entertainment. If you gave a child something that was *functionally* equivalent to a dollhouse, but were just abstract shapes without clear hints as to what kind of narratives they could support, it wouldn't be as rich a venue for play. *SimCity* wouldn't have worked as an "abstract strategy" game about green, blue, and yellow rectangles.

I remember, as a child, coming up with a kind of tetromino-like tessellation (cf. *TILING*, p. 190, *ARRANGEMENT*, p. 192). The strange and jagged pieces fit together cleverly, but it didn't strike anyone else's imagination, because they couldn't look at the pieces and imagine fun shapes it might build. You might be able to ship such a toy with a challenge book ("Try to make these 50 designs!"), but that would be it; upon completing those challenges, the player probably wouldn't come up with their own. Although the pieces constituted a rich combinatoric system, this system didn't allow for any interesting self-expression; it didn't tweak any *ambitions* in the player the way LEGOs or *SimCity* or dollhouses do, in their own distinct ways.

Self-sabotage

One of the player's primary actions causes danger and the primary challenge is avoiding that danger.

Examples

Related to SELF-CHALLENGE (p. 39) are games where there's a straightforward goal set by the game (often PERSISTENCE, p. 38), but most of the actual challenge arises from your own actions and their consequences.

Snake games (in which you control a snake with an ever-elongating tail, and die if you collide with yourself) are a classic example; the original snake game was *Blockade* (1976).

Asteroids (1979), too: there are few enemies or game-created obstacles hindering your path, and your death will probably be because of a situation you caused by your own over-ambition. In general, HIGH INERTIA / LOW-FRICTION (p. 118) games have an element of self-sabotage, as the impatient player finds themselves going too fast to avoid obstacles.

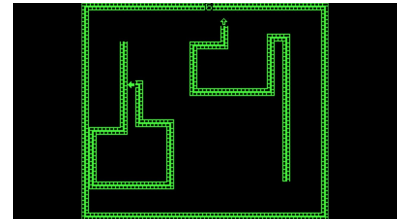
DSY: Don't Shoot Yourself (2013) is a bullet-hell AVOIDANCE (p. 43) game where the bullets to be avoided are your own ship's exhaust pellets.

In *Mr. Driller* (1999), the self-sabotage is more indirect. You are drilling down into the earth, represented by sometimes-irregular blocks, and in doing so you sometimes remove supports that lead to being crushed by falling blocks.

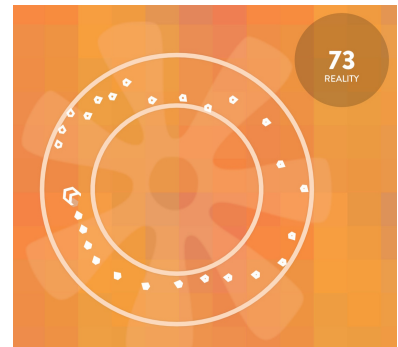
Analysis

I particularly like this pattern in the context of non-violent game design, because it gives another way to depart from player-vs.-enemies challenges, towards player-vs.-self.

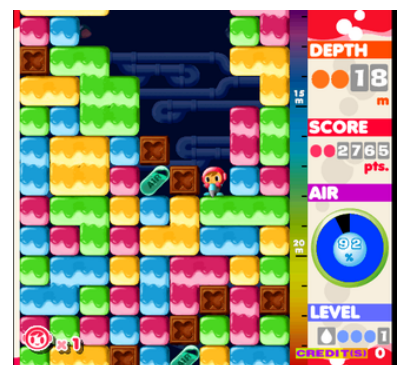
There is a relationship to PRESS-YOUR-LUCK (p. 248) mechanics here, as the player is incentivized to play in such a way now that make the game more dangerous in the near future.



Blockade was the original *Snake* game.



Don't Shoot Yourself



Mr. Driller

Part II

Avoidance

Avoidance

The character must avoid harmful moving objects (e.g. enemy missiles) by moving around in space. (As opposed to, say, pressing a “dodge” button at the right time, which I’ll cover under TIMED INPUTS, p. 108, instead.)

Examples

Players have been attempting to avoid enemy attacks since the dawn of computer games, but the development of the *danmaku* genre of shoot-’em-ups is a good illustration that avoidance can be the core game loop. Even before *Undertale*, designers have realized that the genre’s avoidance gameplay is rich enough that the damage-the-enemy part could be optional.

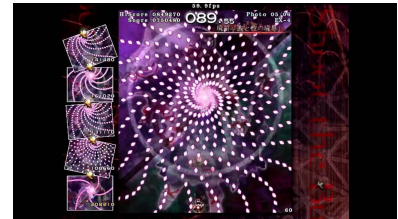
It is possible to attack enemies in *Undertale*, but it is not required; the best ending of the game requires not killing any “enemy” in the game. Rather, the player dodges enemy attacks in a wide variety of creative danmaku-like arenas until the encounter resolves by other means (such as talking it out).

While it’s not as widely known for it, it is also entirely possible to finish the more conventional danmaku game *Ikaruga* (2001) without firing a shot. Every enemy simply leaves the screen after some fixed period of time, even bosses.

The *Touhou Project* games *Shoot the Bullet* (2005) and *Double Spoiler* (2010) keep the projectile avoidance aspect of danmaku games but replace the shooting with PHOTOGRAPHY (p. 91).

Analysis

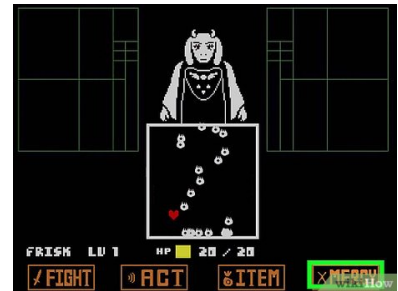
I debated putting all the avoidance patterns in the WIN/LOSS CONDITIONS section, but I feel they deserve a section of their own. Quite a few non-violent games (or games that allow “pacifist runs”) rely on avoidance as the main moment-to-moment gameplay loop.



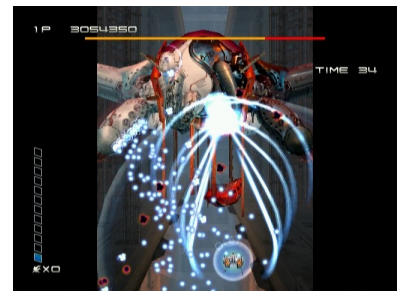
Touhou Bunkachou ~ Shoot the Bullet

DANMAKU:

“Artillery barrage”, lit: “bullet curtain”; typically rendered in English as “bullet hell”. In *danmaku* shoot-’em-up games, enemies fire huge numbers of bullets in intricate patterns, and the player is effectively navigating a shifting, deadly maze in an attempt to stay alive.



Undertale



Ikaruga

Edge avoidance

The player must stay within the borders of an area (or, in a functionally-similar expression of the idea, stay within a particular radius of a point or line).

This is, obviously, another AVOIDANCE (p. 43) pattern, but its implementation/theming are usually sufficiently different that it gets its own section.

Examples

Many early examples of stay-within-the-lines gameplay were driving games. This pattern is particularly obvious in *Wipeout* (1972) for the Magnavox Odyssey, in which the player is guiding a light through a transparent overlay on the TV screen, and losing points every time they step off the path.

The cult classic *Kuru Kuru Kururin* (2001) ups the challenge by having the player control a rotating stick. *Roundabout* (2014) brings this mechanic back to a car theme with hilarious results.

A classical electromechanical game, sometimes called *Buzz Wire* or *Wire Loop*, challenges the player to guide a metal loop along a curving wire without making contact between the two (which will complete a circuit and cause a buzz); this is functionally equivalent to a “stay within the lines” challenge. The classic children’s boardgame *Operation!* (1965) uses a similar principle. This time, rather than be forced to travel a path, the player is a surgeon extracting foreign objects from a patient using tweezers. If the tweezers hit the borders of a slot, the circuit completes and an alarm goes off.

Edge avoidance can also play a role in sports and strategic games. In *Sumo*, one win condition is to force the other player out of the ring. In the boardgame *Abalone*, a line of pieces can push a shorter line of opponent pieces, and the goal is to push your opponent’s pieces off the board. Similarly, *Auro* and *Squids!* are strategy videogames where the goal is to force enemies off the playfield.



Wipeout for the Magnavox Odyssey



Kuru Kuru Kururin



Operation!

Grazing

The player is rewarded for getting close to a collider (enemy, bullet, wall, etc.) without actually hitting it.

Examples

This mechanic is best-known in shoot-em-ups, especially “danmaku” (p. 43) games like the *Touhou* series. User *trap15* at *shmups.wiki* believes the first commercially released game that used this as an overtly was *Raiden Fighters* (1996). The earlier game *Raiden DX* (1994) also gave players “Guts” or “Fighting Spirit” bonuses for playing dangerously; how “Guts” was calculated was hidden, but behind the scenes it added a bonus for every bullet grazed.

Bonuses for near-misses are also found in racing/driving games, including *Burnout Paradise* (2008), the *Need for Speed* series, the *Grand Theft Auto* series, and the *Saints Row* series. One of the challenge modes in *Grand Theft Auto Online* challenges players to get as many near-misses as possible in 5 minutes.

In non-shmup, non-driving games, *Dino Run* (2007) was a “runner” game in which the player controls a dinosaur trying to outrun the extinction meteor’s blast radius (cf. MOVING SAFE AREA, p. 48), and the player could get bonus points for getting close to the blast radius without dying.

Analysis

Shmups typically implement this by giving the player character two hitboxes, a smaller hitbox (often quite smaller than the sprite) for the actual, negatively-rewarded collisions, and a larger “grazing” hitbox for which collisions are positively rewarded.

This obviously contributes to a risk/reward tradeoff. Some games like *Raiden Fighters* also incorporate a cumulative PRESS-YOUR-LUCK (p. 248) mechanic where each successful graze increases the amount of the reward (e.g. increases a score multiplier), but getting hit even once resets this to zero.



Raiden Fighters



Grand Theft Auto Online

Ground avoidance

The player attempts to avoid touching the ground or to keep objects from hitting the ground (“keep-it-up”).

Often, the challenge is heightened by limited control over the object (e.g. you can only touch it in certain ways), and this pattern is frequently associated with *RICOCHET* (p. 136), which provides uncertainty about the object’s trajectory and thus added challenge.

Examples

Ground avoidance is a common game pattern across cultures, from children’s make-believe games like *The Floor is Lava*, combat sports like *Capoeira* and *Sumo*, dexterity activities like *Juggling*, team sports like *Volleyball*, and electromechanical games like *Pinball*.

Often the player’s goal is to keep-it-up indefinitely (e.g. pinball), but flight simulators combine this with *REACH A DESTINATION* (p. 24): hitting the ground is considered failing, but landing in particular places is success. Flying games with a landing emphasis include *Lunar Lander* (1973), *Space Taxi* (1984), and *Pilotwings* (1990).

In *Avalanche* (1978) (more familiar to many as its Atari 2600 clone *Kaboom!*, 1981), the player tries to catch falling rocks before they hit the ground. In *Fire!* (1980) and *Bouncing Babies* (1984), the player plays as firefighters with a trampoline attempting to save people jumping from burning buildings, and bounce them to safety. The *Breakout*-like *Circus Atari* (1980) uses a similar mechanic to collect balloons, this time using a see-saw.

An interesting use of keep-it-up is in the “bell” power-up system of the *TwinBee* and *Parodius* shoot-em-up games. Power-ups are themed as bells which fall down the screen towards the player; if the player manages to shoot them, they are briefly juggled upward and change what upgrade they offer when caught. Juggling bells to maximize upgrades is crucial to high-level play in these games.



Bouncing Babies



Circus Atari



TwinBee

Analysis

The basic goal of ground avoidance is extremely intuitive, and has been invented, presumably independently, by many cultures. Unlike many objectives in videogames, it only barely needs to be explained, any more than you need to explain that it's bad to fall into a pit of spikes.

Challenge

Keep-it-up games are often chaotic action games involving RICOCHET (p. 136). Design a slower-paced, strategic game (maybe turn-based?) where keep-it-up is the primary goal.

Moving safe zone

Only part of a level is safe at any given time, and the safe area moves.

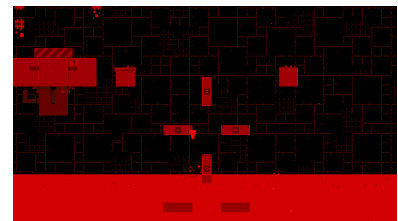
Examples

This is implemented/themed in a wide variety of ways:

- Platformer levels where rising water or lava forces the player upward, like in *FEZ* or *New Super Mario Bros.*, an “Advancing Wall of Doom”
- A wall that moves in from the left, like the blast radius of the extinction event that the player must outrun in *Dino Run*, or the darkness from the runner-JRPG hybrid *One Way Heroics*. (Or, in some games with autoscroller levels, the screen itself is the safe zone, and spending too much time offscreen results in death.)
- The periodically-contracting safe zone in “Battle Royale” games like *Player Unknown’s Battlegrounds*.
- An external force/ray/magic creates a moving safe zone, like the Matter Splatter levels in *Super Mario Galaxy*, the contracting circles of LIGHT (p. 219) protecting the player from deadly darkness in *Ori and the Blind Forest*, or the crystal in *Final Fantasy Crystal Chronicles*.



Super Mario Galaxy



FEZ



One Way Heroics



Player Unknown's Battlegrounds

Stealth

The player attempts to avoid being seen or otherwise detected.

Examples

Two early examples of stealth in games were *Castle Wolfenstein* (1981) and *Metal Gear* (1987), but 1998 marked the year when stealth became a major mainstream genre, with the hit releases *Thief: The Dark Project*, *Metal Gear Solid*, and *Tenchu: Stealth Assassins*. Stealth mechanics have since infiltrated a wide variety of first-person adventure games. A frequent meme about *Skyrim* and similar games from publisher Bethesda is that, no matter how you intend to play, you inevitably end up as a stealthy archer/sniper.

Monaco: What's Yours is Mine (2013) is a notable example of a multiplayer stealth game, and *Invisible, Inc.* is a rare example of a turn-based tactics game centering around stealth. The *Ape Escape* and *Sly Cooper* series integrate stealth mechanics into 3D collectathon platformers.

Analysis

Abstractly, this is another sort of AVOIDANCE (p. 43) – many stealth games are essentially avoidance games where the thing to be avoided is enemies' vision cones! But the theming is distinct enough, and the genre so well-established, that if I didn't include it I would probably get questions like "Where's Stealth!?"

Stealth games are interesting studies from the point-of-view of non-violent game design, because many *do* allow combat, but primarily as a last-ditch strategy when non-violent methods fail.

On the other hand, the association of non-violence with high skill play in stealth games has a perhaps counter-intuitive consequence: sometimes a player can't realistically do a fully non-violent run on their first playthrough, because the designer intended that as a *second* playthrough once the player has mastered the basics. It might require a more sophisticated understanding of the mechanics, money or EXP



Thief: The Dark Project



In *Skyrim*, all roads lead to Stealth Archer.

gathered on a previous violent playthrough to unlock non-violent skills and tools, or foreknowledge of quest design that a new player wouldn't have.

For example, in *Dishonored*, it can be quite difficult to avoid killing a particular NPC, Granny Rags. You can avoid fighting her entirely by never interacting with her to begin with, but how is a player supposed to know that hours before she becomes hostile? You can also sleep dart her, but this can result in her death if done at the wrong time. Even leaving her alone can lead to her death. The player simply cannot know the details of what will lead to her death without consulting internet forums or walkthroughs.

Challenge

By their nature, stealth games encourage careful and slow movement. Frustrate that by combining it with a pattern that encourages fast or wild movement like HIGH INERTIA (p. 118) or FUMBLECORE (p. 120). For example, what would a stealth game look like in space, where the player cannot stop on a dime?



Dishonored

Part III

Relationships and service

Relationship building

The player builds a relationship with an NPC, by interacting with them, doing things they like, and avoiding things they dislike (cf. NPC PREFERENCES, p. 54).

Examples

This is a central pattern in dating sims like *Tokimeki Memorial* (1994) and many modern life sim games like *Harvest Moon* (1996). It also shows up in many RPGs such as the *Star Ocean* and *Dragon Age* series, and even shows up in small-business sims like *Recettear* (2007) and *Touhou Mystia's Izakaya* (2021).

Historically, this pattern evolved out of Japanese eroge (erotic visual novels). Originally, eroge simply shepherded you through a linear series of trysts with different girls. But in the hugely influential *Dōkyūsei* (1992), the player instead chooses a particular girl to pursue and does so through gameplay systems (in particular by SCHEDULE DEDUCTION, p. 70). *Tokimeki Memorial* (1994) took this pattern and removed the explicit elements, marking the turning point from eroge to modern dating sims.

Relationship-building does not necessarily have to involve romance; they can represent things like customer loyalty as well. In *Recettear*, for example, the player must HAGGLE (p. 169) with customers over the price of goods. While you *can* drive a hard bargain every time, in the long run it's much better to offer deals that feel fair to both parties; satisfied customers return more often and spend more over the course of the game.

Analysis

To some extent this is a way of theming VICTORY POINTS (p. 33), since it's typically implemented by a counter from zero (e.g. strangers) to, say, 100 (e.g. marriage), and affected by certain actions (gifts, dialogue choices, etc.) getting some number of positive or negative points. But it also differs from victory points in an important way,



Tokimeki Memorial



In pigeon dating simulator *Hatoful Boyfriend* (2011), the human protagonist must achieve romance with one of many eligible pigeon bachelors. However, if she does not, the pigeon government concludes that humans and pigeons are incompatible and kills off the remainder of the human race.

that instead of a single counter towards victory, there are often a dozen or two different counters, one for each character. The designer can use these additional counters to incentivize broader exploration of gameplay systems.

For example, imagine a cooking game where you're just cooking for yourself (in, for example, a survival game) or for money. There is a tendency for the player to spam out the exact same recipe repeatedly – typically, whatever dish maximizes HP recovery or profit given the most easily-gathered local ingredients. This is a natural consequence of a “value chain” that ties all the different food ingredients, in the end, to one kind of output. Even if there are 100 different ingredients, the player will eventually figure out that they only need to concentrate on a few of them.

That's fine if your game really is about HP, and you just want to have a way to heal/cure that isn't strewing red potions everywhere. But say you want the game to really be about cooking, and incentivize the player to explore the possibilities of all those ingredients and recipes you took so much time designing.

This is where NPC relationships and preferences comes in handy. *Touhou Mystia's Izakaya* initially seems like it's a small business sim with a get-out-of-debt goal, but it's really not; this aspect of the game is trivialized midway through, and the game shifts to being almost entirely about satisfying the complex preferences of 15 named NPCs. (E.g., one character might like sweet, vegetarian, and Japanese foods, and dislike spicy or trendy foods.) Because your relationship with each of these people is a different counter, and different dishes raise/lower each counter to different degrees, you can't just focus on churning out the same few dishes. Granted, you will eventually find ideal dishes for each person – when Marisa appears I start making truffle fried tofu without a second thought – but figuring that out for 15 people, and developing new solutions as new recipes become available, is a rich enough puzzle that it can support an entire game.

Another thing relationship-building brings is a way for the game to give the player lots of choices, and acknowledge them meaningfully, without branching the main narrative into an exponential number of branches. Consider, for example, the player's early-game dialogue choices in *Final Fantasy VII*. Branching the narrative for every such choice in a 40-hour game would be prohibitive, especially ones so close to the beginning. Instead, most of these choices affect Cloud's relationship meters with the party members, culminating in a date later in the game. This date is short and epiphenomenal, it doesn't affect anything else in the game, but as minor as it is, it's a way for the game to make the player feel like their choices mattered.

Daniel Cook. 2021. “Value chains: A method for creating and balancing faucet-and-drain game economies”. In *Lost Garden*, Dec. 12, lostgarden.home.blog/2021/12/12/value-chains/



Final Fantasy VII was many Western gamers' first introduction to dating-sim-like relationship building. *Harvest Moon* did not receive a Western release until later in 1997. *Star Ocean* (1996) predated FF7 in adding dating-sim-like mechanics to a JRPG, but wasn't released in the west until 2008. *Tokimeki Memorial* never received a Western release at all.

NPC preferences

The player gradually learns the likes and dislikes of NPCs (e.g. friends or customers), generally in order to give them gifts to improve your **RELATIONSHIP** (p. 52), or to provide better **CUSTOMER SERVICE** (p. 56).

Examples

In *Stardew Valley*, every NPC has some level of preference towards each item in the game (Loves, Likes, Neutral, Dislikes, Hates), and giving them an item increases or decreases your relationship score with them accordingly. Some of these are a bit arbitrary and hard to predict, but some come to make sense as you grow to understand their hobbies and stories.

In *Touhou Mystia's Izakaya* (2021), named NPCs have specific likes and dislikes, which you slowly deduce through their orders, and by trial and error. If you manage to serve customers dishes that satisfy their order, hit several of their likes, and avoid all of their dislikes (cf. **CONSTRAINT SATISFACTION**, p. 34), they give you special bonuses.

In most games, you learn NPC preferences through repeated trial and error. In a few games, you only get one chance, and just have to guess. The *Iron Chef*-style cooking competition in *Suikoden II* works like this; three judges are chosen randomly from the gigantic cast of characters, you are given hints about what they like (e.g., that they like meat or vegetables), and you have to concoct a three-course meal that you think will outscore your opponent's meal. This is effectively a one-off guess; you're unlikely to have the same judge twice. This sort of guessing probably wouldn't work as a game on its own, but you can get away with more in one-off minigames like this.

Analysis

NPC preferences are usually a case of **HIDDEN RULE** (p. 213) deduction, and much of the analysis in that section also applies here. Like hidden rules, the player is rewarded and punished for reasons they



This character screen in *Touhou Mystia's Izakaya* shows what the player currently knows about Hieda no Akyuu's food preferences: she likes soups, sweet foods, etc.

do not, initially, understand. You give Penny a bottle of wine, she reacts poorly and you may even lose a “heart point”, and you don’t yet understand why.

However, NPC preferences aren’t as dangerous to include as hidden rules can be. Penalties are often slight and easily recovered from, and it doesn’t feel as arbitrary. “Is a bottle of wine an appropriate gift?” is an ordinary real-world uncertainty.

When using NPC preferences in a game, try to signal these outside of the “gift” or “sales” system itself. When possible, make it a part of the NPC’s story, dialogue, character design, or even environmental design. Emily’s room in *Stardew Valley* centrally features a bolt of cloth and a pair of scissors, reinforcing that Emily likes gifts related to textile crafts.

Remember that you the designer get to choose what items are in your game. You don’t have to decide on the items for purely mechanical means, and then make up a just-so-story about why each character likes each thing. You can do it the other way around: write characters’ stories and then populate the game with items that reinforce those stories mechanically.

Since NPC preferences are a kind of information with gameplay significance, gaining this information can itself be a gameplay reward (cf. *CLAIRVOYANCE*, p. 218). As mentioned, satisfying customers’ unique preferences in *Mystia’s Izakaya* grants you special buffs or rare ingredients, but serving them food they hate can lead to harsh penalties like scaring away all other customers. In general, you only learn about customers’ dislikes the hard way, after serving them something they dislike. However, one customer’s special reward is information about other customers’ preferences, letting you “skip ahead” and avoid some of the trial and error of working it out for yourself.



Emily’s room in *Stardew Valley*

Customer service

The player fulfills customer requests, often under time pressure.

Examples

This is the central pattern of many small business simulations, especially restaurant simulations like *Diner Dash* (2009), *Cook, Serve, Delicious!* (2013), or *Dave the Diver* (2023). A queue of customers wanting particular dishes sets up a time-sensitive optimization puzzle: how to fulfill these orders before the customers get angry and leave? Frequently, the number and diversity of orders slightly exceeds what would be straightforward to fulfill, and “rush” periods (e.g. lunch and dinner hours) further stress the player’s resources.

Customer service finds its way into a continuum of game styles, from hectic TIME MANAGEMENT (p. 68) games where the player personally has to serve every customer, to management simulations where customers are served by automated employees. Some games span the continuum during play, with the player initially serving customers directly but eventually hiring automated help.

COOKING (p. 57), and more broadly food/beverage service, is the most common theming of this pattern, but *Airport Mania* (2009) themes this as anthropomorphized airplanes queuing for boarding and refueling. *Hot Springs Story* (2011) has the player building an onsen to serve guests. *Theme Hospital* (1997) is effectively a MEDICINE (p. 58)-themed customer service game.

Analysis

Customer service has structural similarities to combat: something shows up, your goal is to hit some threshold to make it go away, and you get a reward when it leaves. We can especially see this in combat games like *Final Fantasy X* where the majority of enemies are meant to be dispatched by a particular kind of attack (magic, aerial, etc.); these are very much like customers needing to be dispatched with coffee and a doughnut.



Cook, Serve, Delicious!



Airport Mania: First Flight



Hot Springs Story

My new head-canon for *Final Fantasy X* is that the enemies are hungry customers who won't go away until they get several helpings of Blizzara.

Cooking

The player turns one or more ingredients into a meal.

Examples

In many games this is simply another kind of CRAFTING (p. 163) without embellishment, just one where the kitchen is the “work-bench”. Cooking in *Fallout: New Vegas* or *Stardew Valley* is indistinguishable from other forms of crafting in these games.

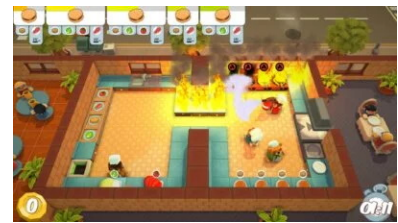
In other games, cooking also requires timing: the player must keep an eye on the clock to end up with dishes that are properly done as opposed to underdone, overdone, or on fire. *Cooking Mama* (2006) and *Cook, Serve, Delicious!* (2013) task a single player with a variety of ingredient- and timing-based micro-games. *Overcooked!* (2016) turns cooking into a chaotic co-op game where the players run a restaurant, often under absurd conditions that frustrate the best-laid plans.

Another special thing about cooking is that, unlike most CRAFTING games, you can sometimes customize recipes. The crafting recipes in many games are fixed; randomly adding brooms or telephones into armor recipes is thematically absurd and usually made mechanically impossible. Cooking is thematically more flexible; adding a chili pepper to fried rice still results in fried rice, just with (say) an additional “spicy” trait (cf. TRAIT PROPAGATION, p. 164). One of the progression systems in *Cafeteria Nipponica* (2012) is “leveling-up” your recipes, by proposing compatible toppings and setting one of your chefs to “research” it. Good combinations increase the quality and price of the dish; great ones can evolve into their own dishes.

TerraFirmaCraft, a hardcore survival mod for *Minecraft*, has only two recipes (sandwiches and salads), but the simulation underlying them is unusually detailed. The player puts quantities of ingredients into slots, and the resulting dish receives a 5-D flavor vector that’s a weighted average of the ingredients’ flavors. Meanwhile, each player has a unique flavor vector that they prefer; if a player eats a food close to their favorite flavor, they receive a nutrition bonus.



Props to *Cooking Mama*’s commitment to knife safety here. Also, that’s a nice bias cut. I’ve never actually played this but I’m impressed seeing two real-world cooking things in one screenshot.



Overcooked!



Recipe research in *Cafeteria Nipponica*



TerraFirmaCraft

Medicine and healing

The player is challenged to heal the ailments of NPCs.

Examples

Surgery is a long-standing (albeit a bit uncommon) theme for games. *Operation!* (1965) is an early example of a battery-operated game with a surgery theme. *Life and Death* (1988) was perhaps the first medical simulation videogame. *Surgeon Simulator* (2013) provides a more humorous take on the same genre. There are also an absolute flood of low-budget surgery games on the “girl games” side of the internet, where the player operates on unlicensed Disney princesses.

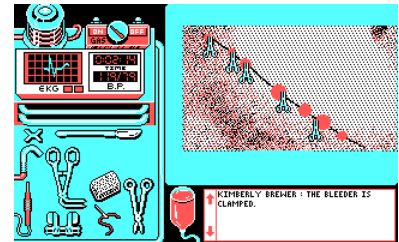
Dr. Mario offers a more abstract approach, where the viruses are puzzle elements that must be removed by MATCHING (p. 196). Sometimes, standard combat systems are themed as healing in this way, like in *Microcosm* (1993) or *Princess Remedy in a World of Hurt* (2016).

Like other CUSTOMER SERVICE (p. 56) games, these games span a continuum between role-playing an individual physician healing patients personally, and management games like *Theme Hospital* (1997) and *Project Hospital* (2018).

Being the healer in MMORPGs and other multiplayer online games is effectively a game unto itself: a sort of “spinning plates” mechanic where the healer must pay attention to their party members’ declining health and regularly top them off. *Little Healer* (2015) is a game that abstracts that process into a game of its own.

Analysis

Mechanically, healing is often treated as the dual of damage, and thus supports an easy conversion of combat-themed systems to non-violent themes, at least. In a combat game, the player’s actions decrement an HP bar until it reaches 0% and the enemy rewards the player with loot. Healing offers the possibility for the same game in different clothes: the player’s actions increment an HP bar until it reaches 100% and the patient rewards the player with payment.



Life & Death



Two-Point Hospital is a modern spiritual successor to Theme Hospital.



Little Healer

Rescue and emergency response

Examples

Obviously, thousands of games from *Super Mario Bros.* to *Maniac Mansion* have you rescuing *one* person at the end, but here I'm more considering games where rescue is a primary thing you do, like the *Emergency* series of RTS-like games where you deploy a variety of rescue workers/vehicles to save NPCs from emergency scenarios.

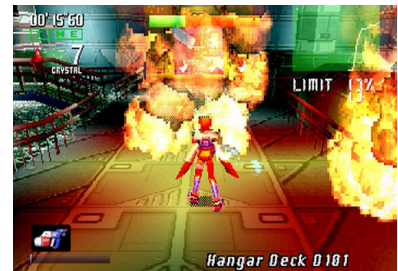
Firefighting games are a perennial way of converting shooter-style games to a non-violent theme, from light-gun rail shooters like *Brave Firefighters* (1999), to top-down shooters like *The Firemen* (1994), to vehicular shooters like *Air Inferno* (1990). The primary goal in most firefighting games is, unsurprisingly, rescuing civilians, or in the case of *Nuclear Blaze* (2021), cats.

Historically, *Fire!* (1980) and *Defender* (1981) are the first games I know of where you rescue people. *Choplifter* (1982) seems to be the first where you actually have to do search-and-rescue, traveling around scrolling map to find hostages and return them to your headquarters.

Shiryō Sensen: War of the Dead (1989), often considered the first survival horror game, is the first game I can find where the rescued NPCs are *characters* in the sense that they have names, dialogue, give hints and quests, etc.



Emergency: Fighters for Life



Burning Rangers (1998), from Sonic co-inventor Yuji Naka, was developed because Naka wanted to create a game where the player saved people rather than killing them.

Parenting and education

The player helps raise or educate an NPC (often a child), deciding on their activities in preparation for their future.

Examples

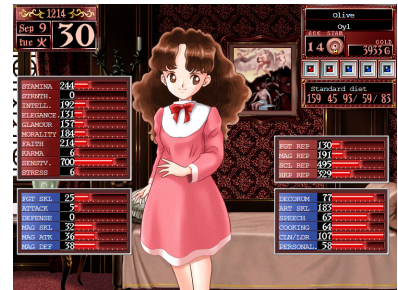
Notable examples are *Princess Maker* (1991) and its sequels and spiritual descendants (sometimes called “raising sims”). In them, you are tasked with protecting and educating a young person, and by your decisions influencing who they become. These games often have many endings – the original *Princess Maker* has 74 – and part of the meta-game is collecting as many endings as possible.

Crusader Kings II (2012) has another interesting spin on this mechanic. It’s a strategy game where, rather than being an immortal, disembodied leader (like in *Civilization* games), you play a mortal leader in the world; when they die, you play as their heir. The decisions you make about raising children determine their stats as an adult... and thus *your* stats when, in a few years, you come to inhabit them in turn.

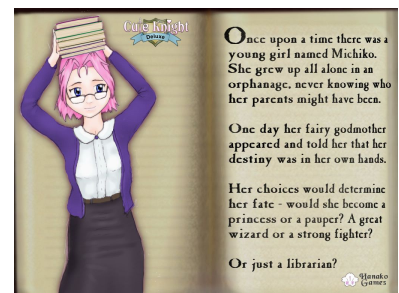
Analysis

I want to highlight here how parenting (and educational supervision, and sometimes employee management) differ from other kinds of RELATIONSHIP BUILDING (p. 52), which are typically about fulfilling NPC PREFERENCES (p. 54). Why *not* spam everyone’s favorite foods and other “kindness coins” until everyone adores you? It’s not that interesting a decision because there’s no real downside, except that you might have spent your time/money doing something else. When NPCs like cheap things with low opportunity costs (e.g. Leah in *Stardew Valley*), why not just give them that all the time?

Raising/educating a child is very different; you have bigger goals than just making them happy. They might *prefer* to eat cake and play games all day, but keeping them happy isn’t your only goal; you have more important overarching goals in their survival and success.



Princess Maker 2 Refine



To some extent parenting vs. other kinds of growth is a matter of theming; *Cute Knight* (2005) is mechanically very similar to *Princess Maker*, except that the player is themed as *being* the young person rather than *raising* them. In a way, every RPG player is “raising” their characters!



Crusader Kings II

Matchmaking

The player encourages relationships between NPCs.

Examples

This is rather less common in games than protagonist-NPC RELATIONSHIP-BUILDING (p. 52), but does appear in what we might call “virtual dollhouse” life sims like *The Sims* (2000) and the *Virtual Villagers/Virtual Families* games, where the player is more of a god-like influence rather than a character, and can indirectly form romantic relationships by (for example) putting characters in proximity. In village management sim *Lakeburg Legacies* (2023), the player takes a more active hand; pairing up villagers is central to gameplay.

In strategy games, the *Crusader Kings* series, *Fire Emblem: Awakening* (2012), and *Massive Chalice* (2015) are notable for having NPC matchmaking that begets future generations of NPCs (cf. ANIMAL HUSBANDRY, p. 63).

Analysis

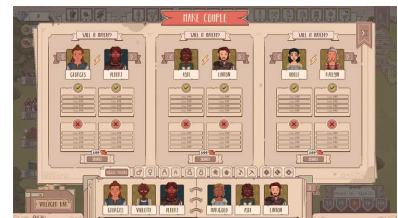
Romantic matchmaking between NPCs seems like it would be an obvious addition to cozy village life sims, so why is it relatively rare? But it’s one thing to dream about the possibilities of your characters pairing off (and even making new generations), and another thing to actually implement the factorial explosion of story content.

Therefore, many matchmaking games don’t have NPCs with pre-defined characters and pre-written content, but instead procedurally generate NPCs and have their relationships emerge systemically (e.g. from a giant matrix of friendships and animosities in *Crusader Kings* or *Rimworld*).

Games with variable relationships between predefined characters, like the *Star Ocean* series, the *Fire Emblem* series, or *Unicorn Overlord* (2024), tend to keep the consequences light. While relationship meters exist between all characters, non-protagonist relationships mostly trigger simple skits or slightly different ENDINGS (p. 262).



Virtual Villagers: A New Home (2006)



Lakeburg Legacies



Star Ocean: The Second Story R (2023)

Animal companions

The player befriends and/or raises one or more animals, which confer advantages or special abilities.

Examples

Animal companions in games are often a sort of semi-autonomous weapon (e.g. *Pokémon*, or Dogmeat in the *Fallout* series), but there are also many non-violent uses of animal companions in games.

In *The Boy and his Blob* (1989) or *Crusaders of Centy* (1994), all of the player's special abilities are embodied as animal companions.

Horses and similar mounts are common in games; I cover those more thoroughly in VEHICLES (p. 113).

Some animal companions serve as pack animals, increasing the player's carrying capacity, like the mule in *Dungeon Siege* (2002). The pets in the *Torchlight* series (2009–) go one step further: your pet can run back to town to sell the loot that you give it. In a game where you can get hundreds of pieces of loot per dungeon delve, this is a very useful ability, although while it's doing so you're left temporarily shorthanded in battles.

The pets in *Stardew Valley* play little in-game role besides contributing to the cozy flavor of the game, although petting them regularly does eventually gain you some VICTORY POINTS (p. 33) towards Grandpa's evaluation.

Analysis

Be aware that players demand that dogs/cats in games be pettable even if this has no gameplay consequence, and will hound you (no pun intended) until they are. So *let them pet the dog*.



Crusaders of Centy



Dungeon Siege

Animal husbandry

The player attempts to breed animals to have particular traits.

Examples

Many *Pokémon* games, starting with *Gold* and *Silver* (1999), allow you to breed Pokémon; two Pokémon of opposite genders and compatible species, if left at a “Day-Care Center” together, may result in an egg. After the player has taken a certain number of steps, the egg hatches into a new Pokémon, their species and stats determined by a complicated algorithm that is not revealed to the player.

Final Fantasy VII (1997) let you breed chocobos. Chocobos can be rescued from random encounters anywhere on the overworld map, and brought back to the chocobo stables. Breeding special chocobos results in in-game advantages, not just in the Chocobo Racing minigame, but because different colors of chocobo can cross different terrain on the overworld map (cf. *VEHICLES*, p. 113), allowing the player to access otherwise-inaccessible treasure troves.

Analysis

Animal husbandry is a kind of *CRAFTING* (p. 163) applied to NPCs, and much of the discussion there applies; like crafting, the process is transitive (children can in turn become parents) and *TRAIT PROPAGATION* (p. 164) is possible through the generations.

A common factor of husbandry systems is that they typically operate by *HIDDEN RULES* (p. 213); it’s rare that the system of inheritance is spelled out explicitly to the player. *Pokémon*’s husbandry system isn’t explained in-game; what we know about it has been worked out by players reverse-engineering the game’s systems.



Pokémon Gold/Silver



Final Fantasy VII

Gardening

The player plants a seed or sapling in the ground, and a plant grows, which in turn may give one or more items, or serve as a tool or platform. This growth may take time and/or watering (as a DAILY ROUTINE, p. 74), or it may be instantaneous.

Examples

Obviously, there are the many farming games like *Harvest Moon* or *Stardew Valley* where the primary game loop involves spending seeds, water, and time and getting useful and/or salable plants in return.

There are also many games where the plants serve as useful in-world tools and platforms. *Plant Cat: First Blossom* (2013) is a clever puzzle-platformer where the player plants *Mario*-style vines, but with a more complex growth algorithm than in *Mario* games. The Planter in *Lemmings 2: The Tribes* (1993) similarly grows a plant that lemmings can walk on. Also, while they are not *plants* exactly, *Starseed Pilgrim* (2013) has you planting seeds that grow into variety of platforms with different effects on your platforming.

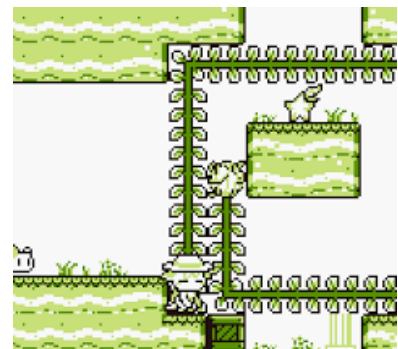
Grow Home (2015) does something similar in three dimensions, with the main character growing and riding giant vines into the sky.

Challenge

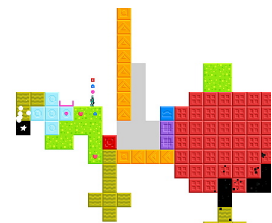
The vines in *Plant Cat: First Blossom* work by a simple algorithm that takes into account the environment, much like the Chus in *ChuChu Rocket!* (1999) and other LEMMING-type (p. 128) games. Design an exploratory platformer about manipulating the environment to manipulate the growth algorithms of plants, that the player character can use as platforms and ladders to reach inaccessible areas.



Harvest Moon



Plant Cat: First Blossom



Starseed Pilgrim

Ecosystem maintenance

Rather than just concentrate on animals and plants as individuals, the player is caring for or otherwise manipulating a complex ecosystem of animals and plants.

Examples

SimEarth (1990) and *SimLife* (1992) are two early ecosystem simulators of note.

Earthtongue (2015) is a relaxed virtual terrarium centered around fungi and insects, with fungi spores growing into a variety of structures that have complex relationships with each other and with the insects. Also be sure to check out the other modes in this game for whimsical alternative ecosystems.

Waking Mars (2012) uses an alien ecosystem to support an exploratory puzzle-platformer. The plant-like creatures have complicated relationships with each other and with the local fauna, and the player must explore and exploit these relationships to progress through the Martian caves.

Analysis

One important thing about ecosystem simulation is to remember Sid Meier's maxim: to make sure the *player* is the one having the fun. Complex systems like ecosystems and economies (cf. TRADING, p. 102) are fun to program, but can be impenetrable and frustrating to the player, as their attempted interventions have minimal, unclear, or counter-intuitive effects.



SimEarth



Earthtongue



Waking Mars

Cleaning

The player cleans up a messy or degraded environment.

Examples

The first major cleaning-oriented game I can think of is *Super Mario Sunshine* (2002), although as a relatively minor part of gameplay.

Recently, however, clean-'em-ups have emerged as small genre of their own, including *Viscera Cleanup Detail* (2015) and *PowerWash Simulator* (2022). These games take the basic controls and point-of-view of the FPS and replace the weapons with cleaning tools like hoses.

Analysis

While cleaning is not most people's favorite chore, the popularity of powerwash videos online reveals that there's something viscerally satisfying about turning something *very* dirty into something sparkling clean.

The developers of *No Place Like Home* (2020) got so much customer feedback wanting more cleaning in that game that they released a stripped-down cleaning focused successor: *Fresh Start Cleaning Simulator* (2022).

I think cleaning works best in games where it's a central activity rather than presented in opposition to another activity. For example, *Super Mario Sunshine* is still mostly about Mario's acrobatic platforming, and in contrast to that the requirement to clean every speck of goo from an area feels slow and tedious. This complaint doesn't strike while playing *PowerWash Simulator* or *Fresh Start*; I knew that slow and methodical gameplay was in store when choosing to play them in the first place.

As mentioned in AREA CONTROL (p. 25), part of the appeal of this sort of gameplay can be revealing interesting things under the grime, so if you're making a clean-'em-up I suggest making sure there are regularly interesting art assets underneath the grime.



PowerWash Simulator



Fresh Start Cleaning Simulator

Part IV

Time-related patterns

Time management

The player has a limited amount of time that they must decide how to spend.

Examples

Most games have time management to some degree – you have various choices of things to do and have to mentally calculate the payoffs of spending your play time this way or that.

However, it's the central pattern in certain nonviolent games to the point that whole swaths of them, particularly ones involving time-limited CUSTOMER SERVICE (p. 56) like *Diner Dash*, are sometimes called “time management” games.

Time management doesn't necessarily involve hectic real-time running around, though. It can be discrete as well, like raising-sims or dating-sims in which you are deciding what a student should do during their lunch break – read in the library? Hang out with friends? – with differing rewards.

Analysis

The challenge is that there's always more to do than time to do it in.

Players differ significantly as to whether they find this challenge engaging or stressful. A common complaint about *Stardew Valley* is that players feel overly-anxious about time pressure, feeling like they have to constantly pack more activities into the short days than actually fit. This is actually somewhat ironic because *Stardew* doesn't have any time limits or fail-states. If you want to spend a whole year fishing, go ahead.

I think the *lack* of fail-states may even exacerbate this effect. As Hudson (2015) noted in an essay about the appeal of frenetic “pink collar” time-management sims:

While that might not sound fun, exactly, time management games are designed with a very smooth and deliberate learning curve that



Diner Dash

...And in the game.

Laura Hudson, 2015, *Subversive games about waitresses and hairdressers*. At boingboing.net, Apr. 22.

This essay is quite insightful in general, about the paradox of why stressful games about overworked women are so popular among stressed and overworked women.

makes you feel challenged, and occasionally overwhelmed at times, but always like you can handle it in the end.

Did your customers end up storming out, red-faced and furious, because you didn't serve them fast enough? Just play the level a few more times, and you'll be able to pick out the entitled executive who's a big tipper with a short temper, and serve him before the kindly old man who seems happy to wait all day. Soon enough, the screen will be festooned once again with hearts and praise.

These games are always up-front about what is expected of the player, and success at meeting them is always acknowledged and rewarded.

Stardew, on the other hand, is actually rather misleading about what it expects from the player – it ominously tells you Grandpa's ghost will return to evaluate you, but gives no clue as to *what* you're being evaluated on. In reality, Grandpa's evaluation is easily met and even if you fail it you can try again... but new players don't know that! Unlike *Diner Dash*, the game doesn't regularly tell the player whether they're using their time correctly – but this means that players wait in limbo for three years, all the while imagining they're secretly being judged. If the game is too coy about what "winning" or "losing" might be, that can be very anxiety-inducing in itself.

Schedule deduction

The player learns to anticipate a recurrent event, to perform some time-limited action.

Examples

This pattern adds a mild puzzle element to many dating and life sims in which NPCs have autonomous routines that the player can learn to predict.

The first game I'm aware of that implemented this systematically was proto-dating-sim *Dōkyūsei* (1992); as I mentioned in *RELATIONSHIP BUILDING* (p. 52) this game also introduced the idea of multiple love interests with whom romantic success depended on gameplay systems. A key aspect of this was learning the girls' schedules; you cannot romance a girl that you cannot find!

Schedule deduction is also very important to *TIME LOOP* (p. 72) games like *The Legend of Zelda: Majora's Mask* (2000) and *Outer Wilds* (2022). The planets in *Outer Wilds*, for example, undergo some changes over time that must be understood and anticipated in order to delve their mysteries.

Analysis

While schedule deduction isn't exactly a *hard* puzzle, it does take up a lot of mental space, so I feel that this pattern works well in mechanically-minimalist games like visual novels or *Outer Wilds*, where it's one of the main things going on and thus the player can focus on it.

In a game where there's a lot more going on and *TIME MANAGEMENT* (p. 68) is rather more hectic, not being able to find an NPC when you need them can be very frustrating.



Dōkyūsei. A display in the corner indicates the in-game date and time, which are needed to learn the girls' schedules.



The modern remake *Dōkyūsei: Bangin' Summer* includes an Easy Mode that simply tells you where the girls are on the town map. This removes most of the challenge from the game, and shows how crucial schedule-deduction was to the "game" aspect of this visual novel.



Some areas in *Outer Wilds* can only be accessed during brief periods.

Time travel

The player travels to the past or future, often making changes that affect later time periods.

Examples

There are many “time travel” games where this does not affect the mechanical structure of the game, and only serves to allow diverse level themes (say, dinosaur times, ancient times, the Wild West, and spaceships) in the same game (e.g. *Back to the Future* (1989) on the NES). *Space Quest IV* has the clever premise that the protagonist is time-traveling through their own series’ (including fictional future games that do not exist yet like *Space Quest XII*), but so far as I remember, your actions in past games do not cause consequences future games, it’s just a fun way of theming the different worlds.

Here, I’m concentrating more on games where the player’s actions in past periods cause changes in future ones, like *Chrono Trigger* (1995), *Steins;Gate* (1999), or *Shadow of Memories* (2001). For example, *Day of the Tentacle* (1993) takes place in three time periods – the American Revolution, the present day, and 200 years in the future. A major puzzle involves getting one of the characters down from a large tree in the future, and the solution involves convincing George Washington that it’s a cherry tree, 400 years in the past, so that he cuts it down and thus makes it cease to exist in the future. Another puzzle requires a tiny, hamster-sized sweater, which you create by putting a human-sized wool sweater in the dryer, and finding enough quarters to run the dryer for 200 years.

Analysis

Many time-travel game shade into PARALLEL WORLD (p. 99) games – they take place on the same map (or *almost* the same map), just in different times. Other times it varies; in *Chrono Trigger*, the 600 A.D. and 1000 A.D. time periods share the same map, but the other periods are largely different maps.



Day of the Tentacle takes place in one location over three time periods.

Time loops

The player is trapped in a *Groundhog Day*-like time loop where the world regularly resets.

Examples

To some extent most games are time loops already – you can reset games at any point and tackle them anew with better understanding of the world and its rules – but some games more deliberately weave the expectation of resetting into their stories and core mechanics.

The Legend of Zelda: Majora's Mask (2000) is the first major game I can think of that does this. (*The 7th Guest* (1992) involves a time loop story but this doesn't really play into the mechanics of the game.)

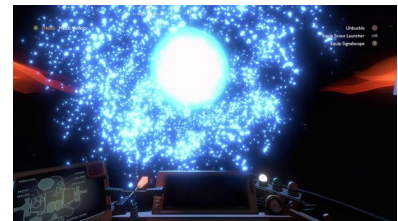
Sometimes, things the player does (like obtaining items or opening doors) survive into the next time loop, as in *Minit* (2018), in which the player is cursed to die every 60 seconds. The player's goal in each life is to make it far enough to achieve at least one permanent change in the game world. The spin-off *Minit Fun Racer* (2021) adapts these ideas to an arcade racer, with an even tighter time limit of 10 seconds. (Thankfully there are time pickups to mitigate this tight schedule.)

In *The Forgotten City* (2021), the player is physically jumping into the time portal, so their inventory survives even though their changes to the world are undone.

In *Outer Wilds* (2022), on the other hand, nothing is retained except knowledge. It opens 22 minutes before the sun goes supernova and destroys its solar system, at which point the protagonist wakes up, for reasons they do not initially understand, 22 minutes earlier. Since the solar system is going to be destroyed, no changes that they make to the world carry over into the next cycle. However, things that they *learn* are carried over (and are helpfully retained by the ship's log, even though this technically doesn't make a lot of sense lore-wise).



The Legend of Zelda: Majora's Mask



Outer Wilds



Minit Fun Racer

Day/night, weather, and seasons

The gameworld changes based on the time of day, the season in the year, or the weather, often changing what items are available for gathering or constraining what the player can do.

Examples

These are common in games; *Stardew Valley* has all three. What you can collect and grow depends on the time of day, the season, and the weather – fish availability, for example, are dependent on all three factors.

Analysis

Adding cycles like this is common in games to encourage or force the player into different activities – when night, or winter, rolls around, some activities become difficult or impossible (e.g. you can't farm in winter), whereas other activities become available (e.g. the bar opens in the evening).

It also can add a strategic layer to gathering by encouraging the player to think about their schedule. How will I structure my day so that I end up at the shore at night, because night fishing is the only way to get a particular fish?

On the other hand, you should be careful especially with seasons, because seasonal cycles are slow. Several fish in *Stardew Valley* are only available a few days a year (e.g. on rainy summer days), and if the player misses this, they'll have to wait until that season rolls around again. If you have rare seasonable collectables, also consider including something like a randomized SHOP (p. 168) that gives the player a chance to obtain missed items in another way.



One screen from *Stardew Valley* in all four seasons.

Streaks

The player is required or encouraged to do the same action every day (or whatever the in-game cycle is), with rewards if you keep it up.

Examples

This is a central pattern in many farming games: water your crops every day and eventually you'll receive something you can eat or sell, but fail to water your crops and they can wither and die. Feed and milk your cows every day and you'll get milk, keep it up for days in a row and you'll get higher quality milk, but if you forget a day you get an angry cow and its milk is reset to its basic quality.

Daily routines sometimes involve DIARIES (p. 78), in which some character records the events of the day (or other cycle).

Analysis

While this pattern might seem inconsequential and boring, players really do seem to enjoy daily routines like this, especially when they're themed to be cozy and pleasant.

One thing to keep in mind, though, is that streak-maintenance can also be stressful; did I remember to milk the cow today? It's often good to allow a bit of lenience in streak tracking – e.g. a single missed day does not end the streak, but two missed days in a row does.

When tied to a real-world clock, as they are in *Animal Crossing*, this pattern can also encourage players to come back to the game every day, as well as limit the amount of time a player can play on any one day.



Stardew Valley

Festivals

Periodically, part of the gameworld is replaced by a festival in which the player can participate, interrupting normal gameplay and replacing it with time-limited activities.

Examples

Stardew Valley provides many examples: periodic festivals introduce mini-games (like the Easter-egg hunt or the strength-trainer), new dialogue for each of the NPCs, and change up the maps (as in the Night Market event).

Festivals are also used frequently in MMORPGs and other online games, especially as time-limited special content is a way to draw lapsed players back into the game.

Analysis

Obviously, festivals add variety and break up the monotony of doing the same thing every day, just like in the real world.

It's also worth pointing out that they're a way to take some otherwise simple activities that don't really hold up as full-fledged mini-games, and motivate the player to participate by making them time-limited. If the strength-trainer or spin-the-wheel mini-games in *Stardew Valley* were available in the saloon, would you play them more than once? I certainly wouldn't, but collecting them into a once-a-year festival gives them a specialness that they wouldn't have as stand-alone activities.



The Night Market event in *Stardew Valley*



The Starlight Celebration in *Final Fantasy XIV*

Resting

The player rests to regain health, remove status effects, save the game, or otherwise reset some aspect of gameplay.

Examples

Most traditional RPGs have resting as a key part of their game loop, typically at “inns”. (There are usually some other ways to heal and rest, spells and “potions” and “tents”, but inns are still the prototypical means of resting, and often are the only rest type that fully heals all ailments.) This sets up a town-field-town cycle that forms the medium-term backbone of play: the player plays as long as they can before their HP/MP/items are depleted, then returns to town to rest and spend money.

Although resting is central to many RPGs, in actual implementation it’s usually an afterthought. The lights dim and brighten, and it’s a new day. But some games stick out for the way they emphasize rest periods to reinforce a cyclical game feel and deepen the relationships between party members.

Grandia II (2000) and *Final Fantasy XV* (2016) are memorable examples. Resting in these games features the party camping or eating together, and the player gets humorous conversations and relationship development between party members. This helps emphasize their identity as a group, but also as individuals with relationships to the other individuals in the party.

Ico (2001) has a less-elaborate system of rest but it’s still worth a mention. Save-points in *Ico* are themed as couches, and when the player saves the two main characters take a nap, often leaning against each other. This highlights the sibling-like physical affection between the main characters. It’s quite rare that videogame characters show physical affection; it’s tricky to animate characters touching without clipping through each other, and as a result even close friends and lovers practice social distancing in games. *Ico* and *Yorda* are rare exceptions – holding hands is even relevant to the gameplay – and the developers saw that even a mere save-point animation could



Grandia 2



Ico and *Yorda* take a nap.

reinforce this aspect of the game.

Sometimes the resting stage involves a recap or diary of the day (or cycle's) events; I detail this in *DIARIES* (p. 78).

Analysis

Most games give little thought and effort to elaborating this part of the game... and that's entirely reasonable. Developers have a finite amount of time to add features and "resting" is honestly far down the list of important things to care about.

Nonetheless, I mention it here because highlighting the process of rest is a potentially useful tool in your toolbox. It can give a game a distinctly cozy, friendly feel. I wouldn't necessarily include this in a general book about game design patterns, but cozy/wholesome game feels are what a lot of non-violent game designers strive for.

Diaries

Someone keeps a periodic log of the player's accomplishments.

Examples

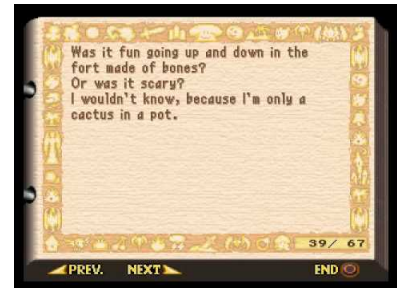
Many games keep some kind of log of plot events and other things that the player has done, which are generally ignored by players until they forget a mission briefing. But a few log systems are worth exploring in more detail because of their uniqueness and their contributions to the emotional feel of their game.

Legend of Mana (1999) has what's effectively a DAILY ROUTINE (p. 74) that corresponds to its mission cycle. The game has no single overarching plot, but instead is structured as a series of short vignettes. After completing a vignette, the player can go home and talk to their sentient cactus pet, Li'l Cactus. After you leave, he writes a summary of the event in his diary. This diary, although optional, is your main record of what you've done in the game, and completionists fill it out in its entirety. Your bucolic home and the childlike writing of Li'l Cactus set up a cozy home-adventure-home loop.

Final Fantasy XV (2016) has an interesting photography system that serves, in part, as a diary. Your party member Prompto is a hobbyist photographer, and snaps photos of what you do that day. When the party camps for the night, you can look at those photos and save the best in an album, which can serve as a visual SHOWCASE (p. 85) of the game's iconic characters, scenes, and settings.

Analysis

As with RESTING (p. 76), I wanted to highlight these games because they show that an often-ignored part of the game UI – the log – can be given a distinct in-game voice in such a way that it reinforces the central pillars of the game. *Legend of Mana's* childlike observations fit in perfectly with the game's pastel coziness, whereas Prompto's photos reinforce *FFXV's* central vibe of taking a roadtrip with a close-knit band of bros.



Li'l Cactus records his impressions of your accomplishments in *Legend of Mana*.



The character Prompto in *FFXV* takes photographs that serve in part as a visual record of your adventures.

News

The character can read, watch, or listen to segments that describe current events, foreshadow future events, recap past events, or reveal hints and other useful information.

Examples

Many games theme event announcements and/or player triumphs as newspaper headlines; the *SimCity* series is a notable example.

The *Grand Theft Auto* games are masterclasses in worldbuilding by radio broadcasts, foreshadowing future events and describing the player's own actions after-the-fact as news bulletins. Like Li'l Cactus's DIARIES (p. 78), they provide an outsider's perspective on events; the broadcasters don't completely understand the details or motivations behind the events they're describing.

Stardew Valley lets the player turn on the TV, with several different programs: weather reports, a COOKING (p. 57) show revealing new recipes, a fishing show revealing seasonal fishing opportunities, etc.

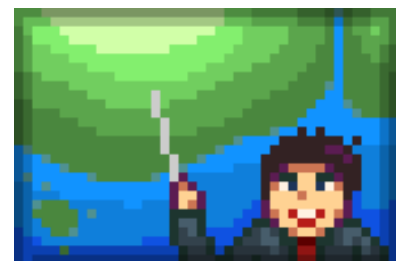
Analysis

Although these are minor patterns and rarely impact the main game loops, I want to highlight the role that news, DIARIES (p. 78), and SHOWCASES (p. 85) can have in making the player feel like the game is acknowledging their accomplishments.

I think this is particularly important in a non-violent RPG. In a typical RPG where you're going around slaying dragons, it's easy for the player to imagine that their actions are important and valued even if the game is a bit silent about this. But a lot of non-violent game protagonists walk rather lightly on the earth, and are doing more quotidian things. The protagonist of *Alba* (2020) is a child taking PHOTOGRAPHS (p. 91) and picking up litter, and it would be very easy for the player to decide their actions don't matter, but the game closes off missions with newspaper headlines showing how your actions have big consequences for your community.



SimCity 2000



The weather report in *Stardew Valley*



Alba: A Wildlife Adventure

Part V

Building patterns

Map building

The player builds the game-world itself during play.

Examples

You certainly know *SimCity* (1989), but you may not be familiar with its origin story. While making the shoot-em-up game *Raid on Bungeling Bay* (1984), Will Wright realized he enjoyed designing the cities that you flew over more than he enjoyed actually playing the game. Taking a city map editor and adding a simple economic simulation underneath begat *SimCity* and launched an entire genre of games.

In adventures/RPGs, a notable use of map building was in *Legend of Mana* (1999). Instead of having a pre-made gameworld, the player makes the worlds by placing artifacts onto a map, which turn into towns and dungeons. Like many building games, there are elements of a TILING PUZZLE (p. 190); the elemental properties of each location “bleed” into neighboring tiles and affect game systems that are sensitive to the elements. *Final Fantasy Tactics Advance* (2003) used a similar system.

Speaking of tiling puzzles, this section probably shouldn’t go by without a mention of *Carcassonne*, one of the most venerable modern “gateway” boardgames, and map-tiling games inspired by it like *Dorfromantik* (2021). In *Carcassonne*, the adjacency constraints are strict – you *cannot* place tiles next to each other unless their edges match – making map-building more like a jigsaw puzzle than the more freeform placement of *SimCity* or *Legend of Mana*.

ActRaiser (1990) and *The Horde* (1994) also deserve special mention here as early examples of hybridizing city-building into more action-focused adventures. *Star Control III* (1996) attempts something similar in an open-world space game, although not entirely successfully.

Analysis

City builders proper are a well-established genre and I won’t concentrate much on them here. If you were building a pure city-builder, I



SimCity



Legend of Mana

figure you'd just build one rather than read this book for inspiration.

If you're making a more general adventure/RPG, letting the player build the map can seem like a neat feature to add, but you should definitely pause and ask yourself what it adds from the player's point of view.

What are some of the benefits?

- Player-built maps can be a powerful vehicle for player SELF-EXPRESSION (p. 39). Part of the draw of *Cities: Skylines*, *Minecraft*, or even *Carcassonne* is just that you're making something visually interesting, something you might even like to show off. In *Townscaper* (2021), you're not making towns for any reason except that they look cool and you might want to show them off, and it totally works.
- It naturally handles MAP EXPANSION (p. 228). *Legend of Mana* doesn't need to have lots of locked doors, grumpy guards, and broken bridges to constrain the player's access to the full map. Later areas aren't merely *inaccessible*; they *don't exist at all* until the player finds the associated artifact.
- As I mention in HUB EXPANSION (p. 230), when the town is small and populated with named NPCs like *Ni no Kuni 2: Revenant Kingdom* or the *Dark Cloud* series, it can contribute to the feeling that you're caring for and building RELATIONSHIPS (p. 52) with these people. What's more fundamental to caring than giving people a place to live, and work? Giving the player more agency in town planning can solidify this feeling.

A caveat of map-building is that you the designer lose some of your ability to put together a coherent setting. A common criticism of *Legend of Mana* is that its plot arcs are disjointed and hard to follow, and I think the lack of a fixed world map contributes to this.

Another issue is that that, if there are few actual consequences to placements, it can feel like the game not adequately acknowledging your decision and work. Your placements in *Legend of Mana* do have some gameplay consequences, but they're subtle and only players who like to dive into and optimize mysterious systems will really take advantage of it. I never felt like my map-building decisions in *Final Fantasy Tactics Advance* really made much of a difference at all.



Townscaper

Level editors

The player designs a level or puzzle to challenge other players, or themselves at a later time.

Examples

A very early example of this was *Mouse in the Maze* (1959) for the experimental TX-0 computer at MIT, in which players used a light pen to draw a cheese-filled maze for a simple AI mouse to navigate.

The first consumer example of this that I know of was *Pinball Construction Set* (1982), which allowed players to design, play, and share their own pinball tables. *Lode Runner* (1983) and *Excitebike* (1984) are other notable early examples of games that shipped with level editors.

Guild of Dungeoneering (2015) and *Loop Hero* (2021) are interesting examples of incorporating level editing directly into RPG gameplay. The player has no direct control over the heroes, but can place level elements with the goal of challenging them just enough to level them up, but not so much that they perish in the attempt.

To some extent multiplayer builder/siege games like *Clash of Clans* (2012) or *The Castle Doctrine* (2014) might fall under this as well; the player is building a castle to challenge other players seeking to invade.

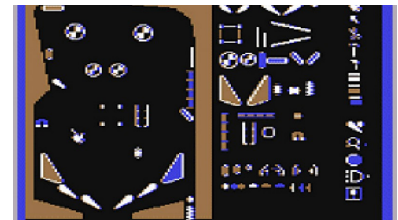
Analysis

There's a fine line between this and MAP-BUILDING (p. 81) but that chapter was already two pages. So I put the more small-scale examples here, particular ones where what you're placing are elements that cause yourself challenges in a different mode of play (cf. SELF-CHALLENGE, p. 39). After all, we're not typically placing elements in *SimCity* or *Legend of Mana* to deliberately frustrate ourselves in the future, but that's exactly what I'm doing in *Super Mario Maker*.

An interesting cross-over pattern is seen in *SimCopter* (1996), a flight simulator that let you use your saved *SimCity* cities as levels.



Super Mario Maker



Pinball Construction Set



Excitebike



Loop Hero

Base-building and home decoration

The player gets to build or decorate their own home base.

Examples

Building and/or decorating your home has become quite common in games, especially in crafting and survival games like *Minecraft* and *Subnautica* and life sims like *Animal Crossing*. It's even worked its way into platformers like *Kirby's Epic Yarn*.

It's also an important component of hospitality management games like *Hot Springs Story* or *Bear and Breakfast*, where decorations can have a gameplay value (e.g. contribute to a score that raises the value of the resort) beyond their purely aesthetic contribution.

Building/decor can also be a spur for role-playing in multiplayer games. The decor in *Habbo Hotel* has little gameplay significance, but the game itself is a sort of elaborate chatroom anyway, and players use decor to reinforce the themes of the role-play (e.g. that they're on vacation in a luxurious resort or employees of a business).

Analysis

Even when decoration doesn't lead to direct gameplay rewards, getting decor to match is a straightforward sort of optional SELF-CHALLENGE (p. 39). The game can't necessarily decide whether a given arrangement "works" aesthetically, but the player can, and that can be enough to fuel hours of arranging, re-arranging, and tracking down that last missing red chair to complete the dining room.

The amount of base-building and decoration has skyrocketed in games within the last 15 years, and this is probably due to social media and online multiplayer making it easy to show off your work (cf. MULTIPLAYER VISITATION, p. 86). *Minecraft* would probably have been a huge hit even before the Internet Age, but not *that* huge. As I note in PHOTOGRAPHY (p. 91) and TILING PUZZLES (p. 190), when player can create something beautiful, make it easy to share on social media; it's free advertising.



Subnautica



Kirby's Epic Yarn



Bear and Breakfast



Habbo Hotel

Showcases

The player's home base contains a showcase representing what they've collected or made in-game.

Examples

In *Dinosaur Fossil Hunter* (2022), the protagonist is not just finding the fossils, they are also creating a museum to display them. This is not only a thematically-appropriate addition to the game, it also helps to remind the players of their achievements so far.

This can also work well in crafting games, particularly ones where the player's creativity plays a role in the item's visual appearance. *Master of Pottery* (2019), for example, lets players arrange their best works in a showroom.

Analysis

This is fairly closely related to the **BASE-BUILDING/HOME-DECOR** (p. 84) patterns in the previous chapter, but here I'm concentrating more on collections that have a more direct relationship to in-game achievements. (E.g., while I know the furniture in *Kirby's Epic Yarn* came from me completing levels, I don't actually look at the couch and remember fondly the level it came from.)

Like a photo album or yearbook, showcases provide an opportunity for the player to look back over their playthrough and remember the high-points. In some ways they're an instantiation of the **DIARY** (p. 78) pattern, just one that you can walk through.

They can also be a motivation for the player to do their best. The grange exhibit in *Stardew Valley* doesn't have a huge in-game importance, but just giving the player an opportunity to show off their best work (even just showing off to NPCs) can reinforce the idea "What you're doing is important and people in this world care about it."



Dinosaur Fossil Hunter



Master of Pottery



The grange display in *Stardew Valley* is an opportunity to show off your best produce... or just play a prank on the mayor.

Multiplayer visitation

Players can visit other people's home bases or other creations.

Examples

Various games have let you share things that you've designed (e.g. *Pinball Construction Set*, 1982), and text-based MUDs/MUSHes had allowed players to build shared online spaces, but I think it was *Animal Crossing* (2001) that had the biggest influence here. As the series originally launched on the N64, it did not have *online* multiplayer, but although only one player could play at a time, four different players could play in the same world at different times: visiting each others' towns, sending gifts, etc.

Visitation remains a common semi-“multiplayer” pattern in games that allow creative SELF-EXPRESSION (p. 39), letting the player show off what they've made in a more immersive way than just sharing a screenshot. It's one of the main sorts of multiplayer interaction in *No Man's Sky* (2016), for example.

Analysis

Starting a new creative/building game can be like sitting in front of a blank document. What should I write? What *can* I write? One benefit of multiplayer visitation is that it can help to show new players what is possible to make in-game.

For example, consider the ORCHESTRATION (p. ??) *My Singing Monsters* (2012), in which you assemble an orchestra of, you guessed it, singing monsters. The game doles out monsters rather slowly – it's a free-to-play game, after all – so in the first few hours it's difficult to understand how complex the resulting compositions can become. However, the game also lets you visit orchestras from other players. The top-rated compositions from around the world are complex and creative, and provide a spur to the player to keep playing despite their initial paltry orchestra.



Animal Crossing: New Horizons

Technically *Habbo Hotel* had launched player-owned rooms a few months before *Animal Crossing* released. But at that point in time, *Habbo* was a fairly small startup and its period of influence was later.

Structural engineering

The building system involves a physics engine and the player's creation must actually stay up (or complete some other objective).

Examples

Playing with *Building blocks* is a perennial children's entertainment, and various physical games involve trying to set up, remove, or balance blocks without destroying the whole structure. These often shade into dexterity games, like *Jenga*. *Tokyo Highway* is an interesting mix of city-builder-like mechanics and dexterity challenge.

Pontifex (2001), which later evolved into the *Bridge Constructor* series, challenged the player to build bridges strong enough to hold traffic.

World of Goo (2008) was a notable physics-building game and one of the biggest early hits of the 2010s "indie boom". Little goo balls are both the LEMMINGS (p. 128) that you're attempting to shepherd to their destination by building towers and bridges... and the building material out of which those towers and bridges are built. This duality creates a natural challenge: using a lot of goos makes for solid structures but fewer goos saved at the end, whereas using few goos makes unstable structures that whip around wildly but, when successful, save more goos.

Analysis

Building something that actually stays upright and functions for some purpose is a very intuitive and human kind of challenge. Even without any external motivation, people will try to build towers of random objects as a kind of SELF-CHALLENGE (p. 39).

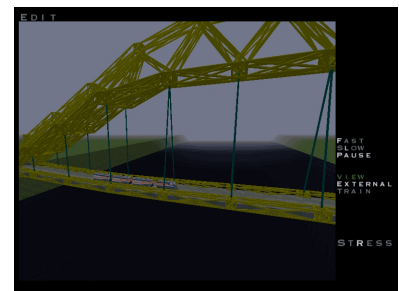
Building upward inherently has a kind of PRESS-YOUR-LUCK (p. 248) mechanic, and the same sort of emotional arcs apply: even if progress is steady, setbacks are immediate and brutal.



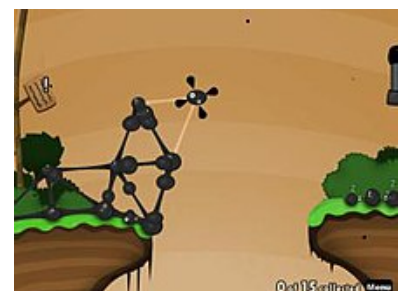
HABA building blocks



Tokyo Highway



Pontifex



World of Goo

Demolition

The player gets to destroy buildings, cities, or other structures.

Examples

Although *SimCity* (1989) is mentioned many times in this book, I haven't yet mentioned one of its most famous features: the ability to destroy your own city with earthquakes, tornadoes, and even Godzilla.

Demolition in games sometimes goes hand-in-hand with BALLISTICS (p. 135), as in *Boom Blox* (2008), *Crush the Castle* (2009), or *Angry Birds* (2009). Rather than have complete control over the destruction, you can only launch an initial projectile, and hope that physics and CHAIN REACTIONS (p. 134) do the rest.

Analysis

If you have the ability to build something in your game, consider putting in the ability to purposely destroy it. After all, half the fun of building sandcastles as a child was watching their slow destruction by the tide – or participating more directly.

This was almost certainly part of the slow viral popularity of *SimCity*; it was almost universally mentioned in early descriptions of the game, and the disasters were prominently featured in the box art. (The initial box featuring Godzilla was later amended to feature a tornado due to trademark concerns.)



Godzilla destroys a player-built city in *SimCity*



Boom Blox was intended as a non-violent alternative to combat games, according to its director, Steven Spielberg. Yes, *that* Steven Spielberg.



Crush the Castle

Part VI

Art and music

Picture revelation

The player's actions gradually reveal a picture.

Examples

Connect-the-dots puzzles are a classic example of a picture-revelation puzzle; *Jigsaw puzzles* are another.

Nonograms (also known as *Oekaki Logic*, *Picross*, *Pic-a-Pix*, or *Paint-by-numbers*) are CONSTRAINT SATISFACTION (p. 34) puzzles that, when complete, reveal 1-bit pixel art. Since the global success of these puzzles circa 2000, many similar puzzles have been developed; I particularly like *Link-a-Pix* for how it creates multi-colored pixel art without requiring the player to engage with any sort of "color picker".

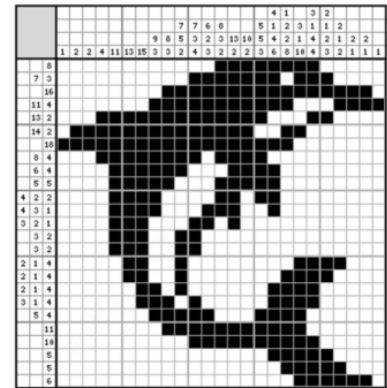
For a non-pixel art example, *Vertex* from the *New York Times* puzzle site is a *Hashi*-like network puzzle where the solution makes a "low-poly"-style picture made of triangles.

Picture revelation is also a common inclusion in QIX-like ENCIRCLEMENT (p. 27) games like *Gals Panic* where the player slices pieces off the gameboard and reveals a picture underneath.

Analysis

Picture revelation can add a level of motivation to otherwise rather abstract puzzle or even action games. One complaint I've heard about *Sudoku* is "When I solve one, it doesn't feel like I've actually accomplished anything. It's just a grid of random numbers." Creating a picture doesn't have that arbitrary feeling.

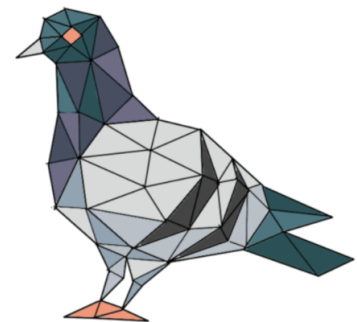
On the other hand, basing puzzles on pictures means that someone has to make those pictures; unless you're running a live-service game with an artist on staff, the player is eventually going to run out of puzzles. Meanwhile, something like *Sudoku* can be generated endlessly without human involvement.



A Nonogram puzzle



Link-a-pix



Vertex

Photography

The player needs to take photographs of things (rather than, say, collect them or kill them).

Examples

Photography is an obvious drop-in replacement for shooting (we even use the same verb for the action), and often serves to adapt conventional “shooter” genres for non-violent gameplay.

The *Pokémon Snap* games take the basic structure of a rail shooter (e.g. *Time Crisis* or *House of the Dead*) and adapt that into a non-violent game where you snap pictures of Pokémon.

The *Touhou Project danmaku* shoot-em-ups *Shoot the Bullet* and *Double Spoiler* replace the shooting aspect of the genre with photography; instead of trying to destroy the target, the protagonist is a tabloid journalist trying to get photographs of it.

Photography often also involves scavenger-hunt gameplay, in contexts where finding the item is more important than GATHERING (p. 152) it. *TOEM* (2021) is fundamentally a scavenger hunt game, where the photography theme provides an occasional twist through puzzles where you have to find special vantage points or set up a tripod to take a picture remotely. In another context, photography is part of the CRAFTING (p. 163) in *Dark Cloud 2*. Crafting requires not just material prerequisites but also Ideas (e.g. crafting “Dragon Shoes” requires not just materials but the Idea of a dragon), and Ideas are obtained by photographing relevant objects in the game world (e.g. a dragon statue).

I don’t include in this section photographs that other characters have taken; but cover them partially in DIARIES (p. 78).

Analysis

As I note in TILING PUZZLES (p. 190): any time the player is making any sort of artistic composition, make it easy to share it on social media. It’s free advertising!



New Pokémon Snap (2021)



TOEM

Playing musical instruments

The player's inputs are mapped to notes on an in-game instrument.

Examples

This is, of course, the basic pattern of the *Guitar Hero* and *Rock Band* series of games, and I think everything I would need to say about it here I already covered in TIMED INPUTS (p. 108).

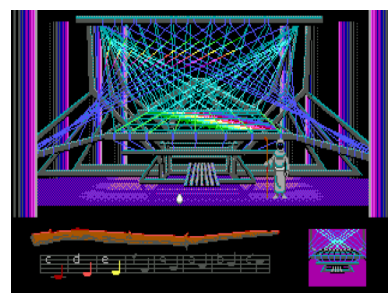
Playing music is a common way of expressing magic spells in games, for example in *LOOM* (1990) or *The Legend of Zelda: Ocarina of Time* (1998). *LOOM* had an interesting twist halfway towards SPELL-CRAFTING (p. 206) in that playing the spell melody backwards resulted in a spell with the effect reversed (e.g. OPEN becomes CLOSE, SLEEP becomes WAKE).

In *Genshin Impact*, playing the zither was originally added as part of a rhythm game (p. 108) in a special event, but their use isn't restricted to those songs; players can also just play whatever notes they please. This illustrates another function of in-game instruments: to show off for other players or for social media/streaming.

ODDADA is a free-form music sandbox game – a sort of “music box simulator” – letting the player play and modify a variety of whimsical musical instruments.



The Legend of Zelda: Ocarina of Time



LOOM



Genshin Impact

Music composition and orchestration

Examples

Music composition is not a *common* feature in games, because music composition is rather difficult and only a fraction of players will be able to do it at any but the most elementary level. It sometimes appears as a side activity in games with level editors or map/base-building; *Super Mario Maker* and *Stardew Valley* have blocks that play notes or drumbeats, allowing clever players to set up levels that play familiar songs.

Even where the player does not personally write or perform the melodies, they may be able to put together larger pieces of music from pre-composed pieces; I'll call this "orchestration". This is the central mechanic of *My Singing Monsters* (2012), in which you place monsters into an orchestra, where they periodically sing short melodies which combine into more complex works. Orchestration also plays a role in *Legend of Mana* (1999), where you choose the instrumentation of different songs. Even *Proteus* (2012) is an implicit orchestration game; most objects in the world contribute one instrument to the soundtrack, and where you stand determines what aspects of the soundtrack you can hear.



A composition in *Stardew Valley*



My Singing Monsters

Part VII

Maps and travel

Breadcrumbs

Rewards such as coins are laid out in a path for the player to COLLECT (p. 30), usually to incentivize the player to follow that path.

Examples

A particularly worthwhile example to study is *Donkey Kong Country II*. Part of the game's contract with the player is that banana rewards are never misleading – they never lead the player to their death or punishment. They can thus reliably be used to lead the player to secrets, or to encourage the player to make blind leaps that they otherwise would avoid making.

A game that relies very heavily on lines of rewards is *Katamari Damacy* (2004). In this game, the player is pushing a growing, infinitely-sticky ball; anything smaller than the ball will stick to the ball and make it larger. The game intentionally lays out similar-sized objects in lines, and following these lines is the best path towards growing larger more quickly.

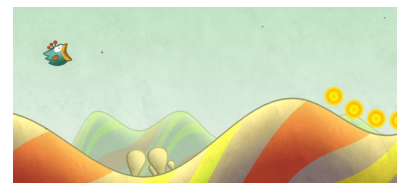
Along similar lines, *Tiny Wings* (2011) uses reward lines to solve a Buridan's Ass problem. The player character slides down hills to build up flying speed, but the hills are all pretty similar; what incentivizes me to slide down this one rather than that one? Well, some hills have coins on them, and that's enough to help me choose.



Donkey Kong Country 2



Katamari Damacy REROLL



Tiny Wings

BURIDAN'S ASS:

A dilemma in which someone has no means to decide between two alternatives, illustrated by a donkey placed equidistant from two piles of hay. In the absence of any reason to choose one over the other, the donkey starves.

Mazes

The player is offered a complicated level with many dead ends and cycles, and few valid routes through.

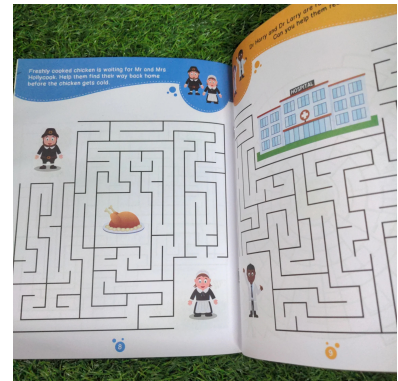
Examples

By “maze” I don’t mean “maze games” in the sense of “Pac-Man clones”; many of these are really AREA CONTROL (p. 25) games. What I mean is “mazes” of the sort that you used to do in childhood activity books.

Mazes were extremely common in 1980s games, often as filler to pad out the playtime of a text or graphical adventure game. The maze description in *Colossal Cave Adventure* – “YOU ARE IN A MAZE OF TWISTY LITTLE PASSAGES, ALL ALIKE” – has become a well-known catchphrase. The first-person mazes in Sierra adventure games like *King’s Quest V* (1990) or *Quest for Glory II* (1990) are particularly infamous because your point-of-view changes with every step.

First-person mazes like this were also common in RPGs like *Wizardry*, although I give them a pass retrospectively because dungeon crawling is what these games were *about*, whereas in the Sierra games they are more obviously filler. Due to the enormous popularity of *Wizardry* games in Japan, first-person mazes survived much longer there. On the other hand, in these games, the dungeons are maze-like but not always MAZES in the way I’m using the term here, and the auto-mapping in modern *Wizardry*-inspired RPGs like *Etrian Odyssey* (2007) make traversing them a very different experience.

Another instantiation of the maze pattern is “door mazes”, in which you enter one door and come out another door, and the connections between these doors do not always make any geometric sense. There are several of these in *Indiana Jones and the Fate of Atlantis*, including a frustratingly long one right before reaching Atlantis itself.



This is the kind of “maze” I mean here.



Quest for Glory II



Etrian Odyssey used the second screen of the Nintendo DS for auto-mapping, making it a far kinder experience than *Quest for Glory II* above.

Analysis

I should note here that, in the 1980s and early 1990s, it was basically assumed that computer gamers would keep a notebook and pencil handy. Games didn't usually have auto-mapping, and would not necessarily repeat important clues, so you really did have to write things down if you didn't want to approach the game by pure trial-and-error.

Remember that back then there were clearer lines between computer gaming and console gaming. Computers were things that lived on desks, so it wasn't unreasonable for a designer to assume that the gamer had paper and pens and a hard surface to write on. Designers can't assume that anymore, and therefore typically don't include many puzzles that the player can't keep in entirely in their head.

Mazes in computer games are thus a kind of artifact that's largely died out: a computer-aided *paper* puzzle where you're working out something on paper, based on clues the computer is giving you. I still get a nostalgic satisfaction from game puzzles sufficiently complex that I have to reach for a notebook.

That defense aside, I think it died out for good reasons. One crucial reason is that mazes offer little in the way of intermediate rewards. The player might be stuck in a maze for hours, but only at the end of it do they feel any rush of satisfaction. Giving a player no reward or sense of accomplishment for (potentially) hours is an invitation for the player to quit the game forever.

It became fairly common, especially in RPG "dungeons", to mitigate this by putting a treasure chest at the end of major dead ends, so that the player doesn't feel their time was wasted. However, as soon as you do this, the maze isn't exactly a maze anymore. Because the player doesn't want to miss out on potentially powerful equipment, the player's goal becomes going down *every* path rather than finding the one *right* path. Traversing a maze in its entirety is barely a puzzle (there are few decisions to make, just try every route in turn), so I hesitate to classify these as mazes.



Final Fantasy IX fairly reliably places treasure chests in dead ends. Because **EQUIPMENT-BASED ABILITIES** (p. 178) are so crucial in this game and you don't want to miss a single piece, this incentivizes *delaying* rather than *seeking* the solution of the maze, until all paths have been completely traversed.

Constrained traversal

The player needs to traverse a level under a special, unusual condition, like not retracing their steps, not touching the ground, touching every tile exactly once, etc.

Examples

A classic example of this is an *Eulerian path* puzzle, where the player is given (in the abstract) a graph of vertices and edges between them, and must traverse all the edges exactly once. (Think of those “Trace this figure without lifting your pen” puzzles.) Similarly, a *Hamiltonian path* puzzle requires you to visit all vertices exactly once.

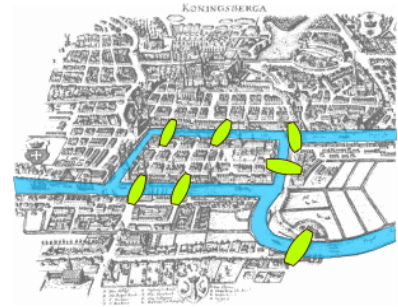
Ordinary MAZES (p. 96) can also import some of the challenge of these puzzles, by implementing one-time elements. Crumbling floors, which allow the player to walk on them once but crumble once they have passed, are prototypical examples.

Another kind of constrained traversal puzzle are ice mazes of the sort common in *Zelda* games. Link cannot stop on ice until he hits an obstacle, and therefore the position of obstacles on an ice field sets up a maze.

Analysis

An important use of this pattern is to force the player to look at familiar levels through a new lens (similar to VARIANT FORMS, p. 144).

For example, in *Final Fantasy V* (1992), you traverse the world map using a variety of VEHICLES (p. 113) of increasing power: first on foot, then using chocobos, then using ships and airships. Each of these puts different constraints on which squares you can move into; chocobos cannot cross mountains, but rivers are accessible to them. At a late point in the game, when you already have airships and thus full freedom of traversal, an NPC challenges you to circumnavigate the world using only a chocobo. These are mostly places where you’ve been before, but because of the unique constraints of chocobo travel, the path that you have to take through them is novel.



In 1736, Leonhard Euler challenged himself to walk across all seven bridges in Königsberg using each bridge only once. Realizing that this was impossible laid the foundation for what we now call Graph Theory.



Indiana Jones and the Fate of Atlantis (1992) had a clever variant of the crumbling floor. When Indy steps onto a tile, the *next* tile crumbles; he cannot step twice in the same direction.



Final Fantasy V

Parallel worlds

The player navigates two or more variations on the same world.

Examples

A classic example of this is *The Legend of Zelda: A Link to the Past* (1991), in which the player can warp between a “light” and “dark” version of the same world map. Many puzzles require warping between the two worlds. *Metroid Prime II: Echoes* (2004) has a very similar structure.

An earlier example might be the double maze in *Goonies II* (1987), where the maze has both a “front” and “back” side and traversing it involves transiting between them. However, in this game the two sides are not consistently differentiated graphically/thematically, which can be rather disorienting. *A Link to the Past* is rather more legible in this regard, because the two worlds have consistently different tilesets, enemies, and music.

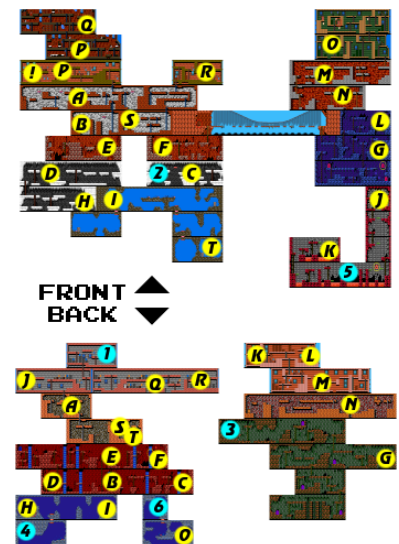
Many TIME TRAVEL (p. 71) games like *Day of the Tentacle* (1993) or *Chrono Trigger* (1996) could also count here, with an added twist that actions in “past” worlds cause changes in “future” ones.

Analysis

We can view parallel worlds as a continuum, between games where few locations are shared and the worlds are largely separate maps, to ones where only a few elements change between the worlds. Even powers like changing the seasons in *The Legend of Zelda: Oracle of Seasons* (2001) can be seen as creating parallel worlds; it’s not themed as the player traveling between worlds, but there’s a functional similarity: the player activates some power, and many elements of the world change all at once. Many parallel world games land somewhere in the middle: sharing many aspects of the map layout, having a few true differences between to set up puzzles, and handling most of the differences as purely visual theming (e.g. different tilesets or textures).



The “Light World” and “Dark World” versions of Kakariko Village in *The Legend of Zelda: A Link to the Past*



The double map of *Goonies II*, courtesy strategywiki.org

Delivery

The player carries mail, parcels, or a passenger to a destination.

Examples

This is a fairly common pattern in games, making it hard to choose any one as an example.

One important distinction is time-sensitivity. In *Final Fantasy IX*, for example, the player is free to deliver “Mognet” mail at their leisure; moogles are happy to receive their mail regardless of whether you deliver it immediately or carry it around for 40 hours.

On the other hand, the delivery activities in the *Crazy Taxi* or *Grand Theft Auto* series are time-sensitive: you have to taxi the passenger to their destination, get the patient safely to the hospital, or get to the scene of the fire as quickly as possible. They thus also have elements of RACING (p. 103).

Analysis

One useful consequence of a delivery mini-game is that it incentivizes the player to traverse and re-traverse the world, especially regions where they might not spend much time otherwise, or regions they would experience once during the main quest but wouldn’t usually return.

Delivery quests are also relatively easy to procedurally generate, but don’t (to me at least) feel artificial. I would get annoyed at doing 200 procedurally-generated fetch quests in a row, but I’m perfectly happy moonlighting as a taxi driver or pizza boy in *Grand Theft Auto* games for hours. I think this is another instance where *institutionalization*, as mentioned on page 21, works to give a thematic backbone to a repeated activity. Outside of institutionalization, fetch quests can feel like an imposition (“Since when am I everyone’s mailman?!?”); this objection is somewhat nullified when I’m actually their mailman. I knew what I was getting into when I signed up!



Crazy Taxi

A downside of delivery is that it offers relatively little player agency. We can see this in *Cloudpunk* (2020), in which you play a delivery driver in a cyberpunk dystopia. The player only makes one delivery at a time, and has no choice about what deliveries to take. Most of the game is traveling to the next navigation beacon; the player has few decisions to make, and the few that are offered make little difference to the gameplay or plot.

Even giving the player multiple deliveries at once can offer some minor strategic decision-making: it sets up the classic *Traveling Salesman* problem about how best to optimize a route so that it hits all the delivery points.

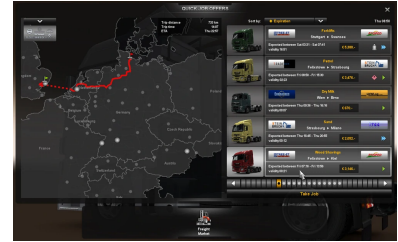
Another alternative that allows some agency is to give the player their choice of jobs/routes. In the *Euro Truck Simulator* series, you are not just a delivery driver following someone else's instructions, you're also the business owner choosing what contracts to take on in the first place. While the moment-to-moment gameplay is similar to *Cloudpunk*, the player has more high-level agency to author their journey.

Challenge

Delivery mechanics are a sort of "A-to-B" gameplay, but rarely do they intersect with "A-to-B" INDIRECT CONTROL patterns like LEMMINGS (p. 128) or RICOCHET (p. 136). Design a game around delivery, but where the player controls the carrier(s) only indirectly.



Cloudpunk



Euro Truck Simulator 2

Trading

Buy low in one area, travel to another area, sell high.

Examples

Star Trader (1974) was (to my knowledge) the first videogame implementation of this, and *Drug Wars* (1984) was probably the first big hit. Some other games primarily built around this mechanic are *Gazillionaire* (1994) and *Tradewinds* (2004).

Elite (1984), *Sid Meier's Pirates!* (1987), and *Wing Commander: Privateer* (1993) are examples of trading in multi-mechanic open-world games; trading is just one of the ways the player can approach the game.

For an interesting example of a trading subsystem in a completely different genre, the text adventure *80 Days* has a randomized market mechanic that incentivizes the player to take different routes each playthrough; I talk about this more in *RANDOMIZERS* (p. 252).

Analysis

Trading provides much of the same gameplay consequence as *DELIVERY* (p. 100) and *RACING* (p. 103), in that it incentivizes the player to visit and re-visit places in the world. Unlike delivery, however, the player is given more freedom about where to go next and why.

It's important that the values of goods fluctuate somewhat, so that the player is nudged away from simply doing the same routes over and over. This might be random, or be based on an economic simulation (so that over-delivering the same goods too many times temporarily depresses the local market for it).

Trading games usually also have the possibility of loss in transit – having a ship be boarded or lost, having a customs inspector seize *CONTRABAND* (p. 173), etc. – to add an element of risk. Without this, the player has less of a decision to make in each area, because there's little need to balance the risk/reward of carrying expensive cargo.



Sid Meier's Pirates!



Elite

A relatively shallow simulation (e.g., simple oscillations and some random events) may serve better than a deep, realistic simulation; you don't really want prices to soar or crash or stagnate due to market forces outside of the player's comprehension.

This calls back to Sid Meier's maxim, "The player should be the one having the fun." Rich simulations are fun to program, but can result in limited or unpredictable gameplay. The player only sees the results of the simulation, and can't necessarily reason about how those results came to be, or act in a way to promote/avoid them in the future.

Racing

The player attempts to reach a destination before a timer times out, or before a competing player/NPC reaches the destination first.

Examples

Humans have probably been racing at least since we stood up on two feet, so like *GROUND AVOIDANCE* (p. 46) it's a game pattern that barely needs to be explained.

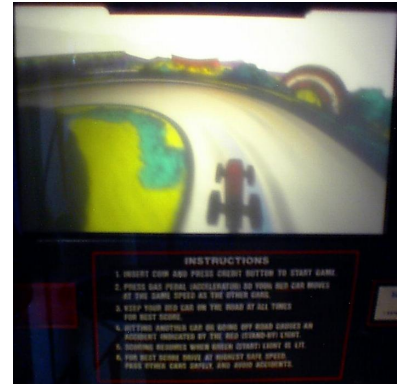
Race games (e.g. *Pachisi* and *Backgammon*) are also the among the oldest genres of boardgames; “all cultures that have games at all have race games.”²

Unsurprisingly, racing has also always been a stalwart of electronic entertainment. Even before digital arcade games, racing was a staple of coin-op electromechanical entertainments like *Drive Mobile* (1941) and *Speedway* (1969).

Some examples of racing in other genres of games include *Super Mario 64* (1996) where you race a Koopa through an already-familiar level, and *A Short Hike* (2019) where you race an NPC to prominent locations like the lighthouse.

Analysis

Aside from its usage in actual racing games, the inclusion of racing can be a way to reward the player for mastering the level layout. It can thus play a similar role to *DELIVERY* (p. 100) – really, the timed deliveries in that section are races by another name; it's just a matter of theming.



Speedway

² Parlett, David. 1999. *The Oxford History of Board Games*. Oxford: Oxford University Press

Shortcuts

The player discovers or opens up a quicker route between two places.

Examples

Opening up shortcuts is particularly important in Metroidvania-style exploration games where backtracking is expected, so that players do not have to re-tackle a challenge every time they re-explore an area.

They are also important in games where time is a limiting factor, like RACING (p. 103), DELIVERY (p. 100), or TIME LOOP (p. 72) games. For example, in time loop game *Outer Wilds* (2022), exploring planets fully requires returning to them multiple times. Exploring areas thoroughly reveals shortcuts you would never manage to find by chance, letting you explore further in future loops.

Analysis

Keep shortcuts in mind when designing exploration games without combat. In a combat-oriented platformers like *Castlevania: Symphony of the Night*, easier re-traversal comes “for free”, as your growing combat power trivializes previously-difficult enemies.

But say you’re making an exploration platformer without combat, and replace combat with precision platforming full of insta-kill spikes, like *You Have to Win the Game* (2012) or *Under the Moon* (2013). Games in this mini-genre (which I call “masovianas”) tend to fall prey to the same issue: unlike in combat-centric games, there’s no automatic mitigation of difficulty when re-exploring an area.

The first time the player finally gets through your deadly platforming gauntlet, it’s a huge rush. But then, later, if they’re just exploring and scouring the area for secrets, they have to go through the spike gauntlet again and again. No matter how well-designed your spike gauntlet is, the player won’t get the same rush the fifth time. It’s particularly important that you design these levels so that shortcuts open up, so that the player can bypass completed challenges before they become stale through repetition.



Going through this waterfall in *Mario Kart 8* serves as a shortcut.



This ice mound in *Outer Wilds* contains a cave entrance that leads to an underground city. Finding this cave by exploring the planetary surface would be extremely unlikely, but once you’ve found the city by other means and explored it thoroughly, you’ll discover that one of the exits leads you here.



You Have to Win the Game

Not to pick on these games; these were small hobby games released for free! But I think they represent an interesting design experiment, and it’s worth digging into the problems they face because I think it’s a natural, but not inevitable, consequence of this genre mashup.

Fast travel

The character is given ways to move more quickly between points of interest.

Examples

Most open-world games have some means of fast travel, often not exactly *theming* it as anything, just letting you pause and choosing a previously-visited landmark.

Sometimes the fast travel network is built or expanded by the actions of the player. In *Stardew Valley*, for example, the mincart network is initially in disrepair, and some achievements reward you with portions of the network being repaired.

Spelunky (2008) is a “rogue-lite” platformer where every run features new, procedurally-generated levels, and fast-travel upgrades are the only form of permanent META-PROGRESSION (p. 259). Rather than start from level 1-1 in every run, the player can open up tunnels to 2-1, 3-1, etc. by completing increasingly difficult challenges.

Analysis

This is a minor pattern that is usually implemented without much fanfare, but I mostly wanted to highlight *Stardew*- and *Spelunky*-style fast travel expansions as something you can reward the player with, in case you’re running out of novel ability upgrades and new content.

It’s also worth pointing out the way that *Spelunky* doesn’t just open up the tunnel to 2-1 upon *reaching* that level. Instead, Tunnel Man demands a significantly harder challenge (for 2-1, managing to acquire \$10k in the first four levels). This means that the player can’t simply luck out once and bypass an area; it’s only once they’ve demonstrated greater mastery of the area that they can bypass it. This strikes an elegant balance between the player never again returning to an area once they’ve beaten it once, and getting bored of an area because they’ve played it too many times.



The Elder Scrolls: Morrowind (2002) had a variety of fast travel systems, from boats to spells, but the most memorable were these giant insectoid Silt Striders.



Spelunky

Shortcut worlds

The normal world and a parallel world are connected at different scales, so that travel in the parallel world is effectively a **SHORTCUT** (p. 104) or **FAST TRAVEL** (p. 105).

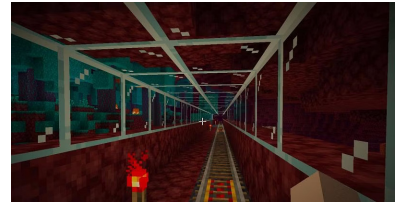
Examples

I'm sure this is not the only example, but the only one coming to mind at the moment is the "Nether" dimension in *Minecraft*. The Nether is scaled at a 1:8 ratio with the normal "Overworld" dimension, meaning that every block you travel in the Nether is equivalent to traveling 8 blocks in the Overworld. Since the player can build portals between the worlds, they can use the Nether to build a custom fast-travel network where they can effectively travel at 8x speed.

Analysis

I mention this mostly because of the ongoing debate whether **FAST TRAVEL** undermines the experience of exploring large open worlds, to point out that there are other patterns beyond instant teleportation that can achieve this.

Why not always travel in the Nether? Well, it is also a dangerous place of aggressive monsters and lava, giving the player a choice between slow-and-safer and fast-and-more-dangerous.



Some players even build train networks through the Nether dimension in *Minecraft*, allowing them to travel with maximum efficiency and relative safety.

Part VIII

Control and movement

Timed inputs

The player must time their input within a short duration, for example to dodge a projectile or hit a musical note.

Examples

This is the core mechanic of most rhythm games. *PaRappa the Rappa* (1996) was an early hit and has been argued to be the first “true” rhythm game. *Dance Dance Revolution* (1998) and *Guitar Hero* (2005) are also particularly notable for the way their innovative control schemes bring the game into the real world.

Timed inputs are also obviously an important aspect of many combat-oriented games (especially fighting games and beat-em-ups), and sometimes add mild action elements to low-action genres like RPGs, e.g. *Super Mario RPG* (1996) and *Legend of Dragoon* (1999). *Crypt of the Necrodancer* even requires the player to move on the beat.

Fishing mini-games often depend on timed inputs, where the player must respond quickly to (e.g.) a bobber dropping, or else lose the opportunity to catch the fish.

“Quick-time events” during otherwise non-interactive sequences obviously fall under this pattern as well, although like many gamers I don’t really enjoy them and don’t have much to say about them.

Analysis

Rhythm-based gameplay is a particularly rich quarry for action games looking to avoid violence. It’s also interesting to explore how timed inputs support non-violent (or less-violent) gameplay in combat games; while I wouldn’t consider the *Batman: Arkham* series to be non-violent, their block/parry/dodge-oriented fights do manage to capture Batman’s commitment to avoid killing. This is reminiscent of what we saw in *AVOIDANCE* (p. 43); while games like *Undertale* and *Ikaruga* do allow combat, the designers leaned into rich avoidance systems as a way to make non-violent playthroughs possible and compelling.



PaRappa the Rappa



For sheer variety, the *Rhythm Heaven* series is worth a mention for its relentlessly inventive sample of ways that animation can cue a timed input.

In this scene, the player needs to high-five a series of monkeys while riding the second hand of a monumental clock.

Combos and trick chaining

Particular sequences of inputs or actions, done one after the other, result in special effects, increased power, or scoring bonuses.

Examples

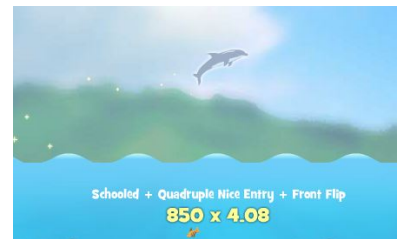
Many games have combos of some sort, especially fighting games, but this pattern probably sees its most complex development in skateboarding and trick-based “extreme sports” games like the *Tony Hawk’s Pro Skater* series. In these games, the player can do a variety of tricks, most of them associated with environmental features like rails and ramps, and the strategy of the game is to find routes through the environment that allow the composition of long and difficult performances.

For a completely different theming of similar mechanics, *Dolphin Olympics* (2006) puts the player in control of a playful dolphin, breaching from the water and doing flips.

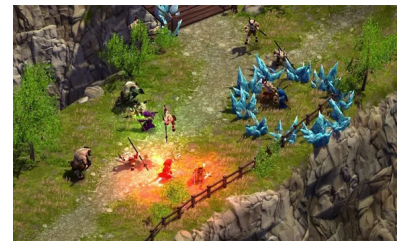
Combos can also build into a complex SPELLCRAFTING (p. 206) language like in *Magicka* (2011).



Tony Hawk's Underground 2



Dolphin Olympics



Magicka

Climbing

The player attempts to scale a steep slope, cliff, or tower.

Examples

Grow Home (2015) is a particularly notable climbing game, in which a robot who fell to earth grows and then climbs giant plants to get back up to orbit.

GIRP (2011) is a climbing game married to distinctly inappropriate controls (cf. *FUMBLECORE*, p. 120).

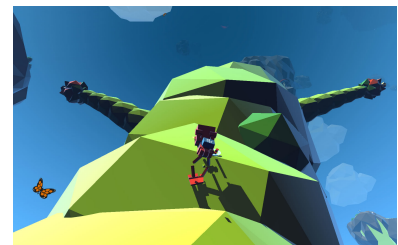
Climbing games often have stamina systems that limit either what distance you can move (e.g. *Breath of the Wild*) by climbing or how long you can hold on (e.g. *Shadow of the Colossus*). In *A Short Hike* (2019), stamina is the game's primary means of gating content (cf. *MAP EXPANSION*, p. 228). The game world centers around a mountain, and you are tasked with reaching the top. Most of the activities you do give you "feathers" that increase your stamina, letting you reach higher areas.

Analysis

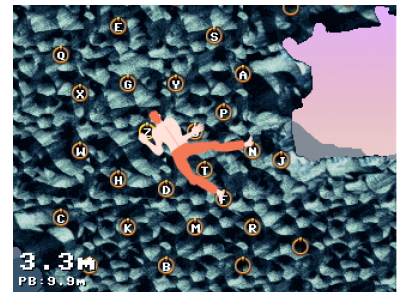
Games with a lot of climbing have a particular rhythm: methodical progression upward punctuated by sudden and catastrophic failure.

So be cautious about how much progress players lose when they fall; you might want to make a game that's as frustrating as *Getting Over It with Bennett Foddy* but be deliberate about this choice, don't make *Getting Over It* by accident.

Grow Home is an interesting study in this regard. While you do fall quite frequently in the game, this is mitigated somewhat by tools like the glider. They don't save you from falling, but they make falling fun; falling is not an immediate failure but an involuntary switch to another game mode! The game is also generous with teleporters so that even a fall to ground level isn't a permanent loss.



Grow Home began as an experiment by Ubisoft in handling climbing by inverse kinematics, for possible inclusion in games like *Assassin's Creed*. Playing the system demo, the developers realized that it was sufficiently fun to elaborate into a full-fledged game.



Girp



A Short Hike. In a marketplace where most games have you saving the world, it's a breath of fresh air to play a game where the central conflict is a lack of cellphone reception.

Flying, swimming, and gliding

The character flies in the air, glides, or swims in the sea.

Examples

2D flying games often use a control scheme similar to *Joust* (1982), in which the player presses a button to flap upwards, and otherwise floats downward. You may also be familiar with this control scheme from *Jetpack Joyride* (2011) or *Flappy Bird* (2013). It's rarer in 3D flying games; *A Short Hike* (2019) is one example.

Microsoft Flight Simulator, *AER: Memories of Old*, *Glyder*, *Abzû*, and *Subnautica* are examples of non-violent (or at least less-violent) 3D games that prominently feature flying, gliding, and swimming to support very different game goals.

Analysis

Military flight sims aside, 3D games centered around flying and swimming games are often non-violent, or at least less violent than their running- and jumping-oriented counterparts. I think it's just that it's difficult, when navigating with six degrees of freedom, to orient your and your enemies' exact locations, and flying itself represents a challenge on its own.

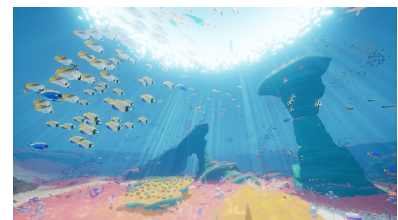
It's extremely common that 3D flying and swimming games have SLALOM (p. 114) challenges.



AER: Memories of Old



Joust



Abzû

Digging

The player removes earth or rock to create a passageway.

Examples

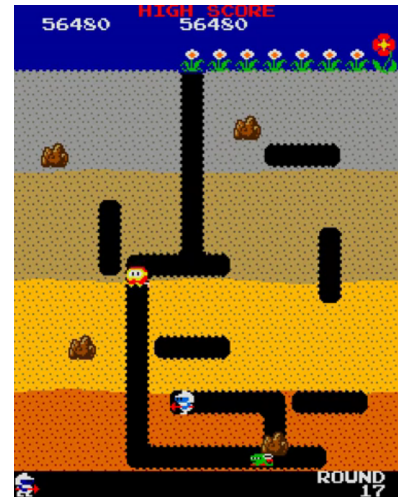
Several early arcade games featured digging, like *Dig Dug* and *Mr. Do!*

Related to digging is *mining*, where the player receives an item when removing earth/rock; this is of course the central pattern of the seminal *Minecraft*. *Steamworld Dig* is a notable 2D exploratory platformer centered around digging/mining.

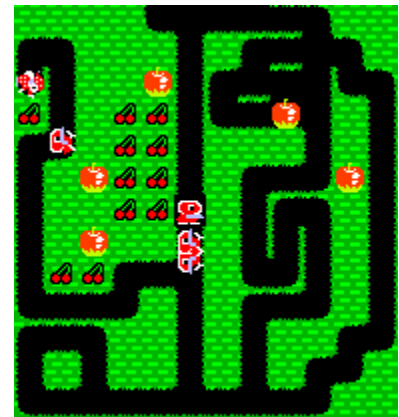
Analysis

One underappreciated aspect of digging is that it allows the player themselves create the maze in which they will subsequently play. This is exploited very elegantly in *Dig-Dug*. The player runs more quickly than enemies through open space, so when you stick to existing passages you can reliably outrun them. Meanwhile, the player digs more slowly than enemies run, but doing so expands the maze in which the player can subsequently run.

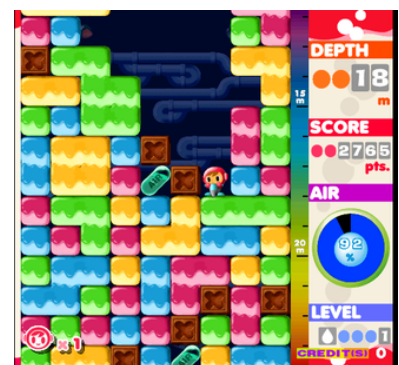
It's common in digging games that digging poses the danger of dislodging falling rocks and other objects. These are often used to set traps for enemies, but they can also pose threats to the player. It therefore sets up the possibility for *SELF-SABOTAGE* (p. 41) as the player succumbs to mine disasters entirely of their own making. *Mr. Driller* (1999) is entirely based around this; there are no enemies, and the primary danger is avoiding falling blocks that you yourself have dislodged.



Dig Dug



Mr. Do!



Mr. Driller

Vehicles

The player controls a vehicle or an animal mount.

When this control is temporary and the player switches between walking and driving, this can overlap with POWER-UPS (p. 141) and VARIANT FORMS (p. 144). But vehicles often have a very different control scheme than typical movement in games, which is why I put this pattern along with other CONTROL patterns.

Examples

Cars and other vehicles/mounts are frequent in games, from the stealable cars of *Grand Theft Auto*, to the military vehicles of *Halo*, to Epona in *Zelda*.

Vehicles are also used as a means of gating content, for example in *Final Fantasy* games. Since different vehicles have different possibilities of what they can travel on/over, and where they can dock/land, you can use vehicle upgrades to progressively expand the area that the player can visit.

Analysis

Although vehicles are a faster way to get around the world, they're often harder to control, with higher INERTIA (p. 118) and a wider turn radius than a human just running around. This is often mechanized as part of the challenge; in many games where you both run around and drive vehicles, only crashing into things with vehicles is penalized.

Keep in mind when adding vehicles that it can be very annoying for the player to dismount their vehicle, wander far away, and slog back to it, or forget where they left it entirely! Be sure to at least mark where it is on the map. If it's a horse or otherwise reasonably intelligent, have a way for the player to summon it like in the *Zelda* games or *Shadow of the Colossus*.



Some of the specialty vehicles in the *Grand Theft Auto* series have their own associated minigames.



If you get an airship midway through a *Final Fantasy* game, you've entered the more open-world part of the game. If you get an airship at the beginning of a *Final Fantasy* game, it is about to crash.



The Legend of Zelda: Twilight Princess (2006)

Slalom

The player must navigate through or around a series of posts or rings, often while traveling at speed.

Examples

Almost all skiing games have this; if I have to choose one in particular as an example, I have a soft spot for the skiing game in the rare, low-quality East German arcade cabinet *PolyPlay* (1985). The slalom poles are set up randomly, but frequently aligned in such a way that skiing straight down will succeed perfectly. Indeed, since turning slows the skier down, you can easily top the high score table by providing no user input whatsoever.

Driving or flying through rings (or other sorts of checkpoints) is common in series with vehicles, from *Grand Theft Auto* to *FarCry*, from *StarFox* to *Mario Kart*. These may be the point of the challenge such that missing one is a failure, or they may just give the player a speed boost.

Occasionally you see boost rings outside of a racing context. In *Smushi Come Home* (2022), swimming through a bubble ring gives you a speed boost, but the challenge isn't on any sort of timer. It's really just that Smushi swims slowly and there's not much to do here anyway, so swimming through bubble rings serves to add a bit of low-key fun. In a way it's like the coins in *Tiny Wings*, serving as BREADCRUMBS (p. 95) to give the player a reason to go this way rather than that, in a situation where there's not much other motivation for choosing.

I don't think the boost rings in the *Abzû* autoscroll segments have any real gameplay consequence either; I think they're just there as something to aim for while enjoying the pretty graphics and music.



The skiing game in the *PolyPlay* multi-game cabinet. If you want to play this rare cabinet, one is on display at the *Computerspielmuseum* in Berlin.



Despite my choosing *PolyPlay* and *Superman 64* (1999) to illustrate this pattern, I assure you this appears in some good games too.



Smushi Come Home

Wall-running

The character can run up walls and even across ceilings.

Examples

Wall-running is a staple of parkour games like *Mirror's Edge*, but also appears sometimes in more general platformers.

In limited segments of *Yoshi's Island* (1995), the player takes control of a tiny Baby Mario in anthill-like mazes, and freely runs up walls and ceilings.

This is sometimes combined with high-speed/HIGH-INERTIA gameplay, as if the player is going so fast that they can even run up walls; this is present in *Super Mario World* (1990) when Mario dashes with a cape, and *Super Mario Galaxy 2* (2010) when Yoshi eats a “dash pepper”.

To prevent the player from completely skipping normal gameplay by simply running over/under everything, games often limit this ability by level design. The *Mario* examples above require a curved/angled surface between the floor and wall. The Magnet-boots/Gravity Boots from the *Ratchet & Clank* series restrict the player to particular surfaces.

A clever example of wall-walking in the real world is seen in the children's boardgame *Snail Sprint*, in which the box is metallic and forms part of the gameboard, and the snail pieces are magnetic and climb up and over it.

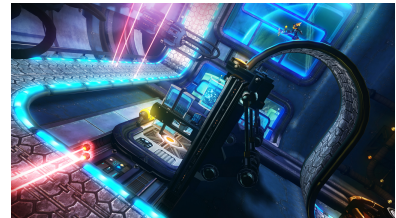
Wall-walking plays into the knowledge-based progression of *Outer Wilds* (2022) – walls lined with “gravity crystals” are traversible, but they're easy to miss if you're not looking for them, and you probably won't attempt to walk up those walls just for the fun of it.



Mirror's Edge



Eating a dash pepper in *Super Mario Galaxy 2* will allow Yoshi to run up this wall.



Ratchet & Clank Future: A Crack in Time



Snail Sprint

Rails and tracks

The player moves along a rail or track that determines their forward trajectory.

Examples

Rail-grinding is obviously an important part of skateboarding games, but also quite a lot of 3D platformers gained rail-grinding following the huge success of *Tony Hawk's Pro Skater* (1999): *Jet Set Radio* (2000), *Sonic* games since *Sonic Adventure 2* (2001), *Ratchet and Clank* (2002) and its descendants, the *Jak* games since *Jak II* (2003), etc.

Rail-grinding varies in how physically plausible it is – in most skateboarding games you need some momentum or a downward slope to grind for long. Sometimes you even have to balance. On the other hand, in *Ratchet and Clank*, rails are effectively a sort of automatic rollercoaster where you don't have to worry about propulsion. Instead, the challenge of rail-grinding is much like that of a lane-based endless runner (e.g. *Subway Surfers*): the rails are usually arranged in parallel “lanes” that the player must jump between to avoid oncoming obstacles and collect collectables.

Related to this are rollercoaster simulations. Often these are non-interactive, just letting the player experience a first-hand view of rollercoasters they designed, like in *RollerCoaster Tycoon 3*. Sometimes, however, there is more to do; in *3D Crazy Coaster* (1983) for the obscure Vectrex home console, which to my knowledge was the first rollercoaster simulation, required the player to lean into turns lest they fall off. This in turn shades into “mine cart” levels where the player is in partial control, like in *Super Mario RPG*.

It also shades into train simulations like *Densha de Go!* (1996), where the challenge is manipulating speed so that the train stops at the right place on the platform at the right time – ideally, within 30cm of the prescribed position and within 0.5 seconds of the prescribed time!



Tony Hawk's Pro Skater 1 & 2



Ratchet and Clank (2016)



RollerCoaster Tycoon 3



Densha de Go!

Trampolines and launchers

The player bounces or launches from something that shoots them further into the air than they can typically jump.

Examples

These are in many games, but they play a particularly important role in games where altitude is progress, like *Grow Home* and *A Short Hike*. In the latter, for example, you need to bring water to flowers, which makes them bloom and let you bounce off them to greater heights.

These are also particularly useful when a movement ability like GLIDING (p. 111) requires a high-altitude start, for example in... well, *Grow Home* and *A Short Hike*.

There is also a relationship here to BALLISTICS (p. 135) and RICOCHET (p. 136) games where the player has limited control of the main character (or object, as the case may be). In *Circus Circus, I am Level*, *Yoku's Island Express*, and even some parts of *Sonic* games, launchers and bumpers play a crucial role in moving the character around the level.



A Short Hike

High inertia/low friction

Rather than controlling the position of their avatar directly, the player controls the acceleration of an avatar with relatively higher inertia and lower friction – i.e., one that does not necessarily stop on a dime, but continues for a brief period in the direction they had been heading.

Examples

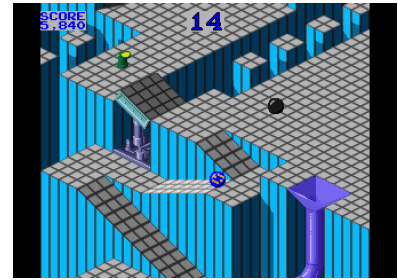
Almost all modern platformers have some degree of inertia in their controls. This was one of the innovations in *Super Mario Bros.* that made Mario feel so good to control, and even today Mario has higher inertia than most platformer protagonists. In this section, however, I'll mostly concentrate on games and levels with even higher inertia, to the point where the primary challenge of the game is keeping the character under control.

Marble Madness (1984) is a classic high-inertia arcade game, made visually intuitive by the fact that you're controlling a marble rather than a human or animal avatar. Note, if you've only played this on a home console or emulator, that the original arcade game was played with a trackball, which makes the control more intuitive as well.

Crystal Quest (1987) for Macintosh computers is an interesting example of controlling an action game with the mouse, leaning into the mouse's inherent indirectness of control by making it affect the acceleration rather than position of a high-inertia avatar.

Many platformer games have "ice levels", like World 4 in *Super Mario Bros. 2* or the Ice Caves in *Spelunky*, where the player character is challenged by surfaces with dramatically lower friction.

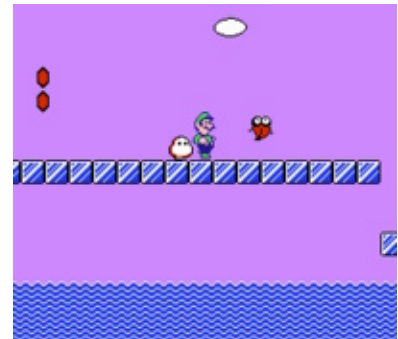
In some games, dashing is associated with higher inertia or even a temporary loss of control. In *Super Mario Galaxy 2* (2010), Yoshi gets a huge speed boost after eating hot peppers, but cannot stop and becomes difficult to steer. The Pegasus Boots in *Legend of Zelda: A Link to the Past* give Link a significant speed boost, but he can no longer stop quickly and cannot turn.



Marble Madness



Crystal Quest



Super Mario Bros. 2

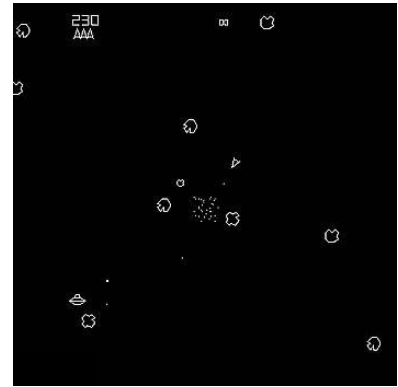
Analysis

High inertia and/or low friction is a classic way to challenge the player without necessarily adding “enemies” – the player’s worst enemy is their own incompetence and impatience (cf. *SELF-SABOTAGE*, p. 41). There are few enemies in *Asteroids* (1979), after all; most of the challenge is just not crashing your ship into the asteroids.

Since the central problem in games with very high inertia is controlling the protagonist at all, this pattern shades into *FUMBLECORE* (p. 120). However, I decided not to try to treat these patterns together. Having high inertia does not necessarily mean the control scheme is inappropriate; as I mentioned, *Marble Madness* has a completely reasonable control scheme for controlling a character who’s a marble.

Since this pattern is about the manipulation of acceleration, and gravity is a kind of acceleration, this also shades into *GRAVITY MANIPULATION* (p. 138). However, although they’re similar games, the physics and theming of *Marble Madness* and *Super Monkey Ball* are subtly different. In the latter, the player is tilting the entire world, whereas in the former only the main character is affected by the player’s inputs.

It’s worth pointing out that high-inertia characters are typically associated with third-person cameras. If you were controlling a high-inertia character in a first-person game, stopped pressing forward before a cliff, but nonetheless your character continued forward due to high inertia and fell of the cliff, it would be a frustrating and probably baffling experience. Inertia is really only intuitive if you can see the character moving from the outside, or if you are controlling something like a car that you expect to be high-inertia in the real world.



Asteroids

Fumblecore

The player has a relatively simple objective that is frustrated by a control scheme that ranges from unusual to wildly inappropriate for the task at hand.

Examples

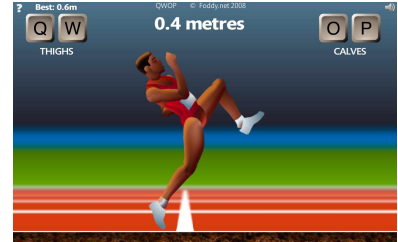
A designer who makes extensive use of this is Bennett Foddy, creator of *QWOP* (2008), *GIRP* (2011), and *Getting Over It* (2017). In *QWOP*, for example, the player only has to run a short footrace, but does so by controlling individual muscles in the player character's legs.

In the physical world, electromechanical *Crane games* are a fumblecore classic, with awkward controls and weak grip designed to minimize the number of toys actually caught. Similarly, *Kingyo-sukui* is a traditional Japanese festival game in which the player attempts to catch live goldfish in a paper net that tears very easily.

Analysis

This pattern is often frustrating to play, but fun to watch. This is especially important in a gaming economy that's increasingly centered around streaming; *Getting Over It*, for example, was precision-crafted to go viral in this environment, and did.

Octodad (2010) amplifies the hilarity of its control scheme (mouse movements control individual tentacles of the protagonist) with its theme, where you are an octopus masquerading as a human and doing your best to not reveal your secret – even to your own human wife and children. This plays a similar role to a “straight man” in a stand-up comedy duo, heightening the comedy by regularly reminding the player how social norms are being violated.



QWOP



The crane game in *Yakuza Zero* (2015)



Octodad: Dadliest Catch (2014)

Consensus controls

Multiple players attempt to control the same object, which is then moved by a majority vote or average among the players' inputs.

Examples

The *Dream Bobsleigh* event in *Mario & Sonic at the Olympic Games* is a straightforward example of this; each player is attempting to steer the bobsled simultaneously. If the players do not coordinate, they'll be working against each other and not steering the bobsled at all.

The two-player co-op mode in *Katamari Damacy* games also involve two players controlling one object. Usually, the sticky ball is controlled by one player using two joysticks (as if each joystick was one hand of the character pushing the ball). With two players, each is pushing one side of the ball, requiring careful coordination.

At a broader scale, this is also the main pattern in *Twitch Plays Pokémon*, in which spectators in a *Twitch* streaming service chatroom control the protagonist of a *Pokémon* game by majority vote. The interest of *Twitch Plays Pokémon* is in part watching the tug-of-war between a faction seeking to actually complete the game, and a troll faction seeking to make this completion impossible by (e.g.) continually causing the protagonist to fall down one-way cliffs.

Analysis

This is a great pattern for couch co-op party games, because it gets the players shouting to coordinate their movements. "Left!!!" "No, your other left!!!"

This was a feeling I deliberately targeted in a little game jam arcade game *Les Blocs de l'Est*, in which four players cooperate to maneuver a trampoline à la *Fire!* (1980) (see *GROUND AVOIDANCE*, p. 46). Getting hit does not harm or remove players from play, but (in this situation) something worse: it causes a "confusion" STATUS EFFECT (p. 146) that reverses their controls (and thus, unless they realize it, causes them to work against their teammates).



Mario & Sonic at the Olympic Games



Twitch Plays Pokémon

Simultaneous control

The player simultaneously controls multiple characters (often two), but separately (e.g. by using two joysticks), or by the characters receiving the same input but moving independently from one another.

Examples

This pattern appears in many “Döppelgänger” or “clone” puzzles, where the player’s input is sent to multiple characters, often rotated or mirrored (e.g., so that one goes east as the other goes west). To my knowledge, this pattern first appears in *Binary Land* (1983), in which mirrored penguins attempt to reach a heart simultaneously.

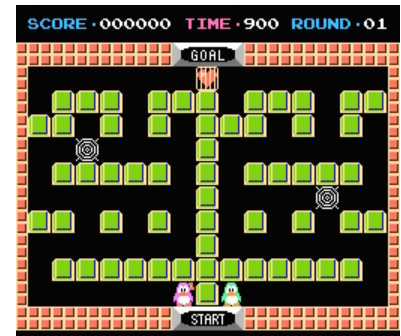
The goal is often to get the multiple characters to achieve something that cannot be achieved with one character, like standing on two pressure plates simultaneously. Also, it is frequent that the initial relative positions of the characters cannot achieve this goal, and you must maneuver one so that they cannot walk forward (e.g. towards a wall) in order to adjust their position relative to others.

This is a common trope in puzzle-oriented Nintendo first-party titles like the *Zelda* games, especially *Four Swords*. *Captain Toad: Treasure Tracker* also uses it quite a bit. *The Nightmare Cooperative* is an example of this pattern imposed on a rogue-like dungeon-crawler.

Mario and Luigi in the *Mario & Luigi* series of RPGs receive the same directional input, but jump independently: one jumps using the A button and the other B. This lets the player decouple the positions of the brothers by having one jump to a different altitude.

Analysis

Simultaneous movement requires the player to divide their attention and thus increases the inherent cognitive difficulty of any task. The aggressive enemies in *Binary Land* would be manageable if you were only protecting a single character, but it’s very challenging to divide your attention between *two* vulnerable characters, especially as they’re often quite far apart.



Binary Land



The Nightmare Cooperative

Blob movement

The player's inputs control a group of characters at once.

When the characters are moving independently, I call this **SIMULTANEOUS CONTROL** and cover it in the previous section; this pattern is more for when the group is moving as a single piece, a formation, or a flock.

Examples

This trope is named for the old dungeon-crawler RPGs like *Wizardry* and *Might and Magic*, where you were purported to be controlling a whole party, but everyone always exists in the same grid space. *Etrian Odyssey* (2007) and *Legend of Grimrock* (2012) are modern resurrections of the "blobber RPG" genre.

In action games, the GameBoy shoot-em-up *Mercenary Force* used blob movement. Instead of controlling a single character, the player controls a formation of multiple characters each of whom shoots in different directions. Hiring your party and choosing their formation for each level adds an additional strategic layer beyond ordinary shoot-em-up strategy.

Sometimes the player controls a somewhat chaotic flock of characters by (for example) controlling an invisible centerpoint around which the characters flock. In *Okhlos*, you play as an unruly, angry mob trying to take down the Ancient Greek gods.

This is a rare pattern in boardgames because of the difficulty of asking the player to move multiple pieces manually, but *Abalone* is an example. Pushing one piece automatically pushes all pieces in front of it (in a highly satisfying manner), with the goal of pushing other pieces off the playfield. Since it's important to keep your pieces packed together for protection, games feel like a sumo match between two amoebas.



Mercenary Force



Legend of Grimrock



Okhlos

Character abilities

The player can switch between characters that differ in abilities.

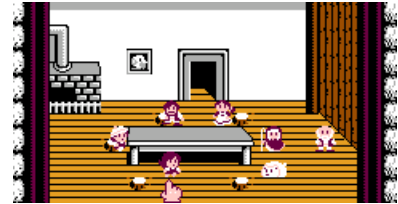
Examples

It's typical that game characters have multiple abilities – e.g. they jump *and* attack *and* push blocks – but sometimes games encapsulate abilities into different characters. This opens up the possibilities for more intricate puzzles, where the character doesn't just have to overcome an obstacle using an ability; instead, they have to prepare a safe path for the other characters without that ability.

An early example of this pattern is the exploration-platformer-RPG *Legacy of the Wizard* (1989). The creator originally envisioned having a multi-character BLOB (p. 123) à la Wizardry, but due to hardware limitations changed to a format where you chose one of five characters at the home base and send them individually into a complex maze-like dungeon. Each of these characters has different stats, jump height, and abilities (one can destroy blocks, another can push them, one is invulnerable, etc.). Since different characters can surmount different obstacles, and some items are only reachable by characters than cannot themselves use them, the player must make multiple delves using different characters to complete the game.

The Lost Vikings (1993) is probably the clearest archetype of a character-ability game. The player controls three characters with different abilities: one can jump and bash through walls, one has a sword and bow, and a third has a shield. The three characters are all present in the levels simultaneously, and levels are only complete when all three make it to safety, so the player must switch between characters frequently to open up routes for the others.

Sometimes the player does not have direct control over the other characters. *Ico* (2001) is an ESCORT (p. 127) game where the main character can run, jump, push boxes, and fight, and the escorted character can do none of these, but can open magical doors that the main character cannot. *Pikmin* (2001) is a MINION (p. 131) game with multiple species of minions, each with different abilities.



The character selection hub of *Legacy of the Wizard*. This was originally published as *Dragon Slayer IV: Draslay Family* (1987), but the Dragon Slayer games actually have relatively little to do with each other and span a variety of genres. This series is still around, actually, in the form of the *Legend of Heroes/Trails* RPGs.



The shield-bearer in *The Lost Vikings* blocks a laser to clear a path for the other characters.



I considered leading with the earlier example *Doki Doki Panic* (1987) (what later became *Super Mario Bros. 2*), but the characters' different abilities don't really interact in interesting ways. Mechanically, *Legacy of the Wizard* is a more interesting example of this pattern.

Part IX

Indirect control

Introduction

INDIRECT CONTROL has been a fixture in games since the 1990s: characters that are primarily controlled by AI algorithms, often simple ones, but that can be manipulated, given orders to, or controlled in a limited sense. The near-simultaneous in the early 1990s of so many indirect control games is no accident – one of the central questions designers were grappling with was “What can we do with these new-fangled *mice*?” Mice didn’t lead themselves to direct character control the way a joystick or directional pad did... but they *were* very useful selecting characters and setting destinations.

A prototypical example is *Lemmings*, a huge early-’90s hit in which the player attempts to guide little creatures with a stupid but predictable AI through a dangerous world. Various sources have cited *Lemmings* as an important progenitor of the real-time strategy (RTS) genre; the designer of *Warcraft: Orcs and Humans* said that the game was originally conceived as a mashup of *Lemmings* and *The Lost Vikings*.

Indirect control patterns can be found in thousands of games, so to split them up I’m going to differentiate them based on how control is split between the AI and the player.

- In some games, the AI-controlled characters are making fairly simplistic local decisions based largely on the environment. Lacking a pathfinding algorithm and a healthy sense of self-preservation, the characters will typically wander off, fall off cliffs, or pace endlessly back and forth, unless the environment is set up in such a way that the algorithm↔environment interaction happens to lead them to their destination. The main role the player has, therefore, is to make that so; I call this pattern LEMMINGS (p. 128) after its most iconic implementation.
- In some games, the AI is a little more sophisticated (e.g. it has pathfinding), and the emphasis is more on the tasks the characters are carrying out; I call this pattern MINIONS (p. 131).
- Where you’re guiding a smaller number (often one) of a more sophisticated and unpredictable AI-controlled character, I call that ESCORT (p. 127).
- In this section, I also include patterns related to physics games like BALLISTICS (p. 135) and RICOCHET (p. 136), in which an object (often a ball) is controlled by a physics simulation, and the player can guide this object only indirectly by launchers, paddles, tilting the playing surface, etc.

Escort

The player has to guide or accompany an AI-controlled character to a destination, and figure out how to manipulate their movement or remove obstacles.

Examples

Escort missions are everyone's least favorite memory of many action games (e.g. *GoldenEye 007*) due to the likelihood that the mission fails purely due to bad decisions on the part of the AI. However, the general design pattern still has many valid uses, particularly in more puzzle-oriented games where the companion character isn't regularly running into gunfire (e.g. *Ico*), or games in which the companion has some degree of invisibility and/or invulnerability (e.g. *The Last of Us*).

Ico (2001) is entirely based around an escort mechanic, but uses the mechanic to set up intricate environmental puzzles, where the younger and more agile protagonist has to manipulate the level so that the older and less agile companion can be led to the exit. It's worth pointing out here the mastery with which *Ico* foregrounds this mechanic through its emotive art and animation: the way the characters gesture and call to each other, the way that the slower character stumbles slightly when they run holding hands, and similar details that emphasize the relationship between these characters as well as their differences.

You can use this pattern without designing the entire game around it, of course. *Chrono Trigger* (1995) has a single sequence where you need to reunite a cat with its owner by leading it through the town fair.

Many puzzle-platformers and other puzzle-oriented games have puzzles where you have to manipulate an enemy in a certain way (often "kiting" them or having them mirror your movements) to bring you a key or powerup.



GoldenEye 007



Ico



The Last of Us

Lemmings

One or more NPCs (often very many small ones) are guided through an environment by a predictable movement algorithm, and the player guides them to a goal by manipulating the environment.

Examples

The earliest game I can find utilizing this pattern is *Gyromite* (1985) Game B, in which you use the NES R.O.B. peripheral to open and close gates to guide a sleepwalking professor safely through a level. Other sleepwalker games include *Sleepwalker/Eek the Cat* (1993) and *Back to Bed* (2014).

However, the classic game illustrating this pattern is *Lemmings* (1991), to the point where all games in this genre are sometimes just called “Lemmings-clones” even when they’re quite different games. The titular lemmings follow a simple walking algorithm, and unaided will generally walk off cliffs to their doom. Your task is to lead the lemmings to safety by giving them limited-use abilities and tools that change the environment, and thus manipulate their walking algorithm.

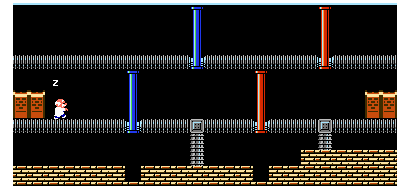
ChuChu Rocket! (1999) is a similar example from a top-down perspective, where the player drops arrows to manipulate the walking algorithm of little mice.

In *Soreyuke!! Amida-kun* (1990), the NPC descends an Amidakuji diagram. *Amidakuji* (also known as *Ghost Leg Lottery*) is a Japanese children’s activity in which the player follows a path downward through a ladder-like diagram according to a particular algorithm. (You can see another Amidakuji diagram in the AREA CONTROL (p. 25) chapter; most enemies in *Amidar* use the Amidakuji algorithm to move.) In *Amida-kun*, the player actually controls a mobile line segment within the diagram, and by moving around leads the NPC to safety or death.

Splitter Critters (2017) involves more drastic changes to the levels: slicing the levels into pieces and sliding those pieces around to create paths that the NPC critters can safely traverse.



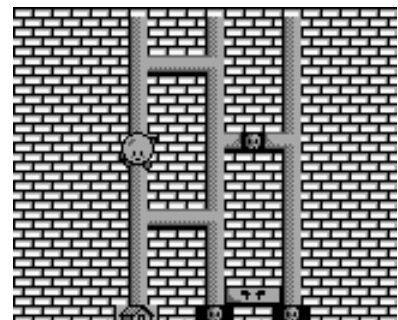
Lemmings began life as part of *Worms*-like player-vs-player combat game. A programmer implemented an algorithm for a ground-hugging missile, but it was so fun to just watch and manipulate the missile that they rethemed it as a “walker” and designed a whole game around it.



Gyromite Game B



ChuChu Rocket!



Sureyuke!! Amida-kun

Many of the puzzles in *The Incredible Machine* series involve setting up the environment to lead a ball or character to a goal. Since many of these elements are controlled by a physics simulation, however, the results can be more chaotic than in Lemmings-like games (cf. RICOCHET, p. 136).

Analysis

This pattern often provides a game with a lot of intricate visual interest; it's fun just to watch the lemmings move through the level.

One downside of this pattern, at least the way it was used in these classic games, is that the iteration cycle between ideation and execution is often quite slow. The player examines the level and hypothesizes a solution, but finding out if that solution actually works requires waiting for the NPCs to gradually work their way through the level. If the solution doesn't work at any point, the player typically has to restart at the beginning. Even with the innovation of a fast-forward button, this was a fairly slow and methodical genre, and while *Lemmings* was a huge hit, the genre as a whole never took off compared to its fast-paced cousin the RTS.

Challenge

Let's reconsider the design of *Lemmings*-style games in an attempt to eliminate the slow ideation/execution loop. Not just band-aiding it with fast-forward or rewind, but eliminating the need for that loop entirely.

How would we make it so that you only ever look forward from your current state (never rewinding or restarting). Could we make a *Lemmings* game with ratcheted progress, where there's no failure condition or significant setbacks? What if the lemmings spawned infinitely, were immortal, could be brought back to life, or just reproduced so quickly their population is never really in danger? What would your level goal be? What else in the game would have to change due to these changes?

Containment

The player places barriers that stop/slow/redirect enemies/NPCs, to prevent them from reaching the player or a destination (cf. ENCIRCLEMENT, p. 27).

Examples

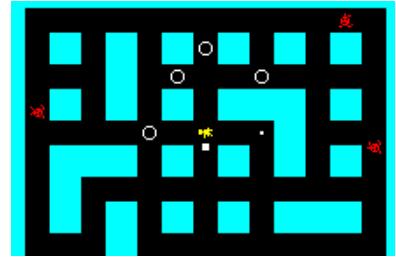
The classic example of a containment game is *Heiankyo Alien* (1979), in which aliens have invaded Heian-era (794–1185) Kyoto and the protagonist's only defense is digging holes into which aliens can fall. This same general idea, presented as a 2d side-on platformer, became *Space Panic* (1980) and *Lode Runner* (1983).

Containment plays a key role in the *Eggerland/Lolo* series (e.g. *Adventures of Lolo*, 1989), one of the best puzzle series of the 1980s. These are BOX-PUSHING (p. 193) games where you mostly use the blocks to sequester enemies in ways that prevent them from harming the protagonist.

Tower-defense-style games with free pathing (i.e. where enemies move freely rather than follow a fixed path) also involve containment; part of the strategy is to use towers as barriers so that enemies can't just directly progress to their goal. *Assembly Required* (2018) is a non-violent tower-defense-like game in which you manage an IKEA-like furniture store, and seek to delay customers on their search for the exit by trapping them in complicated store layouts.

Analysis

This pattern shares a lot with LEMMINGS (p. 128), but the difference is the goal; here, you are generally trying to lead them *away* from something rather than *towards* it. Manipulating an NPC so that it does *not* reach point B is a lot easier than the opposite; if the NPCs were as stupid as lemmings, there would be no game because they would be unlikely to reach point B anyway. To make this a challenge, the game typically has smarter NPCs (e.g. with pathfinding) and real-time action elements to force the player to think quickly.



Heiankyo Alien



Lode Runner



Adventures of Lolo



Assembly Required

Minions

The player assigns tasks to AI-controlled workers but does not directly control them.

Minions, as I'm using the term, are usually smarter and more autonomous than the "lemmings" of *LEMMINGS* (p. 128) – minions often have basic pathfinding, at least – and the focus of the game is more about the tasks themselves.

Examples

This is, of course, the central design pattern of the real-time strategy (RTS) games that dominated the marketplace in the 1990s, and if that's before your time I'm sure there are many better sources to explain them. So I'll illustrate this with games that are more on the periphery of that genre.

This pattern is also common in a lineage of building games, like the builder/RTS hybrid *The Settlers* (1993). It's also worth looking at *Caesar 3* (1998) and *Pharaoh* (1999), in which wandering workers keep your city functioning. In these games, the AI for your "walkers" is sufficiently simple and deterministic that they are almost *LEMMINGS*, but I put them here because the emphasis is on the work the walkers do, rather than leading them to safety. *Lethis: Path of Progress* (2015) is a modern spiritual successor to "walker" games.

Pikmin (2001) is a good illustration of this pattern without mouse control. Little plant-based minions follow the main character and can be activated and assigned tasks using controller input.

Analysis

The obvious advantage of this kind of control scheme is that it allows you to control large numbers of units without having to assign orders to each in turn; it's what allowed war strategy games to go real-time in the first place.

Its downside in competitive RTS is well-known, that it's hard to build a game that isn't dominated by people who are just *very good* at



The Settlers



Caesar 3



Pikmin

using their mouse to issue orders quickly. On the other hand, this is probably somewhat irrelevant to our concerns, unless you're actually making a competitive multiplayer nonviolent RTS.

The primary frustration of this kind of indirect control scheme are minions who get stuck on environmental features, stop working on things for unclear reasons, etc. For these reasons, minions are sometimes actually a sort of illusion. Rather than have the economy dependent on the actual behavior of a flock of little idiots, the minion simulation is sometimes epiphenomenal, a sort of complex animation that represents work getting done in the background. For example, the economic simulation may just state "Lumber production is proportional to the number of workers assigned to the sawmill", and lumber keeps being produced during work hours even if the little animated minion gets stuck on level geometry on the way to work.

Automation

The player sets up machines or similar processes to complete tasks without direct player supervision.

Examples

This is the central pattern behind automation games like *Factorio* (2016–2020) and *Automachef* (2019) as well as many programming games like *SpaceChem* (2011) and *Opus Magnum* (2017). Redstone circuitry in *Minecraft* (2011) is also worth special mention.

The earliest commercial videogame examples I can think of are *RobotWar* (1981), in which players program tanks to fight in an arena, and *The Factory* (1983), which I cover in more detail in ORDERING PUZZLES (p. 188).

Programming mechanics have also been integrated into more varied adventure-style titles. *Final Fantasy XII* (2006) tasks the player with programming the battle AI of their party members (e.g., firing off a Cure spell when a companion’s health drops below 50%). *Unicorn Overlord* (2024) is a more recent RPG with a similar conceit.

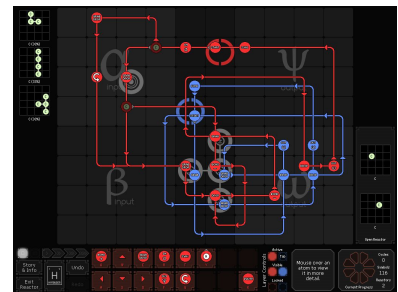
Hack-n-Slash (2012) and *Else Heart.Break()* (2015) take a more direct approach, letting you “hack into” many objects in the game world.

Analysis

The intricate and repetitive motion of automation games is high in inherent visual interest, similar to LEMMINGS (p. 128) and many HIDDEN OBJECT (p. 155) games. Much of *Opus Magnum*’s success was probably due to the ease with which players could share GIFs of their intricate creations on social media.



Factorio



Spacechem



Final Fantasy XII’s “Gambit” system

Chain reactions

The player tries to set up a sequence of objects or events, where each one will cause a reaction in the next.

Examples

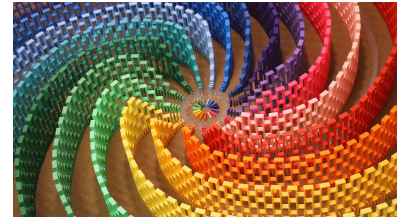
The classic entertainment of chain reactions is setting up a line of upright dominoes and knocking them over, so that each knocks into the next. *Pushover* (1992) is a classic puzzle-platformer in which you have to set up dominoes, many with special effects.

Expanding this to more heterogeneous objects leads to what is often called a *Rube Goldberg machine*, after the classic cartoonist. The children's game *Mousetrap* (1963) involves setting up and trigger a Rube Goldberg machine, but this has little to do with the actual gameplay. The earliest game I can think of in which you play a decision-making role in building Rube Goldberg machines is the edutainment title *Creative Contraptions* (1985), but even here the player had relatively little creative freedom.

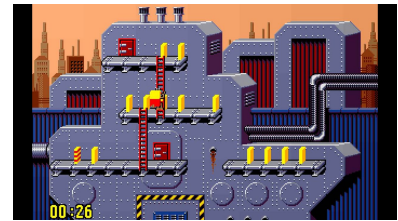
The classic hit in this genre was *The Incredible Machine* (1993), where the player is given much more freedom to build inventive machines out of hundreds of components, like using conveyor belts, balloons, and lasers to direct a pool ball into a basket. *Contraption Maker* (2014) is a spiritual successor by the same designer.

Analysis

There is a close connection between *LEMMINGS* (p. 128)-style games and games like *The Incredible Machine*; both involve a planning and iteration cycle, and if anything goes wrong you have to start all over. (This happens even more often in *The Incredible Machine*, since it has a physics engine; pixel-level placement might cause a ball to *RICOCHETS* (p. 136) in unpredictable ways.) As I note when discussing *Lemmings*, this can make for slow and frustrating gameplay, and you may want to consider having rewind and fast-forward buttons to make it easier for the player to adjust their solutions.



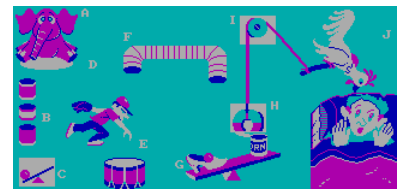
An intricate domino layout by Lily Hevesh (*hevesh5* on YouTube)



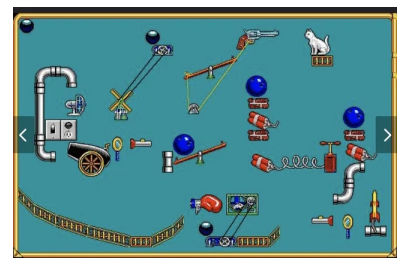
Pushover



A Rube Goldberg comic, courtesy the National Museum of American Jewish History.



Creative Contraptions



The Incredible Machine

Ballistics

The player chooses only the initial trajectory and velocity of an object in flight.

Examples

Most real-world ball/missile sports like *Golf*, *Archery*, and *Baseball* involve only ballistic control over their respective projectiles.

Ballistics also play an obvious role in “artillery” games where players lob missiles at each other. These were some of the earliest videogames, emerging in the 1970s as purely text games, and evolving into their familiar 2d side-view graphical form in the early 1980s with games like *Artillery Duel* (1982). *Scorched Earth* (1991) and *Worms* (1995) were particularly popular implementations.

“Launcher” games were a popular genre in the Flash era, in which you launch an object and various obstacles in its path either relaunch or hinder it, with the goal of having it travel as far as possible. The first launcher I can track down was *Yeti Sports: Pingu Throw* (2004); another notable early example was *NANACA+CRASH!!* (2005), in which a girl on a bike collides with a boy and sends him flying.

Launcher games were probably an influence on the next big Flash genre, catapult/slingshot DEMOLITION (p. 88) games like *Crush the Castle* (2009) and *Angry Birds* (2009).

Analysis

With this and the related RICOCHET (p. 136) games, it’s important for the designer to keep in mind that the player is only directly interacting during limited slices of the total gameplay time, so there is a danger that the player gets bored or frustrated by long periods in which they have limited control. Very much like slot machines, ballistic and ricochet games tend to have explosions, flashing lights, sounds, or other stimuli to keep the player cognitively engaged.



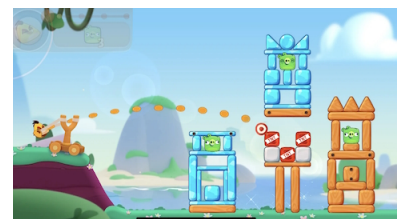
Real Golf 2011



Tank Wars



NANACA+CRASH!!



Angry Birds

Ricochet

An object (typically a ball or puck) bounces chaotically through a level, usually according to the angle at which it impacts level elements. The player only has indirect control of this element, typically controlling it at the time of its launch (as in *Billiards*, *Pachinko*, or *Mini-Golf*), by flippers (as in *Pinball*), or by impacting it with a free-moving paddle (as in *Air Hockey* or *Breakout*).

Single-player ricochet games often contain elements of **GROUND AVOIDANCE** (p. 46) where the player is fighting against gravity to keep the ball from falling through the level to a failure state below.

Examples

The classic ricochet games are those in the *Billiards* family, like *Eight-ball* and *Snooker*. An 18th century variant of billiards, *billard japonais*, inclined the table to introduce gravity as an opponent; this evolved into today's *Pinball* and *Pachinko*. *Peggle* is the pachinko-style game you are probably most familiar with.

Like *billard japonais* and *pachinko*, early pinball machines had no paddle and relied more on luck than skill, and quickly became associated with gambling and even banned. The innovation of the paddle (first introduced in Gottlieb's *Humpty Dumpty* in 1947) was quickly adopted by legitimate manufacturers in part to re-brand pinball as a game of skill.

Breakout (1976) and its many successors fall under this category as well, except that instead of controlling a flipper you control a Pong-like moving paddle at the bottom of the screen.

I am Level (2003) has the structure of an exploratory platformer, but, as the name suggests, the player plays as the level instead of the ball-like protagonist. *Yoku's Island Express* (2018) is a recent take on a similar idea, combining pinball with a metroidvania-style exploration game.

Sonic the Hedgehog games provide many examples of pinball-like elements even in a genre where the player usually has direct control of the protagonist. The invocation of pinball tropes is no coincidence

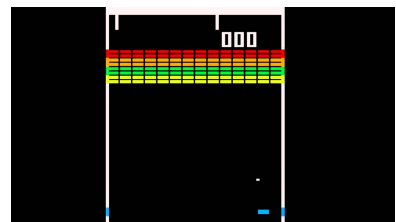


Billard japonais was the ancestral game to *Pinball* and *Pachinko*. Despite its name, it originated in Western Europe.

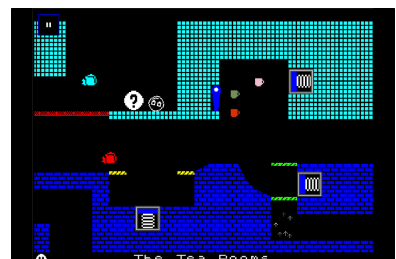


Peggle

Ledino, Jamie. 2020. *Attract Mode: The Rise and Fall of Coin-Op Arcade Games*. Audubon, NJ: Steel Gear Press.



Breakout



I am Level

– among platformers, *Sonic* games have probably the longest periods where the protagonist is out of the player’s direct control, his trajectory is determined primarily by level geometry. The designers particularly lean into this in the platformer-pinball hybrid *Sonic Spinball*: while the player retains some ability to control Sonic himself, for the most part new areas can only be reached by controlling the flippers and hitting Sonic like a ball. In a similar but much-more-frustrating fashion, *SNOLF* (2019) is a fan-made FUMBLECORE (p. 120) romhack in which players play through the original *Sonic* levels using only the BALLISTIC (p. 135) controls of a golf game.



Sonic Spinball

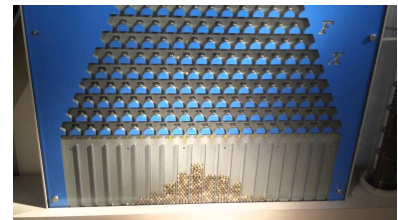
Analysis

Although the connection may not be obvious, this is a related pattern to LEMMINGS (p. 128) – what differs is the nature of the algorithm that controls the NPCs/balls. There, a simplistic and predictable AI algorithm typically leads to a deliberate puzzle-strategy game; here, an unpredictable physics simulation leads to a chaotic action game.

One important aspect of the design of pinball/pachinko-like games is that, although the course of any one ball is chaotic, over many plays a predictable distribution emerges. The player, watching individual runs, may not realize this, but the designer must understand a layout’s longer-term distribution through repeated simulation, in order to apportion rewards and maintain player engagement.

Long-term distributions in these games are illustrated best by *Plinko*, originally from the gameshow *The Price is Right*. In it, the player approaches a sloped board and drops a puck, which falls downward chaotically through a lattice of pegs, and into one of many slots at the bottom of the board. Over many plays, the frequency of different slots being hit is highly-predictable: it approaches a Gaussian distribution around the center of the board.

Pachinko-style mini-games are thus a visually-interesting way to theme a random sample from a Gaussian distribution. Due to its slowness compared to a simple dice roll, this is more appropriate for a rarer kind of draw where the possibility of a large reward keeps the player attentive. When *Plinko* debuted on *The Price is Right* in 1983, the suspense was heightened by the possibility of the largest cash draw yet offered on a TV game show.



This *Plinko* machine at the Math Museum NYC (video © Alex Teich) illustrates how the results approach a Gaussian distribution.

Gravity manipulation

The player controls which direction counts as “down” by changing the slope of the playing field, or otherwise manipulating gravitational attractors or magnets.

Examples

The classic wooden *Labyrinth* (1946) toy illustrates this kind of game in the real world, since changing the slope of the board changes the direction in the ball falls. *Labyrinth*-like games became especially popular in early mobile app stores, after the introduction of accelerometers into mobile devices let the player play by changing the real-world orientation of their devices. For example, *Taxi Ball* turns *Labyrinth* into a *Crazy Taxi*-style DELIVERY (p. 100) game.

An example of a *Labyrinth*-like game with only one dimension of tilt can be seen in *Ice Cold Beer* (1983), an electromechanical arcade game in which the player uses two joysticks to maneuver a rod on which a ball balances. The goal is to maneuver the ball to the top while avoiding the many holes in the playfield. This idea was updated to a modern “rogue-lite” format in *Tumbleseed* (2017).

One advantage of videogames is that we can introduce strong gravity sources in a way that we can’t in the real world. *Heaven and Earth* (1992), a mystically-themed minigame compilation, contains a software toy in which the player influences a pendulum with an “energy vortex” to collect rewards. Along similar lines, *Auditorium* (2008) is a music puzzle game in which streams of particles must be bent by gravitational attractors into goals. Each stream is themed as a track within a music composition, and the goals “amplify” those particles into audible music.

Strange Attractors (circa 2006) is a pinball-like “one-switch game” (a game playable with only one button), in which a ball bounces freely in space, but holding down a button activates “attractors” scattered throughout the level. It was designed to be playable even by gamers with severe motion disabilities, and received or was nominated for several awards for accessibility and innovation



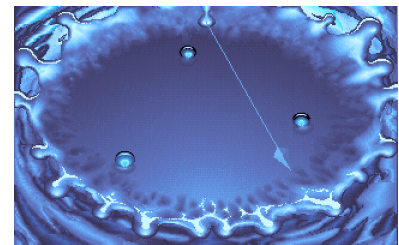
Labyrinth



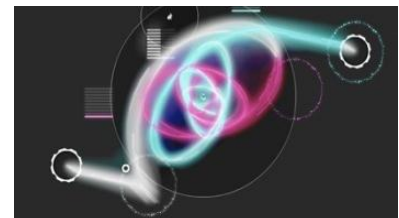
Taxi Ball



Tumbleseed



Heaven and Earth



Auditorium

Part X

Power-ups and status effects

Introduction

In PROGRESSION AND UPGRADE PATTERNS (p. 228), I concentrate on the kinds of upgrades that are permanent, and form the backbone of progression in many games. In this chapter, I concentrate instead on temporary effects that come and go during the ordinary course of play, like Mario's fire flower or a "poison" status effect.

These patterns are all closely related, and I collect them here in their own section because I think that, together, they provide a structure for non-violent (or at least less-violent) gameplay that is underappreciated.

Nintendo, in particular, uses these patterns frequently. We can trace a lineage of platformers that refines these ideas, from the original *Super Mario Bros.*, through the *Kirby* series and *Wario Land II* and *3*, and finally to *Super Mario Galaxy* and *Super Mario Odyssey*.

While none of these games are exactly *non-violent* in the strict sense that I use, they are certainly *less* violent than most competing series, and they became even less violent over time. You don't have to kill *that* many enemies in *Super Mario Odyssey*, and quite a bit of your time is spent doing other things entirely. Indeed, you often avoid harming enemies because they are the source of the abilities you need to overcome the platforming challenges, and your main "danger" in the game isn't dying from enemy attacks directly, but losing the power-up you need for the challenge you're currently facing.

Together, these patterns let us replicate some of the classic action/adventure/platformer challenges, while removing the need for some of the classic combat tropes like HP and damage. In *Wario Land II* and *3*, Wario is completely invulnerable and has no HP. In *Woten*, the "enemies" are completely invulnerable. These are still challenging and interesting games, however, because the primary challenge isn't staying alive, it's navigating a series of temporary abilities, power-ups, status effects, and variant forms.

Power-ups

The player receives a power that gives them greater abilities, but usually temporarily; they lose it after a certain amount of time, when they're hit by an enemy, or under other undesirable conditions.

Examples

There is no better example here than *Mario*: when he receives a fire flower, etc. he gains a new ability, but if he's hit by an enemy he loses this ability.

Many shoot-em-ups also rely on temporary power-ups (as opposed to permanent ones); you typically do not retain them after you die, and sometimes they are lost after incurring a certain amount of damage. The unique power-up systems of *Galaga*, *Gradius*, and *TwinBee/Parodius* are central to those games' identities.

Analysis

I treat this as different from Metroidvania-style upgrades because those are typically permanent. Metroidvania-style upgrades might be disabled temporarily for plot reasons, or for a special challenge section, but they're not regularly lost as a part of ordinary gameplay. Temporary power-ups, on the other hand, are expected to be lost eventually, except perhaps by the most skillful players.

This has a distinct gameplay consequence, because getting a temporary power-up causes the player to try especially hard to avoid the circumstances under which it's lost. If the power-up has special conditions for its retention or loss (e.g. many of the VARIANT FORMS (p. 144) of *Super Mario Odyssey*), this can add another level of challenge to the game.

In many games, power-ups function as a sort of special hit-point (e.g., Mario loses power-ups when hit, but he goes back to regular Mario rather than losing a hit point, becoming tiny, or dying). Note that this creates a positive feedback loop: when the player receives this special sort of "HP" they become stronger; when damaged they

become weaker. This can mean that a player, upon beginning to fall behind, loses more quickly than they otherwise would.

This feedback loop might or might not be something you want in your game, so be aware of it. Some of the less-successful shoot-em-ups of the 1980s had this problem in spades. You progress to a certain point in part because your accumulated power-ups give you the firepower to do so, but as you take damage, you lose them, and are quickly overwhelmed by the enemies' now-superior firepower. Then you re-spawn, still without those power-ups... and are quickly overwhelmed again. There's nothing you can really do except continue to lose lives or quit the game.

While this problem doesn't come up *exactly* in this form in a non-violent game, keep in mind whether taking away a power-up effectively ends the game or strands your player in a place they can't escape.

Challenge

As mentioned, Metroidvanias typically rely on permanent upgrades rather than temporary power-ups. How would you design a Metroidvania in which *all* power-ups are temporary? How do you prevent the player from getting stranded when the power-ups are lost or run out?

Ability borrowing

The player steals or otherwise utilizes the distinct abilities of enemies, in order to achieve the player's goals.

Examples

The earliest example of this I can think of is the rogue-like *Hack* (1984), in which eating monster meat could sometimes lead to you acquiring their abilities. Some RPGs have classes or abilities where characters can learn monsters' signature abilities by observation; in *Final Fantasy* parlance this is "Blue Magic".

In platformers, ability stealing is most associated with the *Kirby* series, in which Kirby can take on the form of most enemies by inhaling them. Many games express the stealing of enemy abilities by reskinning the character sprite to resemble the enemy, and thus this shades into VARIANT FORMS (p. 144).

For an example of a game in which creatures' abilities are utilized without taking on their forms, *Woten* is a puzzle-platformer in which almost all non-trivial movement is achieved by using the special abilities of the local creatures.

Analysis

Along with VARIANT FORMS (p. 144), this pattern is a very useful one in non-violent gaming because it gives the designer something else to do with "monsters". A lot of non-violent game worlds are a bit sterile; either they have relatively little life, or the life is mostly cosmetic, with little reason for the player to really pay attention to it.

By incorporating player abilities into "monsters" instead, the designer has a reason to fill the world with life, and the player has a reason to pay attention to them, and look forward to them. Even when dangerous, the player comes to treat them a little more as "friends" rather than "enemies"; this was an expressed design goal of *Woten*.



Kirby and the Amazing Mirror



Chef-mage Quina learns monster abilities by eating them in *Final Fantasy IX*. Actually, since this is a permanent acquisition, Blue Magic probably belongs as a PROGRESSION/UPGRADE pattern instead, but I'll stick it here anyway.

Variant forms

The player receives a power-up, upgrade, or variant form that changes their basic ability set or the way they act with other objects in the game.

Sometimes, variant forms change which things in the level are desirable and which things are dangerous. This forces the player to look again at familiar level elements in a new light, even allowing novel levels to be embedded into levels where the player has already been.

Examples

This pattern goes back at least to *Pac-Man* (1980), where eating “power pellets” flips ghosts from being dangerous to delicious.

Modern 3D *Mario* games like *Super Mario Galaxy* (2007) and *Super Mario Odyssey* (2017) provide excellent examples of how adding *weaknesses* to variant forms can add challenging new aspects to familiar levels. When Mario acquires the Bee Suit in *Galaxy*, previous hindrances become desirable (honey allows you to climb walls) whereas water, previously innocuous, will rob the player of the suit and often require them to restart the challenge. Similarly, when Mario possesses a Lava Bubble in *Odyssey*, lava is now his safe place, whereas landing on a hard surface turns him back into normal Mario.

Probably the earliest games in which this pattern was fully worked-out were two early (but rarely talked-about) forebears to modern *Mario* games, *Wario Land II* and 3. *Wario* has a number of variant forms... but, in keeping with him being a sort of Bizarro-world *Mario*, these are often themed as *power-downs* incurred as punishment for not avoiding enemy attacks. For example, some attacks light *Wario* on fire... but at the same time, this fire can be used to burn through barriers. The challenge in these levels is to obtain the variant form, and hold onto it long enough to use it where it’s needed.



Super Mario Galaxy



Super Mario Odyssey



Being on fire in *Wario Land II* allows *Wario* to burn through special blocks.

Analysis

This pattern can go hand-in-hand with a more non-violent play style, because “lifts” the challenge away from basic survival. The core challenge of the game isn’t so much “hang onto your life” as it is “hang onto that variant form”. *Wario Land II* and *3* take this idea to its logical conclusion and make Wario himself completely invulnerable.

This pattern synergizes well with ABILITY BORROWING (p. 143), since that in turn encourages the players to see “enemies” as potential benefits. Both patterns help to eliminate an aspect of the traditional adversarial relationship between player character and “enemies”.

I also think we see this pattern so much in modern 3D *Mario* games because it solves a basic problem in Mario’s move to 3D. In a 2D platformer, enemies are a more serious threat because you have fewer ways to avoid them: the *only* way to avoid an oncoming goomba is to jump over them. In 3D, you can just run around the goomba; the main threat a goomba poses in *Super Mario 64* (1996) is that you misjudge your distance and hit it by accident. If enemies can no longer serve as a primary challenge, what can? Later 3D games like *Galaxy* and *Odyssey* lean more heavily into power-ups/variant forms to provide the iconic challenges of these games.

Status effects

The player receives a temporary loss of power, loss of an ability, or loss of control, often as a punishment or as the result of an enemy attack.

Examples

We usually associate status effects with RPGs, but they occur in a variety of genres. We can categorize them in several types:

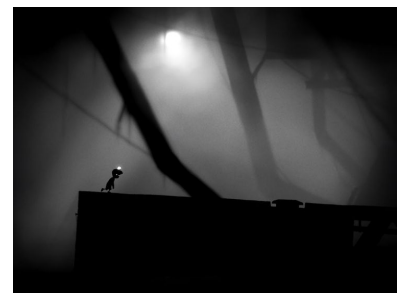
- Gradual losses of HP or other stats (e.g. “poison”), or countdowns to death.
- “Debuffs” in which stats are reduced (e.g. “weakness” as attack reduction, “blindness” as accuracy reduction)
- Inability to use certain abilities (e.g. “silence” as the inability to cast spells) or have abilities used on you (e.g. not being able to be healed).
- Stunning and freezing (p. 148), cancelling the player’s next action (e.g. in *Grandia*), or causing the player to lose a turn.
- Causing a party member to act randomly (e.g. “confusion”), turn against the party (e.g. “charm”), or otherwise making the character difficult to control (p. 150).
- Obscuring information from the player, like the screen-obscuring Blooper Ink status effect in *Mario Kart* games, discussed further in LIGHT AND SHADOW (p. 219).

Analysis

Like other patterns in this chapter, this is an interesting one in light of non-violent design. Combat games are, fundamentally, about *damage*, both mechanically and thematically. Patterns that can stand in for damage are thus particularly interesting when trying to remove prototypical combat from a game.



Several party members here in *Final Fantasy II* have been “silenced”, making them unable to use magic.



When infected with a brain parasite in *Limbo*, the player cannot turn around and can only move in one direction.

Stunning, freezing, and lost turns

When a player/enemy/NPC fails to avoid an obstacle or attack, they are temporarily unable to play or move.

Examples

Freeze Tag is classic children's game where you are frozen in place rather than removed from play. Importantly, other players can "un-freeze" you by tagging you, meaning no one is permanently out of the game until the game is over.

The classic example in videogames of a prolonged freezing attack is in *Metroid*. You can even use frozen enemies as platforms, meaning the ice beam is also used as a way to get the protagonist to new areas (cf. MAP EXPANSION, p. 228).

In games where actions can take a while to prep, or can be queued up (e.g. an RPG where a spell takes a while to say), the actions can sometimes be "cancelled" when they're interrupted. This is a major source of gameplay in the *Grandia* series of RPGs, which depend heavily on timing your actions to interrupt enemy actions.

Analysis

You should be particularly cautious about status effects that cause the player a loss of control and locking of abilities – players do not appreciate being prevented from playing for too long, or kept from using the mechanics that make the game fun. Even when stuns are programmed to be of brief duration, consider whether they can happen in succession, resulting in a player being "stun-locked" and unable to play for longer periods.

Always remember that players have an easy solution for when they feel like you're preventing them from playing: they can go play another game.

Knockback

When a player/enemy/NPC fails to avoid an obstacle or attack, they are knocked backward.

Examples

Many games have stun and knockback effects when a player (or enemy) is attacked, and these can even be more dangerous than the attack's primary effect. For example, by the time the player gets to the aquatic Maridia section of *Super Metroid*, they've typically found enough energy tanks that the enemies pose little direct danger to your life. At this point, the main danger they pose is to your platforming, by knocking you off narrow platforms.

Analysis

We should keep it clear in our heads that knocking a player off a ledge is not necessarily kinder than just killing them. We tend to conflate frequent deaths with difficult and even cruel design, but it's important to keep separate the theme of the setback from how bad of a setback it actually is.

Falling off a high-up ledge in (say) a *Sonic* game doesn't usually kill me, but it's a significant setback. If my goal is to complete that particular challenge, I have to slog all the way back up to where I fell. (If it's even possible! That route may be inaccessible now, and I can only retry by restarting the level.)

Meanwhile, if I'm knocked off a ledge in a masocore platformer, I'm probably immediately impaling myself into spikes and will quickly respawn nearby. That's actually the kinder design. Both *Sonic* and *Super Meat Boy* depend on memorization of challenges, but in the latter this try-fail-and-try-again loop only takes a matter of seconds.

So if you take this pattern to its logical conclusion like *Wario Land II* does (in which Wario has no HP and only suffers status effects), it's important that you take care to design levels so that falls and similar setbacks are not too punishing (cf. the caveats in CLIMBING, p. 110).



The actual damage from this enemy in *Super Metroid* is not very high compared to Samus's HP at this point in the game, but the resulting knockback can be killer.

MASOCORE:

A genre of platformers designed to be difficult, and in which dangers typically kill you in one hit. On the other hand, they tend to balance this difficulty by giving the player infinite lives, and are usually generous in re-spawning the player near the beginning of each challenge.

Pushing and throwing

Players/enemies/NPCs can push or throw each other into different positions.

Examples

The *Sumo*-inspired strategy games *Abalone* (1987) and *Auro* (2014) use pushing/bumping as their central mechanic. In both of these games, pieces are invulnerable, and can only be removed from play by pushing them off the playing field.

One of the most distinctive systems in the *Disgaea* tactical strategy games is the ability to pick up teammates (and pick up teammates picking up teammates) and throw them to other parts of the board.

For an example of using “throws” to make a non-violent chess-style boardgame, *Diplomats* (published in *GAMES Magazine*, February 1985) challenges players to move all of their slow pawns (“VIPs”) into the opponent’s last row. Powerful pieces (“Diplomats”) can move VIPs in ways they cannot move themselves (e.g. “move an adjacent VIP any number of spaces diagonally”).

Analysis

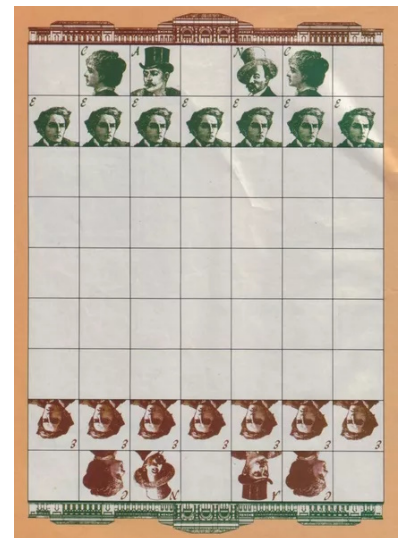
This is an interesting pattern because the benefit/harms stemming from a character’s position are more ambiguous and complex than just losing hit-points. Being moved to another position might be a severe setback, no big deal, or even give you an advantage.



Abalone



Auro: A Monster-Bumping Adventure



Diplomats

Confusion

The character acts or moves randomly, becomes hard to control, or has their input interpreted in incorrect ways.

Examples

Many RPGs have a “confusion” status effect where a character is removed from player control and acts randomly instead, including attacking their own party members. *Final Fantasy* games also have being “charmed” (acting randomly but on behalf of the enemy team).

More interesting to me are status effects where the player retains partial control. For example, in *Cadence of Hyrule*, confusion results in the character moving opposite to inputs (e.g. when the player presses down, the character moves upward).

The most notorious status effect in platforming games may be in the *Yoshi's Island* (1995) level *Touch Fuzzy, Get Dizzy!* Touching a floating “fuzzy” does not harm Yoshi or Baby Mario, but the world starts undulating in a psychedelic way and Yoshi starts to stumble like a man in the depths of an ether binge, making platforming in this level especially perilous.

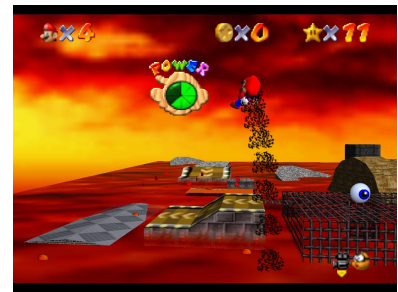
While it's not themed as confusion, we might also consider here 3D Mario's behavior when he falls into lava. Unlike in the earlier 2D games, Mario does not immediately die from falling in lava; instead, he jumps very high into the air and then runs around in a wild and hard-to-control manner (often, right back into the lava). However, in the hands of a skilled player this high jump can even be a bonus; it's an important tool in the speedrunner's toolbox.

Analysis

Confusion-like status effects can introduce an element of FUMBLECORE (p. 120) into an otherwise ordinarily-controlled game.



Yoshi's Island



Super Mario 64

Part XI

Search and gathering

Gathering/foraging

The player finds an item (say, on the ground, or in a designated gathering spot) and it is added to their inventory.

Examples

This is in so many games that it's hard to pick any one of them to illustrate. But one difference in gathering that's worth noting is the difference between seeing the actual item in place, and going to a gathering spot and getting a randomized reward.

In *The Legend of Zelda: Breath of the Wild* or *Fallout 3*, you see the item in the world: the the apple in the tree, the typewriter on the desk. This increases environmental immersion, because things are generally in realistic places and get you thinking about the area as a coherent ecosystem and society.

On the other hand, in the *Atelier* games, the player gathers ingredients from designated gathering spots, and gets randomized rewards. This decreases immersion – it's artificial and reminds the player that this is a game – but also sets up a variable reward cycle.

You can have elements of both of these systems. *Stardew Valley* mostly just places its gathering items on the ground, but also indicates special “Artifact Spots” by the presence of wriggling worms. Digging in an artifact spot might unearth anything from an ordinary resource to a rare artifact.

Analysis

This is so trivial and in some many games that I hesitated to include it. But it's sometimes maligned due to its ubiquity – I sometimes hear designers say “I'm not going to add gathering just to add gathering” – so I wanted to say a little bit more about what gathering is for.

Obviously, gathering isn't very much fun on its own. It only becomes compelling once in service of high-level goals – say, to build furniture, to decorate your house, as a means of SELF-EXPRESSION

(p. 39) to other players. Daniel Cook, of the studio *Spry Fox*, calls these “value chains”.

The core activities in a life sim are generally not full of mastery and depth. You chop trees. You dig holes. You pick up sticks. In isolation, each of these is dull. Our playtesters would harvest a leaf pile, get some sticks, and then put down the controller. They’d turn to me and ask “Uh, okay, where is the game?”...

It wasn’t until we spent 12-months building out the rest of the game – the crafting, the decorating, the daily pacing structures – that players finally began to value picking up sticks. Because it turns out the value of sticks was entirely driven by their utility in reaching future goals. And if those future goals don’t exist, the sticks have no value.

In addition to its role in the game economy, gathering also contributes to the cognitive variety of the game, by making the player remain somewhat mindful of their surroundings.

Players inevitably stop *looking* at your game over time; the beautiful forest path that you spent so much time on eventually just becomes “A 20-second delay in between A and B”. Gathering delays that inevitable abstraction process a bit by keeping the player looking around, and having to recognize level elements as what they are, at least so far as it’s relevant to what they can gather. This is a forest, be on the lookout for berries. That’s an oak tree, there might be acorns on it. That’s a river and bubbles might indicate fish. It requires that the player keep a basic mindfulness of their surroundings rather than zone out completely.

Daniel Cook. 2021. “Value chains: A method for creating and balancing faucet-and-drain game economies.” In *Lost Garden*, Dec. 12, lostgarden.home.blog/2021/12/12/value-chains/

Set collection

This is a sort of GATHERING (p. 152), but with an emphasis on the items being parts of desirable sets, rather than being desirable in themselves.

Examples

Many of the items in *Stardew Valley*, for example, have limited practical use and don't sell for much, but are ultimately necessary to complete "bundles" (collections of related items that are necessary to win the game).

Many CRAFTING (p. 163) systems are set collection challenges. I'm not desperately scavenging typewriters for screws in *Fallout 4* because of the inherent value of screws, but because screws are crucial ingredients for many crafting recipes.

Analysis

Set collection is an important anticipation mechanic (cf. the Anticipation/Honeymoon/Neutral cycle described in LEVELING, p. 232). When I have no berries, I'll probably stop to pick a berry, but I don't really desire it. When I have maybe 3 out of 5 berries in the "berry set", I start specifically looking out for more berries, and when I have 4 out of 5 I *really* want that last one. Being in a set gives the berries meaning that they don't have inherently.

The other thing that set collection can give you is mild strategic decisions, due to an item being needed by two different sets. Do I use this item in set A now, or do I save it for set B later? This is the central decision of *Mahjong/Rummy* family of games, in which tiles or cards are combined into "melds": either sets (same number, different suits) or runs (consecutive numbers, same suit). Since every card can be used in either a set or run, there is a constant, mild tension, occasionally punctuated by a change in which a single new card makes you re-evaluate your entire strategy.



Stardew Valley bundles



Screws don't seem valuable in *Fallout 4* until you keep finding recipes that you can't make because you're out of screws.



Gin Rummy melds

Hidden objects

The player has to find objects that have been purposely made hard-to-find, by embedding them in a busy scene or disguising them within misleading geometry.

Examples

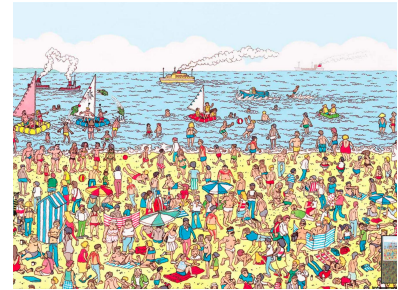
Where's Waldo (or *Wally*) is a well-known paper implementation of this, and there are hundreds of dedicated “hidden object games”, especially on mobile and tablet, that use this pattern as their primary game loop. Some hidden object games also require some light environmental manipulation (e.g. opening closets and window-blinds) to find the objects, as in *Hidden Folks*.

Spot-the-difference puzzles are, in a way, another kind of hidden object game; the player is given two similar pictures and has to find some number of differences between them.

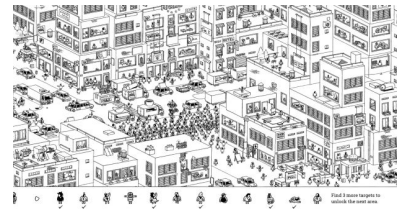
Aside from hidden object games, many adventure-style games have some sort of hidden collectable or environmental feature that you're encouraged to find. Sometimes these have gameplay significance (the bobblehead collectables in *Fallout 3* grant stat bonuses); other times they're just for completionists. Sometimes the hidden object represents a **HIDDEN PASSAGE** (p. 217), like the bombable walls in the *Zelda* games.

A simple but satisfying sidequest in *Final Fantasy V* is finding all the pianos. Each time you find a new piano and play it, a little tune plays, representing the main character's progress in finding pianos. At first, he's attempting incompetent scales and simple tunes; by the end he's playing elaborate pieces. Playing high-level tunes for a particular NPC grants the player new songs for use by the Bard class.

Another common and easy-to-implement hidden-object-like challenge in RPGs is, whenever you enter a town, to have your other party members scatter and become “NPCs” in the town. When you find and interact with them, you may get a reward, some character development, or some **RELATIONSHIP** (p. 52) progression.



Where's Waldo/Wally



Hidden Folks



Final Fantasy V has you on a constant lookout for pianos. Piano Flex!

Analysis

Part of the fun of hidden-object challenges in videogames is that they're often embedded in busy, intricate, animated scenes that are just fun to watch on their own. The German word for these scenes is "Wimmelbilder": literally, "teeming pictures". In *Where's Waldo* or *Hidden Folks*, looking for the hidden objects gives you a higher-level goal, but a lot of the actual fun is just enjoying the intricate art and tiny visual jokes.

Because adventure-style games have items anyway, and worlds in which to hide them, it can be quite straightforward to incorporate hidden objects into your game. For example, in *Final Fantasy VI*, almost every grandfather clock in the game contains a valuable Elixir, setting up a mildly amusing, entirely-optional spot-the-clock game in every indoor scene. Adding this probably didn't take any real development time; the "room" tileset probably already included clocks, and someone decided to give them a bit of gameplay significance by hiding a distinct reward in them.

An important downside, however, is that a well-hidden object in the critical path can be highly frustrating. Early graphical adventure games like the *King's Quest* series or *Maniac Mansion* had the problem that some important objects were quite small and easy to miss. Newer games typically have hidden objects be entirely optional, or (when finding objects is a necessary inclusion on the critical path because of the nature of the game) have commands or overlays that highlight/label all the interactable elements on the screen; the *Blackwell* series of graphical adventures is an example.

Scanning

The player focus their attention/camera on something in the world, and in return receives information on it.

Examples

This pattern is associated especially with *Metroid Prime* (2002); by pointing the camera at something in the world and holding a button for a particular length of time, Samus receives a datalog on that thing. The information she receives is sometimes gameplay-relevant (e.g. explains the AI and weaknesses of enemies) but also provides much of the lore; the main plot of the game is largely revealed through these datalogs.

This pattern has subsequently adopted into exploration/survival games like *Subnautica* (2014) and *No Man's Sky* (2016).

Analysis

Like PHOTOGRAPHY (p. 91), scanning can provide some of the challenge of an FPS in non-violent form. Scanning can be quite challenging on its own; it's frequent that the player must keep the target in view for a period of time in order to scan it, which can be difficult when the element moves fast or is dangerous to approach.

It's also a way to provide players game-relevant information in an optional and non-obtrusive way. You don't necessarily have to have, say, a tutorial about handling a new kind of gameplay element if the player is permitted to scan it.

Note that if you require scanning every object in the game for 100% completion, as the *Metroid Prime* games do, this has a significant effect on the way the player plays, as many scans in *Metroid Prime* are missable. Missing your single opportunity to scan a one-time object (e.g. a boss) makes it impossible to achieve 100% completion. If you do this (which I'm not recommending), consider perhaps rolling over scan completion into a NEWGAME+ (p. 257), to give the player additional opportunities to scan missed items.



Metroid Prime



Subnautica

Radar and triangulation

The player uses some variety of device to narrow down the relative position of something (cf. LOCATION DEDUCTION, p. 160) with respect to their character's position.

Examples

"Compasses" in many games are not actually used as compasses in the traditional navigation sense; it's not as if you're wondering where north is in *The Legend of Zelda*. The "compass" is usually a sort of special-purpose radar-like device pointing you in the direction of the boss, treasure, or your next objective.

Scanning the sky for audio signals is an important component of *Outer Wilds* (2022), and play into the knowledge-based progression in this game. Since the protagonist is stuck in a TIME LOOP (p. 72), physical changes to the world are ultimately temporary, but scanning frequencies count as "knowledge" and are remembered between loops.

One of the most important tools in the *Ape Escape* games is the "Monkey Radar", which tells you the direction and rough distance of the closest targets.

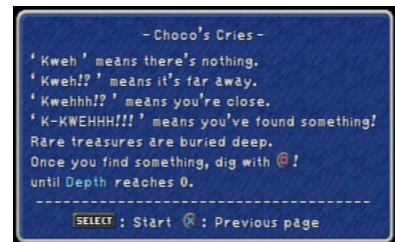
Sonar is interesting because active sonar also reveals information on your own location; *Subwar 2050* (1993) concentrates in particular on this.

The moonstone puzzle in *Indiana Jones and the Fate of Atlantis* (1992) might also fall under this pattern. You are searching for an artifact whose location is unknown, but you know that it was previously visible from particular vantage points. Using a sextant lets you triangulate exactly where the artifact is buried.

"Hot and cold" is a traditional children's game: the clue-giver chooses an object, the players select candidate objects, and the clue-giver responds with clues like "cold", "getting warmer", and "hot" depending on how far or close (respectively) the guessers are. The "Chocobo Hot & Cold" minigame in *Final Fantasy IX* (2000) is a straightforward videogame implementation of this.



Indiana Jones and the Fate of Atlantis



Choco the Chocobo's cries give hot/-cold clues in a minigame in *Final Fantasy IX*.

Treasure maps

The player is given a representation (often abstract/schematic) of some small part of their world, or a picture of it from an unfamiliar angle, and if they can figure out the place that it represents and go there, they get a reward.

Examples

It's typical that treasure maps show only a stylized or silhouette version of the level, as in *FEZ* (2012). Sometimes maps are incomplete and the player can assemble them, as in *Sid Meier's Pirates!* (1987).

3D games offer another twist on this, due to the ability of the camera to show the player levels from a novel angle. In *Super Mario Galaxy* (2007), for example, Luigi sends the player pictures taken in a level the player visited before, but often from a vantage point that the player would not ordinarily have encountered. In *Elden Ring* (2022), the player finds paintings by long-dead painters, and upon finding the site receives rewards from their ghosts.

Sometimes the scene is animated. The fortune teller in *The Legend of Zelda: Twilight Princess* (2006) shows you brief scenes of areas where you've missed a heart or other collectable object, but the player has to use clues in the scene to figure out exactly where that object is. The "Monkey Radar" in *Ape Escape* (1999) likewise can give you a brief glimpse of monkeys you have yet to catch.

Analysis

Of all the patterns in this book, this is among the easiest to implement; it can fit into pretty much any game where the player can move and there exist rewards to give them.

An added value of treasure maps is that they get the player looking closely at the game world through a different lens than normal. Not just "Those are a bunch of trees" or even "Do any of those trees have apples yet?", but "Are any of those trees forming an equilateral triangle, north of a rock?"



Sid Meier's Pirates!



FEZ



Super Mario Galaxy



Elden Ring

Location deduction

The player needs to logically deduce the location of one or more people/items, typically on a map.

Examples

Location deduction forms the core of the early classic *Hunt the Wumpus* (1973), in which the player hunts a monster through a series of caves. *Minesweeper* (1990) can be seen as location deduction as well: the player is attempting to deduce the location of many elements (themed as landmines).

In boardgames, the classic location deduction game is *Scotland Yard* (1983), which features a schematic map of London with bus and train routes connecting locations. One player is a fugitive, and they do not reveal their location on the board; however, they do reveal which method of transit they use each turn. The rest of the players are detectives, attempting to use this information to narrow down the fugitive's location.

GeoGuessr (2013) is a location deduction game where you are trying to determine your *own* location. The game randomly chooses a real-world location on Earth, and allows the player to look and move around on Google Street View from this point; the goal is to pinpoint this location on a map. Getting the maximum score requires the player to guess their location within about 30 meters. Top players use an astonishing variety of clues to narrow down where they are: signage, language, architecture, flora, and even details about the Google cars themselves (e.g. how far off the ground the camera is).

Analysis

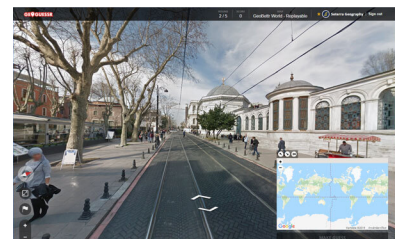
This is fairly closely related to TREASURE MAPS (p. 159), but distinct. Treasure maps locate the hidden person/object precisely, but it's unclear exactly what you're looking at, requiring you to match the map with features of the world. This pattern is more focused on the accumulation of clues and working out their consequences.



Scotland Yard



Minesweeper



GeoGuessr

Tracking and identification

The player is given clues about an animal or plant's appearance (and/or habitat, behavior, etc.), and has to find it.

Examples

Hunting games (both realistic and fantastic) rely on tracking and identification. In *Monster Hunter* games, one of your key clues is a stylized drawing of the animal, which will somewhat but not perfectly resemble your target.

Related to this, but on the non-violent side, are safari and conservation games. *Alba* (2020) is an open-world adventure game in which your primary task is to find and PHOTOGRAPH (p. 91) many animals (mostly birds). Like *Monster Hunter*, you do not have complete pictures of them, but you can deduce some of what you're looking for from partial pictures from damaged nature-preserve kiosks.

Finding and identifying plants is similar. An additional pattern in plant-tracking games is the presence of similar plants that are worthless or even poisonous. For example in *Carto*, one mission has you finding a particular medicinal herb, but it's planted among similar herbs that fit *most* but not *all* of the description.

Analysis

This is broadly a sort of LOCATION DEDUCTION (p. 160) and HIDDEN OBJECT (p. 155) pattern, but hunting/tracking/safari games are sufficiently different from hidden object games that I felt that they deserved their own section.



In the *Monster Hunter* series, the player is only given stylized representations of their targets.



Much of the side content of *Final Fantasy XII* is about monster hunting.



Alba: A Wildlife Adventure

Part XII

Crafting and item patterns

Crafting

The player crafts a collection of items into another item.

Examples

In most games, crafting is a kind of SET COLLECTION (p. 154), but (like STEALTH, p. 49) if I didn't include a chapter on crafting I'm sure people would ask "Where's crafting?!?"

Crafting often goes one step further than other set-collection mechanics, however, in that it turns a set of items into another item, which makes possible transitive relationships: an item can be the result of a recipe and also the ingredient to a further recipe. The resulting directed graph of recipes can support a non-trivial maze-like puzzle, e.g., "What sequence of recipes gets me from the ingredients I have to the result I need?" This is particularly explored in the *Atelier* games, which I discuss further in TRAIT PROPAGATION (p. 164).

Crafting sometimes involves elements of simulation as well, where they results and quality of the item depend on the player's performance in thematic minigames. For example, *Master of Pottery* (2019) takes the player through the steps of throwing, glazing, etc.

Analysis

When putting a crafting system into a combat-centric game, designers are often a bit slipshod, adding recipes without a lot of thought about how it affects the "crafting graph" as a whole. And that's understandable, because in these games something else is driving engagement and progression.

In the absence of combat, however, crafting often takes center stage, and it becomes extra important for the designer to understand the large-scale structure of their game's crafting graph, and work out the pacing with which the player gets access to parts of that graph.



A rare exception to crafting being set-collection is *Star Ocean: The Second Story* (1998) where crafting is actually a sort of GACHA (p. 250): you spend a single base item (effectively a sort of limited-use MONEY, p. 237) and receive one of many possible dishes (or failures), with the possible outcomes determined by the cook's skill level.



Master of Pottery

Trait propagation

The properties of a crafted object are inherited from its ingredients.

Examples

While this pattern isn't exactly rare, I want to concentrate on a particular series where it's central to gameplay: the *Atelier* games.

In these games, you typically play as an apprentice alchemist working towards mastery, and the core of the game is in trying to craft various objects. (There is also a conventional turn-based JRPG tacked on; they're adequate but not the reason anyone plays these games. Combat is primarily there to provide motivation for the player to strive for increasingly powerful equipment and bombs.)

Ingredients have properties like "Explosive" or "Lightweight" or "HP+10"; when crafting an item, you can choose which of the ingredients' properties should be inherited. As the game goes on, you start to receive seemingly-impossible requests, like (say) an explosive pie when none of the component ingredients naturally carries the trait "Explosive". Solving this requires starting with something that has the trait "Explosive" and propagating that through a series of recipes until you end up with an explosive pie.

Analysis

In most crafting games, even those with a complex graph of transitive recipes, the player doesn't really need to puzzle out the structure of that graph. The player can just craft recipes as they become available, and it happens that they can then use the results as ingredients to new recipes, even if they haven't planned ahead.

A trait propagation system can force the player to engage with the graph as a whole. Random crafting can take you partway into an *Atelier* game, but there's a sudden awakening midway through the game that this will no longer be sufficient. The desired traits are too rare, and the chain of recipes too long, for the right combinations to happen by accident.



Atelier Escha & Logy (2013). Any designer making a crafting-centric game should probably play one or two *Atelier* series games for reference, because few crafting games make you really consider the entire transitive network of recipes the way the *Atelier* games do. And don't just play for a few hours, but to completion, because the nature of the crafting puzzle doesn't become clear until the mid/late game.

First-time item bonuses

The first time the player crafts/uses an item, they get a bonus, often permanent.

Examples

In games like *Bloodstained: Ritual of the Night* (2019) and *Dysmantle* (2021), food is mostly just a kind of health potion. (In true *Castlevania* style, the main character of *Bloodstained* is entirely able to scarf down five turkey dinners in the heat of battle.) However, the *first* time you eat a particular dish, you get permanent stat increases, some of them significant. In *Bloodstained*, for example, the first time you eat flan you get +1 LUCK. In *Dysmantle*, the first time you eat catfish kabayaki your material drop rate doubles.

For a simpler version of this, the popular *Minecraft* mod *Spice of Life: Carrot Edition* (2017) raises the player's maximum HP according to how many different dishes they have eaten so far. This effectively turns eating into a sort of SET COLLECTION (p. 154), and the mod aids this with helpful screens detailing the dishes the player has and hasn't tried yet.

Analysis

This pattern is a way around the problem I first mention in RELATIONSHIP BUILDING (p. 52). When all the recipes in a crafting system boil down into one stat like HP, what keeps the player exploring the whole system rather than spamming out the currently-most-profitable recipe and ignoring everything else? "Flour is cheap and pancakes give decent HP, time to make 999 pancakes!"

First-item bonuses are a simple solution to this problem, since they directly reward novelty. They also give the player a reason to hold back some of their flour; you don't want to be on the cusp of achieving a new recipe and realize you used 100% of your flour on mere pancakes.



A first-time bonus in *Bloodstained: Ritual of the Night*



A first-time bonus in *Dysmantle*



Spice of Life: Carrot Edition

Supply-chain management

The player is concerned with making sure the inputs and outputs of a chain of processes (e.g. an automated crafting system) are available and balanced.

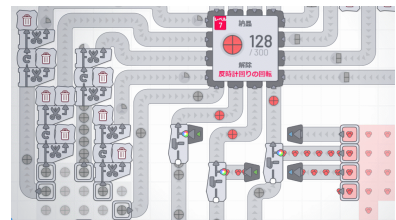
Examples

This is a common pattern in “builder” games in the lineage of *The Settlers*; the main modern builder series centered around this is the *Anno* series. While these games do have some light *SimCity*-like proximity constraints (e.g., the need to build arctic housing near heat-generating factories in *Anno 2205*), the core game loop is more concerned with supply-chain management. Your basic citizens require things like food that are produced directly by farms/factories, but citizens who have their basic needs met can be promoted to citizens of increasingly higher classes. Higher-class citizens produce more tax revenue, but demand access to luxury goods that require several stages of processing. Much of the gameplay involves making sure that sufficient inputs and workers exist for each stage of the processing chain – but building these facilities and attracting sufficient workers changes the balance of demands, so the player always needs to switch their attention between different aspects of the supply chain.

Supply-chain management is also an important aspect of AUTOMATION (p. 133) games like *Factorio* or *Shapez*. Like the *Anno* games, these games challenge the player to create complex goods out of a series of processes, and then once the player has mastered the basics, challenges them to scale it up to achieve much higher throughput.



Anno 1800



Shapez

Shops

The player exchanges MONEY (p. 237) for items, or items for money.

Examples

It's a shop; you know what shops are. I'll just use this space to illustrate some interesting examples of shops.

Most shops in games sell one kind of equipment or item, and tend to keep the same items in stock. However, it's valuable to also consider having a Random Shop that cycles its inventory rapidly between random sets of items, especially items that are seasonal, easy to miss, or have a low drop rate. These are often sold at an exorbitant markup, so that the player is still motivated to try to find these items on their own; this is only a last resort when mistakes or bad luck prevent the player from getting a necessary item.

In *Crypt of the Necrodancer* (2015), shop walls are made of gold. If you're low on money, you can bomb walls for a little extra, at the cost of bombs, meaning that the shops effectively accept two currencies.

Analysis

In most single-player games, shops also allow the player to trade in unneeded loot and equipment for money. Care should be taken in persistent, multi-player online games, though. Simpson (1999) found that this contributed to economic problems in *Ultima Online*.

When shopkeeper AI is smart and tries to run a sound business, they'll refuse to buy items that are in great oversupply (e.g. from many players practicing their crafting skills, as mentioned in LEVELING-BY-USE, p. 234), but then players get upset because they don't feel their efforts are adequately rewarded.

However, if shopkeeper AIs are forced to buy items at a set minimum price (regardless of the the item's true market value), they effectively become gold-dispensing machines, and lead to inflationary spirals. (Although, for better or worse, this was the least of *Ultima Online's* inflationary problems.)



It's easy to miss some seasonal items (like fish) in *Stardew Valley*. If that item was necessary for a bundle, the player would ordinarily have to wait another year to get it, which is frustrating if that's one of the only items left to "win" the game. The Traveling Cart alleviates this somewhat, by offering a random selection of goods that sometimes include these rare items.

Zachary Booth Simpson. 1999. "The In-game Economics of Ultima Online." Presented at the *Computer Game Developer's Conference*, San Jose, CA, Mar. 2000

Haggling

The player and an opponent negotiate for a price, or wages, that they both feel is fair.

Examples

The first computer game where you could set the price of a product (although not haggle with every customer) is probably *Top Management Decision Simulation* (1956). The first such game anyone still plays is probably *Lemonade Stand* (1973).

Theme Park (1994) has you negotiating wages with your park employees, themed as two hands reaching out, one of which you control. Where the hands meet determines what wages you will pay.

Haggling is part of the core game loop of *Recettear* (2007); every item you sell involves a negotiation.

Analysis

Haggling plays a role in relatively few games. It may just be that people just don't enjoy haggling that much; after all, how many people are *enthused* by the idea of negotiating with a used car salesperson?

Recettear avoids the pall on haggling somewhat in that the process isn't entirely selfish. You're not in a large, anonymous modern city, but in a typical RPG town, and all of your customers are regulars. As mentioned in *RELATIONSHIP-BUILDING* (p. 52), you don't necessarily want to drive the hardest bargain you can every time. Understanding your customers' wants and needs, and finding the sweet spot that makes them happy, will keep them coming back and spending more in the long run.



Theme Park



Recettear: An Item Shop's Tale

Provisioning

The player tries to predict how much of a resource they'll need in advance. However, there are penalties for carrying too much: either cost, time, inventory space, or spoilage.

Examples

A classic example of provisioning is *Oregon Trail* (1974), when deciding how much food/bullets/etc. you need to carry. If you run out of food, you have to go hunting, which consumes bullets and time. However, buying too much food wastes money and space. Stocking up on health potions serves a similar role in the typical RPG.

For a health potion system where potions expire if not used, consider *BomberBoy/Atomic Punk* (1990), a *Bomberman* series game for the GameBoy. The player character only has 1 hit-point per level, but can buy elixirs and bring them into the level, which will be automatically consumed if the player is about to die. (That is to say, they function as extra hit-points.) However, all elixirs *not* consumed during the level are lost, and they're rather expensive, so you don't want to bring more than really you need.

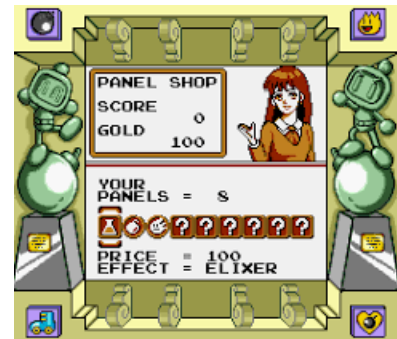
In *Touhou Mystia's Izakaya* (2021), the player is running a restaurant that is only open at night, and can pre-cook dishes in the morning to anticipate the evening rush. However, cooked dishes go bad overnight, so cooking in excess of the actual demand wastes ingredients and time.

Analysis

Provisioning is effectively a sort of WAGER (p. 247) on future challenges. It's interesting design-wise, because it gives the player an implicit difficulty dial. Provisions are usually buffers against potential failures – bad hunts in *Oregon Trail*, deaths in *BomberBoy*, too many orders in *Mystia's Izakaya*. The player is, effectively, gambling on their future performance – “Am I going to need a buffer for what's coming next?”



Buying food in *Oregon Trail*.



Elixirs in *BomberBoy* expire if you take them into a level but don't use them. The interesting gameplay pattern that this induces was first pointed out to me by Jeremy Parish in his YouTube series *GameBoy Works*.

We can compare this to having a fixed amount of food or a fixed number of HP potions. If I decide the player gets 10 HP potions for an upcoming challenge and give them no opportunity to obtain more, I'm fixing the difficulty dial for them. If I give the player some opportunity to stock up potions, they themselves get some control over this dial, albeit at a cost. Confident and risk-taking players can give themselves a greater challenge by betting they'll need little or no buffer, while more conservative or timid players can spend greater resources to buy an easier challenge.

Like breaking equipment in games like *The Legend of Zelda: Breath of the Wild*, item expiry is a contentious issue; many players do not like when hard-earned and relied-upon items expire. I also dislike item expiry in games like *Baten Kaitos* where my inventory is huge and items expire according to their own internal timers; I don't want to devote any of my limited brain capacity to the fact that banana #1 goes bad tomorrow, bananas #2 and #3 go bad in two days, etc.

But I don't mind the expiry systems in *BomberBoy* or *Mystia's Izakaya*; in these games, it's easy to reason about expiry and so it's never a surprise that an item has gone bad. Each of them has a central "cycle" that's crucial to the gameplay (in *BomberBoy*, attempting a level; in *Mystia's Izakaya*, a day), and items last exactly one cycle. Upon deciding that you need to provision an item, you have to use it that cycle or it disappears.

Theft and hacking

The character attempts to obtain or access items illegally, without being detected.

Examples

Theft is a common mechanic in STEALTH (p. 49) games going back, of course, to *Thief: The Dark Project*. Many RPGs have Thief classes as well, who can steal items from enemies, and sometimes enemies that can steal items from the party.

In many rogue-likes/lites including *Nethack* (1987) and *Spelunky* (2008), you can steal from shops, but in doing so you make the shopkeeper your enemy. (And the shopkeeper is a powerful enemy.)

Lockpicking and hacking are common themes for minigames in kitchen-sink RPGs like *Fallout* and the *Elder Scrolls* series. For just some of the variety of minigames that have been themed as “hacking”:

- *Bioshock* (2007) offered a *Pipe Dream*-like minigame combining CONNECTION-MAKING (p. 28) with TILE MATCHING (p. 190).
- The hacking minigame in *Ratchet & Clank* (2002) is a riff on *Tempest* (1981). Other games in the series offer a variety of different puzzles and small action games.
- *Fallout 3* (2008) has the player playing a QUERY (p. 208) game much like *Mastermind* (1970) or *Wordle* (2018).

Contraband

Some items are illegal to possess, with penalties if the player is caught.

Examples

Contraband is a common inclusion in TRADING (p. 102) games going at least back to *Drug Wars* (1984) and *Elite* (1984).

Smuggling was both an important mechanical and story element in *Wing Commander: Privateer* (1993); the plot is kicked off by player coming into possession of a stolen alien artifact.

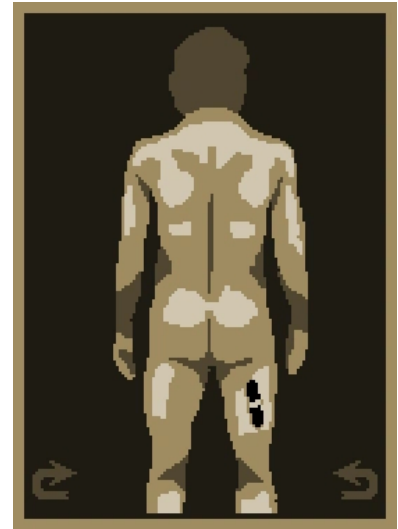
Sometimes the player is on the other side, as in *Papers, Please* (2013) where the player is a border guard who can scan players for smuggled objects.

Analysis

Contraband adds an additional risk/reward element to trading games. If you have the money, why not dump it all into expensive luxury cargo that will turn a high profit? Making some of that cargo illegal, and subject to seizure if found, helps to make sure the most profitable choices come with added risk.



"Brilliance" is an illegal drug in the *Privateer* universe, and if militia ships find it during a scan they become hostile.



A scan in *Papers, Please* reveals drugs attached to a smuggler's leg.

Collectable game components

The player finds and acquires components of a minigame (often a collectable cardgame) and uses them to play against NPCs.

Examples

Triple Triad in *Final Fantasy VIII* (1999) was (to my knowledge at least) the originator of integrating collectable cardgames into broader RPG adventures. According to director Yoshinori Kitase:

This time around, I wanted to make a minigame that was persistent throughout the entire world and could be played anywhere. At the time, *Magic: The Gathering* had just come out and was very popular, so we thought, “Rather than add a card game as a minigame, what if we added in a card game that all of the people in the world played? Some sort of tradition or cultural element that had been carried on from years past?” And we thought by adding that, it would add to the development of the world.

Pazzak (2003) in *Star Wars: Knights of the Old Republic* is another early example of this; *Gwent* in *The Witcher III* is a notable recent example, and was so popular that it was later released as a standalone.

Analysis

In this book, I’ve mostly not treated minigames and other games-within-games as a separate pattern of their own. They’re games, they’re already covered in this book because this book covers games.

But games like *Triple Triad* are different, setting up a second thread of progression through the game. The cards are items that you find in the world, and your opponents are NPCs in the world; your ability to progress in the minigame are thus intimately tied to MAP EXPANSION (p. 228) in the host game. And the minigame can in turn provide host game rewards; *Triple Triad* cards can be turned into rare items and powerful magic. For some players *Triple Triad* becomes the core gameplay of *FFVIII*; combat becomes an activity done to support *Triple Triad* rather than the other way around.



Triple Triad in *Final Fantasy VIII*

Joe Juba. 2019. “Back To School: The Stories Behind Final Fantasy VIII.” In *Game Informer* #319.

Part XIII

Equipment

Equipment

Some items can be held or worn by the player to increase their stats and/or give them additional abilities.

Examples

So many games have equipment that it's basically impossible to choose just one.

Analysis

Equipment systems, at their best, allow the kind of mix-and-match gameplay seen in MULTICLASSING (p. 241), in which the player feels clever for finding powerful synergies.

On the other hand, equipment in some games is just another way to express gaining +5 DEF (e.g., a leather shirt adds 15 to DEF, and a bronze cuirass adds 20 to DEF). This means that, upon finding a bronze cuirass, there is no genuine decision to make: you should definitely equip it. Since item management UI is often cumbersome, this is a lot of effort to make a decision with an obvious answer.

To make these decisions real ones, there should be some additional properties to equipment beyond a single stat, so that there's something in the "lesser" item's column. Maybe it offers an EQUIPMENT-BASED ABILITY (p. 178), or is part of a powerful EQUIPMENT SET (p. 179), or just looks cool (p. 180).

That's not to say that *every* piece of equipment has to be special and contribute to a interesting optimization puzzle. I don't necessarily want Meaningful Choices every single time I put on a shirt, and many shirts just being +N DEF is totally fine. It's when *every* piece of equipment works this way that we have to ask, "Why have an equipment system at all?"

Granted, I can see the attraction in minimalist equipment systems where each kind of equipment is nothing more than a single kind of stat boost, but here I would suggest putting an equipment *theme* on a simpler upgrade system like PSEUDO-EQUIPMENT (p. 177).

As I mention in SORTING (p. 194), trivial choices with cumbersome interfaces is a pet peeve of mine.

Pseudo-equipment

Character upgrades are themed as being equipment without actually being items in an inventory.

Examples

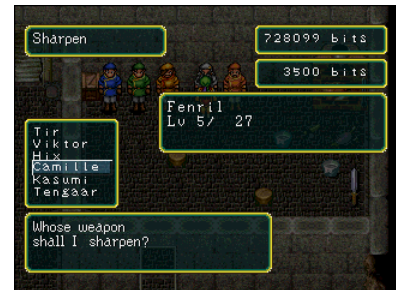
Weapons in the *Suikoden* series are not actually equippable items. The player eventually collects up to 108 characters, and juggling everyone's weapons as actual items would be a slog. Characters are themed as having weapons but it's really a kind of level that adds an attack bonus. Visiting a blacksmith in *Suikoden* and paying for "sharpening" just upgrades this level and sometimes results in the weapon changing names.

Weapons in *Final Fantasy VIII* are themed as CRAFTING (p. 163), but like *Suikoden* there are no actual weapon items to juggle. Once you've collected the ingredients for a higher-level weapon, you go to a "junk shop" and exchange those to "remodel" your weapon.

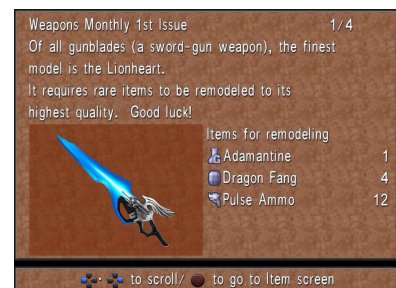
Actually, if we go *very* far back to the dawn of computer/console RPGs, before the rise of the basic "Equipment are inventory items" pattern in the first place, pseudo-equipment is unremarkable. When you buy new equipment in *Dragon Quest*, your old equipment just disappears, it's not an inventory item you can turn around and sell.

Analysis

Some non-violent adventures/RPGs retain traditional "equipment" systems, but they're often a bit anemic – the games don't have much in the ways of stats/abilities, and a piece of equipment often just boosts a single stat. The system is there, but it doesn't really add any interesting decisions to the game. As an alternative, putting an equipment theme on a level can give you some of the benefits of equipment (e.g., something to sink gold into, something to put in treasure chests) without some of the downsides (e.g., having to program an inventory system, making the player scroll through inventory screens, etc.).



Suikoden



Final Fantasy VIII

Equipment-based abilities

The player can only use/learn particular skills when equipped with particular equipment.

Examples

Obviously, lots of abilities in games are themed as equipment (e.g., Samus's ability to move freely underwater is tied to a suit), and many pieces of equipment provide various buffs like cold resistance.

A particularly interesting variant of this combines equipment with RESEARCH (p. 236) (at least in the way I'm using that word). In *Final Fantasy IX* and some subsequent games, abilities are not apportioned on level up, or bought with EXP directly. Instead, most skills are associated with equipment, like a "Counter" skill being associated with a martial arts belt. The player can use that skill so long as that equipment is equipped, but they also gain "Ability points" in battles, which fill up a gauge in each piece of equipment.

Once the player has earned the requisite amount of AP, they learn that skill permanently, even if the equipment is unequipped.

Analysis

Equipment-related abilities and buffs are another possible answer to the question, "Why would I equip a piece of equipment except for its numerical superiority?": because I need its associated ability. *FFIX* adds an additional answer on top of that: I might not need the associated ability at the moment, but am still in the middle of learning the ability for the future. During quite a lot of *FFIX* I didn't have the currently-most-powerful equipment equipped, because it was more important to learn the abilities of "lesser" equipment.

One caveat is that this system makes equipment very important, perhaps too important. In most RPG-like games, missing a piece of equipment usually isn't a big deal; something better will come along soon. Missing a piece of equipment in *FFIX* might mean never learning a crucial ability.



In *Final Fantasy IX*, equipping the "Mage Masher" allows the character Zidane to learn the skills "Detect" and "Flee".

Equipment sets

Equippable items come in sets where, if the player equipped multiple of the same set, they gain additional bonuses.

Examples

It's probably not the first example, but a notable popularizer of this pattern was *Diablo II* (2010). For example, if you find and equip the Arctic Mitts, Arctic Bindings, Arctic Furs, and Arctic Horns, you receive a rare buff that makes you immune to freezing. Originally, you only got bonuses if you equipped all of the items, but bonuses for partial sets were added in a DLC.

Analysis

Equipment sets are yet another answer to the basic problem of equipment mentioned on page 176 (e.g. when you're wearing armor that grants +5 DEF and you come across one that's +10 DEF, there is really no decision to be made; you should absolutely equip the new armor).

Equipment sets help to deepen the equipment puzzle, by giving the player a different route towards powerful equipment beyond just equipping the item with the highest stats.

When equipment changes the appearance of the player avatar (cf. DRESS-UP, p. 180), equipment sets also exhibit an aesthetic consistency; this can be another motivation for the player to "complete the ensemble".



Diablo II

Cosmetic items and dress-up

Items change the visual appearance of a character, often giving the player an opportunity for creative self-expression.

Examples

Dress-up is one of the roles of EQUIPMENT (p. 176) and many games allow equipment to change the appearance of characters. This is particularly noticeable in Squaresoft/Square Enix’s approach to female RPG protagonists; in *Final Fantasy X-2* (2003) and *Lightning Returns: Final Fantasy XIII* (2013) the entire CLASS (p. 239) system is expressed as “dresses”.

Equipment with visual differences but no gameplay consequences are often called “cosmetic items”, “skins”, or “hats” (even when they are not literally hats), and are a very common monetization opportunity in online multiplayer games like *Fortnite* for two reasons. Since these games are ones where other players see you, it provides incentives for looking unique; simultaneously, since cosmetic items don’t change the gameplay, it avoids the “pay to win” problem that can cause non-paying players to check out of the game. But these also occur as DLC and GACHA (p. 250) in single-player games as well, particularly for the purpose of fan-service (e.g. “swimsuit DLC” in girl-focused series like *Atelier*).

There are also a slew of dress-up and makeover-centered games like *Project Makeover* on the “girl games” side of the internet; these are often mechanically very similar to the surgery games discussed in MEDICINE AND HEALING (p. 58).



Fortnite



Final Fantasy X-2



Project Makeover

Disguises

The player wears special costumes in order to change NPC reactions to them and access inaccessible areas.

Examples

Disguises are common in stealth games, like *Hitman* or *Monaco: What's Yours is Mine*, as the player finds uniforms and disguises that allow them to infiltrate areas and have conversations they otherwise could not have.

These outfits might be found in one fell swoop, or assembled piecemeal; a significant early plot arc in *Final Fantasy VII* involves collecting pieces of an outfit in order to disguise the male protagonist as a woman to infiltrate a brothel.

Analysis

At minimum, disguises offer a different theming to a LOCK-AND-KEY PUZZLE (p. 183). Rather than requiring a red key to pass the red door, the game requires (say) a janitor's uniform to get past a guard.

But beyond that, disguises also let the player role-play in a wider variety of roles, and experience new NPC CHATTER (p. 200). For example, *Dragon Quest III* has a memorable segment where the king lends you his outfit and you can go around town role-playing as the king, learning information you wouldn't have as an ordinary person. If your game involves disguises and no NPCs change their reaction (except for the guard you're trying to get past), that's a missed opportunity to add some flavor, clues, or humor.



Hitman (2006)



Even Mario has begun wearing disguises to infiltrate inaccessible areas in *Super Mario Odyssey*. Why is this even in the game? I suspect it's because the game is full of coins, and now that they're not tied to lives there's not actually a whole lot to spend them on. So they introduce COSMETIC ITEMS (p. 180) as a coin sink, and then to encourage you to buy them they make some required to reach special areas.

Part XIV

Puzzles

Lock and key puzzles

The player is confronted with some sort of obstacle that prevents further progress, and must find an item that eliminates that obstacle.

Examples

Sometimes these are literal locks and keys – in *DOOM*, the red key unlocks the red door and that’s as complex as that gets.

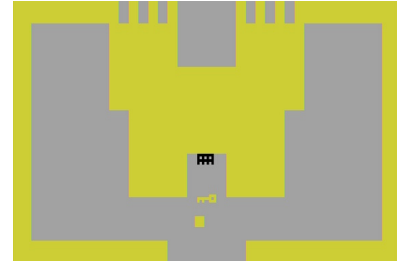
On the other hand, sometimes the lock and key are themed as other things, and the overall puzzle is to figure out which things are “keys” for which “locks”; this was the central puzzle type in the graphic adventure games popular in the ‘80s-’90s. Try *Day of the Tentacle* (1993), a classic of the genre.

Also, note that just because an item is themed as a key, it might not be a “key” in the “lock-and-key” sense. Small keys in *Zelda* games are not keys; any key unlocks any door, and they can only be used once. That’s not a key in the usual real-world sense of “key”! They’re more a form of restricted MONEY (p. 237) that can only be spent on opening doors.

Analysis

This was the central mechanic of the 1990s graphic adventure game, especially those in the tradition of LucasArts. (Earlier text and graphic adventure games offered a somewhat more diverse set of challenges, but as time went on the genre began to center primarily on inventory-based lock-and-key puzzles and DIALOGUE TREES, p. 199.)

While the genre is not dead (studios like Wadjet Eye continue to make excellent ‘90s-style graphic adventure games), it has certainly become niche. The usual diagnosis for its decline is that it was too easy for the player to become stuck, especially when the game requires lock-key combinations that may be obvious to the designer but are “moon logic” to the player.



Adventure (1979)



Day of the Tentacle



Gabriel Knight 3 (1999) is notorious for an absurd sequence puzzles in which you assemble a mustache to disguise yourself as a man who does not have a mustache. The classic reference for this is Old Man Murray’s scathing postmortem for the genre, “Death of Adventure Games” (*Old Man Murray* #77, Sep. 2000.)

After all, if you have a game with 100+ of these puzzles in a row, *some* percentage of them just won't occur to the player. If all of those puzzles are on the critical path, the player has no alternative but to rub everything in their inventory on everything else in the game. (And they typically *were* all on the critical path; these games were basically just sequences of lock-and-key puzzles and if they weren't on the critical path, there's be no game.)

This is true for even a well-constructed graphic adventure, and some were not. In terms of puzzle design and game structure, *King's Quest V* is often held up as an example of what not to do. It is very easy to render the game unwinnable by doing, or failing to do, seemingly minor actions that will lead to your death in the future, with no in-game clues to suggest that those events might be connected. At one point during a mountain climb, the game warns you that you're getting hungry; if you eat an obvious inventory item for this (a pie), you have actually rendered the game unwinnable, because in the near future you were supposed to throw it at a yeti to cause it to fall off a cliff. Many players simply continue to wander the world in search of a way to get past the yeti, not knowing they would never find one. In another notorious example, if you *failed* to throw a shoe at a cat during a roughly five-second animation, hours later you would starve to death tied up in a cellar.

In any case, the traditional graphic adventure game fell out of style for reasons both deserved and undeserved. But inventory puzzles still remain a component of many games, from JRPGs to hidden-object/adventure hybrids, and it's valuable to understand some of the wisdom distilled by that generation of designers, as they created these games and theorized about their construction.

One particular maxim is especially worth remembering: "no backwards puzzles", "lock before key", or "always show the goal".

The backwards puzzle occurs when the solution is found before the problem. Ideally, the crevice should be found before the rope that allows the player to descend. What this does in the player's mind is set up a challenge. He knows he need to get down the crevice, but there is no route. Now the player has a task in mind as he continues to search. When a rope is spotted, a light goes on in his head and the puzzle falls into place.³

Players should see their next goal (or encounter an intriguing mystery) before they can achieve (or explain) it.⁴



In *King's Quest V*, this yeti is a "lock", and the "key" is pie. Let's hope you didn't already eat the pie, because otherwise you've soft-locked the game.



Meanwhile, if you didn't throw a shoe at a cat earlier, to stop it chasing this rat, the rat will not show up to save you in this cellar.

³ Ron Gilbert. 1989. "Why adventure games suck." *Journal of Computer Game Design*, 3(2).

⁴ Spector, Warren. 2004. "Warren Spector's Commandments of Game Design." On gameindustry.biz

Permutation puzzles

The elements of a puzzle can be rearranged according to certain algebraic rules. The puzzle field typically has some initial state of order (e.g., with matching colors or a coherent picture), the pieces are scrambled, and the player attempts to reconstruct the original state.

Examples

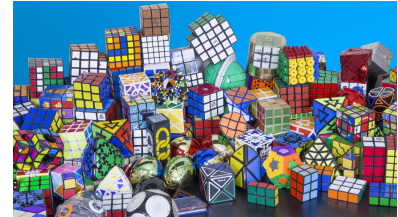
The all-time classic permutation puzzle is the *Rubik's cube*. In trying to solve a Rubik's cube from scratch (that is, working out how it's solved on their own, not simply executing someone else's algorithm), the player is looking for sequences of moves that swap some blocks for others, and leave others in place. (If you've tried to solve one from scratch, you certainly know the frustration of solving a side, working on another, and in the process obliterating your progress on the first side.) By discovering sequences that leave their prior progress in place, they can eventually solve every side.

Sliding tile puzzles are an even older kind of permutation puzzle; the classic *Fifteen Puzzle* was invented in the 1870s, and the *Moving Puzzle* or *Dad's Puzzler* around 1900.

Analysis

The benefit of a permutation puzzle is that it's easy to implement new puzzles. Since every transformation can be undone, you can just make random moves beyond the point where a human could remember them; the solution is just however it was scrambled, but in reverse. This is why they make such evergreen physical toys: you don't need a book of solvable initial states, just the toy itself.

The downside of permutation puzzles, as least as part of a larger game, is that they can be quite difficult. Sitting down to actually work out your own solution to a Rubik's cube is a pretty substantial hobby project – possible, but difficult, and much more difficult if you don't understand *how* you go about inventing a solution.



Rubik's cubes have since evolved into a dizzying array of puzzles; pictured here is the collection of YouTuber RedKB.



This *Fifteen Puzzle*, simply entitled "Puzzle", was the only game to ship with the original 1983 Macintosh computer.



Prismatic (2015)

If you haven't seen someone work out how to solve a Rubik's-like puzzle from scratch, I recommend the YouTuber *Superantoniovaldi*. But I want to emphasize that this is the thought process of a person who has solved *many* twisty puzzles before and understands the method by which you search for these solutions.

Permutable levels

The player can permute spaces on the gameboard or level as if it were a PERMUTATION PUZZLE (p. 185).

Examples

Labyrinth (1986) is a classic permutational boardgame, in which the players change the maze-like gameboard by sliding rows of tiles.

Many puzzle/adventure games have levels where elements rotate or slide, and some games are based entirely around this. *Continuity* is a small puzzle platformer game that takes place inside a sliding puzzle.

The battle system in *Paper Mario: The Origami King* involves a permutation puzzle in which you line up enemies on rotating rings, so that you can eliminate them in one fell swoop rather than tediously pick them off one-by-one.

Enen Angel (2001) uses a similar rotational gameboard to *The Origami King*, but instead of there being one intended solution, the goal is to rotate the rings so that a character can visit every square à la 1980s AREA CONTROL (p. 25) arcade games.

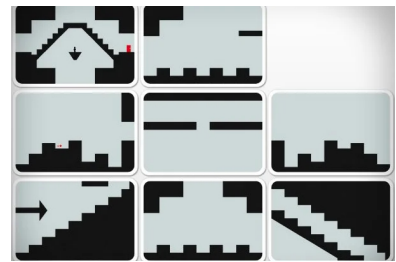
Analysis

One difficulty with permutation puzzles is that they offer only limited incremental rewards – unless you really understand a *Rubik's cube*, you can't easily build on partial progress. This is the same problem I mention in MAZES: if you're lost, you may be lost for hours without a shred of reward; it's really only upon solving the whole thing that you get a rush of satisfaction.

The difference between *The Origami King* and a mechanically-similar puzzle like *Prismatica* is that a complete solution is not required; solving the puzzle completely confers a huge advantage, but partial solutions (e.g., only lining up one of two groups) still confer a partial advantage. While most reviewers didn't especially enjoy *The Origami King's* battle system (and nor did I), I think it's a fascinating



Labyrinth



Continuity



Paper Mario: The Origami King



Enen Angel

solution to a design problem. It incentivizes even partial solutions to a permutation problem, by associating them with rewards in another game system (here, an RPG-style combat system).

Challenge

What would this solution look like in other genres of games? The “downstream” system could be anything, not just combat. For example, let’s consider putting *The Origami King’s* DNA into a rhythm game. In games like *Guitar Hero*, the configuration of notes is the whole of the level. We can imagine a rhythm game where you can rearrange notes according an algebra, and the song is partially and randomly scrambled. Reconstructing the original perfectly might result in the best-sounding and easiest-to-play song, but the game isn’t lost or unplayable if player doesn’t achieve the full solution.

What about a sports-themed game? Or a racing game? Or pinball? These are all genres where the spatial configuration of characters or environmental elements is crucial to the gameplay.

Ordering puzzles

The player has an inventory of actions or items, which often have causal interactions with each other, and the player must determine an ordering that achieves a particular result.

Examples

Many of these puzzles set a particular target result, and the player's task is to re-create it. The earliest example I can think of for this is *The Factory* (1983), in which a series of simple machines rotate, paint, or punch holes in a square of sheet metal. The player is given a target design, and must figure out what sequence would lead to that result.

Please, Touch the Artwork (2022) contains re-creation puzzles in which you attempt to re-create *De Stijl*-style paintings by adding lines and color fills in precise orders.

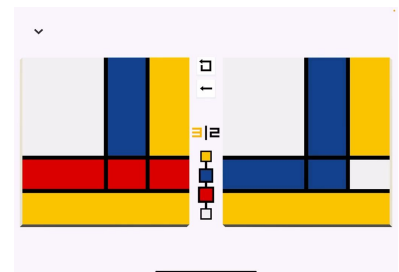
Marble Drop (1997) is an ordering puzzle with an almost LEMMINGS-like (p. 128) feel. The player drops marbles into an intricate "marble run" layout, and by running over switches, diversions, gears change the layout for subsequent marbles. Neither the marbles nor the mechanisms can be controlled directly; the only decision the player makes is when and where to drop marbles into the machine.

Another kind of ordering puzzle is the storyboard rearrangement puzzle, where you have to rearrange panels of a scrambled story. Cartoonist/puzzle-maker Robert Leighton is probably the master of this small genre; what makes his comic rearrangement puzzles so clever is that they sometimes have multiple orderings that make a *coherent* story, but only one where that story tells a *joke*.

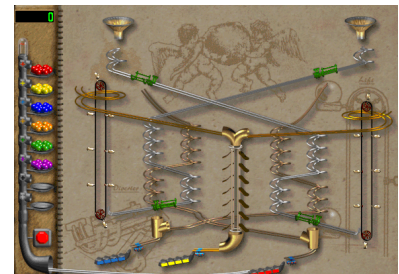
Another classic of this genre scrambles *A Short History of America* (1979), R. Crumb's classic illustration of an American small town changing through the centuries. Since older buildings and elements can be seen (often in dilapidated forms) in later panels, this gives sufficient clues to order the panels even if your knowledge of American history/design is limited.



The Factory



Please, Touch the Artwork



Marble Drop

I struggle to find an example of Leighton's comic rearrangement puzzles online, but there are some in the *GAMES Magazine Giant Book of Games* (1991), pp 16–17.

Analysis

Broadly speaking, these are a kind of PERMUTATION PUZZLE (p. 185), but permuting objects *in time* leads to different kinds of interactions and methods of solving.

While relatively few games *solely* involve ordering puzzles, many games' puzzles require careful attention to ordering actions to achieve a particular result. Among point & click adventure games, *Goblins 2: The Prince Buffoon* (1992) requires particular attention to the ordering and timing of actions. CHAIN REACTION (p. 134) and AUTOMATION (p. 133) games in particular require great attention to ordering; we can see *Shapez* (2020) as the child of *The Factory* and *Factorio*.

It's important when designing ordering puzzles to remember how quickly factorial numbers grow; if you have ten actions and only one correct order, you're looking for one solution among 3.6 million wrong ones. It's important that the player be able to deduce solid sub-orderings (e.g., "I know that these two items have to come next to each other, and these two" or "I know that these four items have to happen in this order, but I don't know if they're adjacent"). Deducing these sub-orderings winnows down possible permutations down to a reasonable level, and count as intermediate "wins". Without these, the puzzle can have the same pitfall as too-difficult MAZES (p. 96) or PERMUTATION PUZZLES (p. 185): a long slog without sufficient feeling of making progress.

The GROW games by On Nakayama are an interesting study in this regard. In these, you have a tiny world (e.g. a cube) and 6–10 actions that (typically) add things to the world. However, you do *not* know ahead of time what your goal looks like; unlike the games above, it is impossible to deduce the correct order ahead of time. Since trial-and-error is the only way to progress, this would seem to be a recipe for an impossible slog.

But the GROW games were among the most popular puzzle games of the Flash era, so what did they do right to avoid the pitfalls? One big reason they worked was their animations, which were numerous, charming, and functional. Every success resulted in an animated chain reaction that both felt good to discover and reinforced that the items were indeed meant to go in that order. This was also true in the opposite direction: failures were also sometimes interesting to watch, and likewise gave useful information. Exploring them can give similar feel to exploring the many ALTERNATE ENDINGS (p. 262) in games like, say, *Princess Maker* (1991).



GROW Cube (2005)

Tiling puzzles

The player is given tiles that they must arrange according to certain constraints in order to fill a space. Often, the tiles' edges have distinct shapes, colors, or symbols that serve as an obvious visual cue for which tiles are permitted to match.

Examples

This is the core pattern of the typical modern *Jigsaw Puzzles*, where all or most pieces are functionally squares, but the edge shapes and colors ensure that each piece only fits in a particular place. On the other end of the spectrum, other tiling games have only a few distinct edges and there are many more valid solutions, like *Dominos/Triominoes*.

This is widely used in MAP-BUILDING (p. 81) games where you build the board layout as you play, like *Carcassonne*, *Railroad Ink*, and *Dorfromantik* (2021).

Carto (2020) is a puzzle-adventure game in which the world is made out of *Carcassonne*-like tiles, and the player rearranges the map to solve puzzles.

Pipe-laying games like *Pipe Mania* (1989) (also released as *Pipe Dream*) are another variation of this. Here, the goal is not just to match up edges, but to make a coherent pipeline that can carry water without spilling it. Modern gamers may be more familiar with this as the hacking minigame in *Bioshock* (2007).

Usually these games involve drawing tiles from an unorganized “bag” of them, but another classic edge-matching puzzle (sometimes called the *Network* or *Circuit Puzzle*, but sometimes just called *Pipes*) involves fixed tiles that you can only rotate, with the goal of arranging the board into a coherent network with no non-matching sides.

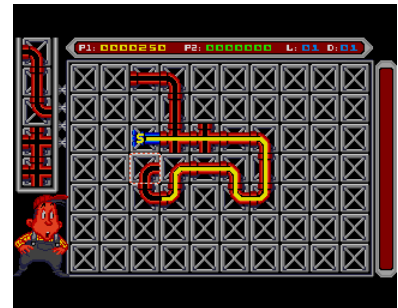
For examples of tiling games that are *not* based on edge matching, consider *Ishidō* (1990) or *Alchemy* (2001). In these games, tiles have both colors and symbols, and tiles are permitted to touch if they match in either color or symbol.



Carcassonne



Carto



Pipe Dream



Ishidō: The Way of the Stones

Analysis

Tiling problems are a venerable puzzle type (there's a whole branch of Discrete Mathematics devoted to them), and when the shapes and patterns are well-chosen the puzzle barely needs to be explained.

Carcassonne is perennially popular as a "gateway boardgame" for this reason; it's extremely easy to teach someone the core gameplay.

Part of the draw is that, by the end of the game, the player has created something intricate and beautiful. When the pieces allow millions of possible solutions like *Carcassonne*, that beautiful thing is also unique to that playthrough; it belongs to the players themselves. If you've introduced people to boardgaming through *Carcassonne*, you've probably noticed that new players sometimes like to take a picture of their creation at the end. If you're creating a videogame that results in a similar unique creation, don't neglect to make it easy to share on social media; it's free advertising!

In **CONSTRAINT SATISFACTION** (p. 34), I mention the difference between hard constraint puzzles (where the constraints must be completely satisfied or else the puzzle simply isn't solved), and soft constraints where you're trying to fulfill as many as possible but can violate them when necessary. In *Carcassonne* you cannot, by the rules, put tiles next to each other that do not match; in *Railroad Ink* you can, but will get fewer points in the end.

Soft constraint optimization puzzles like this also appear in city builder games like *SimCity* (1989). The player typically wants to place residential zones near parks and water to increase property values, and avoid placing them near industrial zones due to pollution. The builder *Reus* (2013) has particularly complex adjacency rules that make clear the shared DNA between builders and tiling puzzles.



Reus

Arrangement, packing, and dissection

The player is given shapes, often irregular, and must arrange them to fit them into a space or to match a pattern. Conversely, sometimes the player begins with an intact shape and must dissect it or slice pieces off to fulfill certain criteria.

Examples

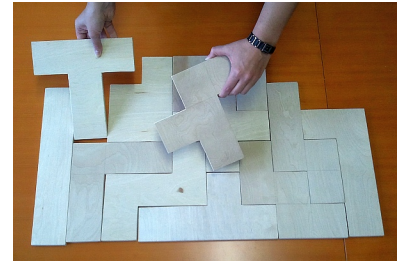
Everyone knows this pattern through *Tetris* (1985), although it should be noted that *tetromino* and *pentomino* puzzles had long been popular; the brilliance of Tetris was not inventing these shapes, but using them to create a novel kind of action-puzzler. Prior to Tetris, these shapes were typically used in packing puzzles (fit all the shapes into a particular diagram); for one of many videogame adaptations of this, see *The Island of Dr. Brain*. See also INVENTORY TETRIS (p. 167).

Digit (1987), also marketed as *Pick-it* and *Stix*, shows a way to adapt arrangement puzzles (in this case, *Matchstick puzzles*) into a competitive game. The players share a layout of sticks, each player holds cards representing possible layouts. If you can move one stick to match a layout, you can discard that card.

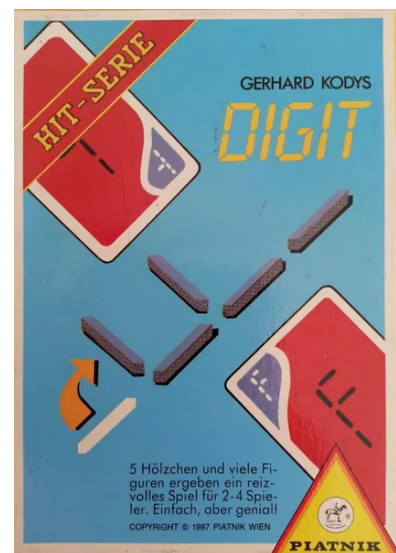
Dissection puzzles (at least how I'm using the term) start with an intact shape, and task the player to divide it into pieces that meet some criterion (e.g. "Divide this shape into four identical pieces"). These often shade into arrangement puzzles, e.g., "Divide this shape into two pieces and rearrange them into a square."

Dissection and arrangement puzzles are very ancient and are tied up closely with the development of geometry. (Consider, for example, visual proofs of the Pythagorean theorem, or the long history of attempting to "square the circle".) *Tangrams* have an unclear origin but probably stem from classical Chinese geometry; similar ideas can be seen in rearrangeable banquet tables popular in the Song dynasty.

Snipperclips (2017) is a multiplayer dissection puzzle in which each player controls a shape. The two shapes can overlap, and each player can snip the intersection out of the other, with the goal of finding appropriate shapes to solve a variety of challenges.



Pentominoes; photo © Museum of Math.



Digit



Snipperclips

Box pushing

The player must push boxes in order to get them to an assigned destination or clear a path.

Examples

The classic example of this is *Sokoban* (1982); many also know this game by its GameBoy release *Boxxle* (1989). So many box-pushing puzzlers were released on GameBoy alone that it is difficult to even list them, but if you need an example I recommend playing the faster and more diverse *Kwirk* (1989).

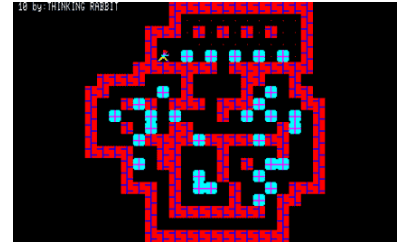
Several indie developers are currently known for their box-pushing puzzle games, especially Alan Hazelden, creator of *Sokobond* and *A Good Snowman is Hard to Build* and Stephen Lavelle, creator of *English Country Tune* and *Stephen's Sausage Roll*.

Analysis

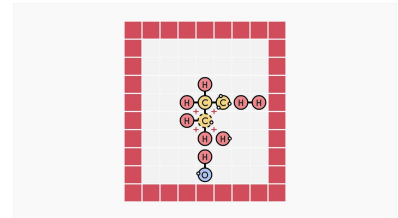
I have to admit I don't like box-pushing puzzles very much, simply from being over-used in too many games. On the other hand, it is still possible to write excellent puzzles even in an over-used medium, as Alan and Stephen have shown. One of the best puzzle games of the 2010s, *Baba is You*, is a box-pushing game at its core.

If you'd like to rapidly prototype a box-pushing mechanic (or something that fits broadly in this vein of grid-based puzzles), you should try Stephen's *Puzzlescript* language. It cleverly expresses interactions as rules in a cellular automaton, and if you can wrap your head around it you can prototype a surprising variety of puzzles.

One interesting property of block-pushing puzzles is that they're non-symmetrical: if you are pushing blocks out of the way in an attempt to go from east to west, and later return going west to east, it's actually a different puzzle. This is utilized cleverly in *The Legend of Zelda: Link's Awakening*; when you first go through a particular cave, there's an easy box-pushing puzzle, but when you return in the opposite direction, the same layout serves as a slightly harder puzzle.



Sokoban



In *Sokobond*, the boxes are chemical elements, and they form bonds with each other according to their valences.



In *A Good Snowman in Hard to Build*, the boxes are snowballs, and they grow as you push them.



This puzzle in *Link's Awakening* is harder the second time through, simply by coming at it from the opposite direction.

Sorting and organization

The player has to repeatedly put items in the right containers, divide items into different types, or put them in a particular order.

Examples

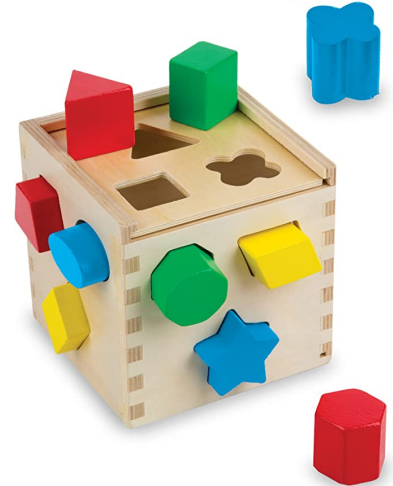
The prototypical sorting activity might be the *Shape Sorter*, a child's toy in which the player attempts to put shapes through appropriately-shaped holes.

A meditative and surprisingly compelling take on this for adults is *Unpacking*, in which the player unpacks boxes upon moving to new apartments, and puts each thing in an appropriate place.

I Love Hue (2017) is a sorting game in which the player sorts only very-subtly-different color patches into gradients; again this supports a meditative experience.

Abstractly, many games have shape-sorting as part of their core interaction loop. In *Minecraft* and many crafting games inspired by it, the basic interaction you do throughout the game is to press a button to collect an environmental resource. This is given some slight cognitive friction, however, by the fact that you need to match the right tool to the right resource: trees require axes, dirt requires shovels, stone requires pickaxes, etc. This is themed differently than “put star-shaped piece into star-shaped hole” but you can see the mechanical similarity.

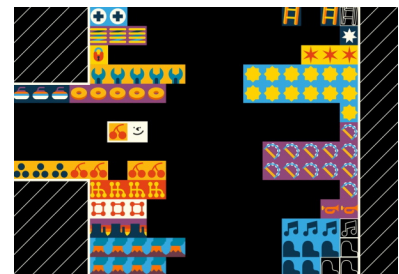
A more complex example of sorting is seen in *Wilmot's Warehouse* (2019), where you are a warehouse worker fulfilling orders in a warehouse of semi-abstract shapes. Here, there are no specific repositories for each shape, you can put them anywhere, but when orders come in you have a limited time to fulfill them, and if you haven't sorted them in *some* manner you'll never make it in time. The challenge of the game in part comes from *deciding* which shapes you want to put together so that you can remember their organization when you need it, and coping with new arrivals that upend your prior categorization decisions.



Melissa & Doug Shape Sorting Cube



Unpacking



Wilmot's Warehouse

Analysis

Abstractly, we might consider many instances of this pattern to be a sort of LOCK-AND-KEY (p. 183) puzzle, just one in which the locks and keys are usually fairly obvious and done as a repeated activity. Sorting is typically a low-intensity cognitive activity, and often supports the kind of gameplay where the player zones out and half-thinks about other things.

I admit that I don't typically like shape-sorting gameplay in the *Minecraft* sense; in games where I harvest 10,000 things I don't want to solve a trivial puzzle each time. Part of this, though, is that I find the "toolbar" interface (where you select one of a horizontal collection of items using number keys or controller shoulder buttons, and your selection changes what the X button does) to be a cumbersome and error-prone way to do what's effectively shape-sorting. It's a case where you put more cognitive effort into using the interface than you did solving the "puzzle" in the first place.

On the other hand, I find *Wilmot's Warehouse* to be a very clever take on shape-sorting gameplay. The player is not mechanically required to do a child's puzzle; their only goal is performing time-limited fetch quests. This goal can only be achieved if the player *stays organized* in some way, but the game doesn't mandate exactly how.

Matching puzzles

The player is tasked with finding pairs, groups, or sequences of matching items, often removing them from play.

Examples

Match-3 and similar genres of casual games dominate the mobile/tablet space right now; in them, you typically swap adjacent items (e.g. marbles) to make lines of 3 or more. The forgotten progenitor of these games was *Shariki* (1994); the first implementation to be a big hit was *Bejeweled* (2001).

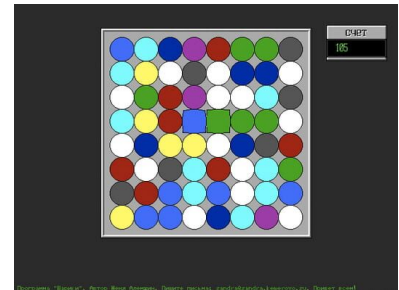
In other varieties of this, the objects are already in large groups and the player needs only select them; this is the core mechanic of *ChainShot* (1985), *Puzzle Bobble/Bust-a-Move* (1994), and *Azkend* (2010).

Pair finding finds its simplest implementation in *Mahjong solitaire* – not the actual four-player Chinese game, but the puzzle where mahjong tiles are piled up and the player must remove them in matching pairs. The matching tiles don't necessarily have to be adjacent, just fully uncovered; the main loop of the game is finding a pair, removing them and possibly revealing more tiles, and repeating. *Pyramid solitaire* is a similar example, although with a slightly more complex matching rule: pairs of cards may be removed if their numbers add up to 13.

Pair-finding also underlies the perennial childhood game *Memory*, in which cards lay face down, the player reveals them in pairs, and only can remove them if they match. If they don't match, the player replaces them face down; the challenge is remembering where the appropriate card is when its soulmate is later revealed.

Analysis

Matching and SORTING (p. 194) are quite similar on a moment-to-moment basis, since both typically involve putting together groups of objects that are the “same” or “match” in some way. Even so,



Shariki



GNOME Mahjong



Chain Shot! Variants of this are commonly known by the name SameGame.

the structure and feel of the games as a whole strikes me as rather different, so I put them in different sections.

Often in Match-3 and similar games, the matching objects disappear and are replaced by random replacements; the player is thus fighting against the slow march of entropy as possible matches are exhausted and the board becomes more chaotic (cf. PERSISTENCE, p. 38). In this way, despite the mechanical similarity, adjacency matching has the opposite feel from SORTING, in which the playfield generally goes from chaotic to ordered.

The visual complexity of the pieces/tiles is an important dimension for the designer to consider. Games where you are matching large numbers of adjacent tiles generally have relatively few, simple designs – if the tiles were as diverse as Mahjong tiles, the game wouldn't work because there would be too few matches. On the other hand, if a Mahjong solitaire game only had 5 distinct tiles, it would be too easy to find pairs. So while Match-3 and Mahjong solitaire games support a relatively chill, low-engagement gameplay loop, they exercise different cognitive capabilities. Mahjong solitaire stresses your visual recognition system more, and thus has a feel more like a HIDDEN OBJECT (p. 155) game.

Taking this idea to an extreme, the puzzle/art book *The Ultimate Noah's Ark* (1993) combines pair-finding and hidden objects to create an extremely difficult puzzle contest. A giant painting includes hundreds of animals, each one having a mate somewhere in the drawing, except for one lonely animal. It took two years for someone to find the unmatched animal, winning a \$10,000 prize.



The Ultimate Noah's Ark

Part XV

Dialogue patterns and language puzzles

Dialogue trees

The player navigates a potentially-complex dialogue, usually but not always structured as a tree, to achieve some goal.

Examples

Dialogue trees are very common in games, so it's hard to decide on specific examples.

Star Control II (1992) was in many ways the culmination of the “fly around and talk to aliens” genre beginning with *Starflight* (1986). While you do a wide variety of things in-game, much of the game focuses around your conversations with strange and memorable alien races (like the Orz described in LANGUAGE LEARNING, p. 204), trying to glean information or convincing them to fight alongside you.

Dialogue trees were also very prominent in LucasArts-style graphic adventure games; an example with a particular focus on dialogue was *Indiana Jones and the Fate of Atlantis* (1992). In this game, the player can choose one of three routes (Fists, Wits, or Team), each emphasizing a different approach; the Team path is particularly centered around dialogue puzzles.

For a recent example outside of the typical LucasArts-style adventure, *Disco Elysium* (2019) was more inspired by Infinity Engine-style games like *Planetscape: Torment*, but the gameplay is almost entirely centered around navigating dialogue trees. Even combat in the game is handled by dialogue-tree-like choices.

Analysis

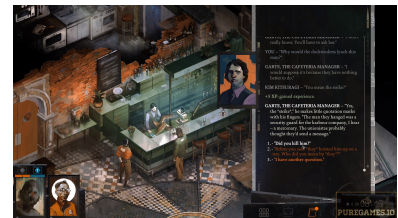
In the abstract, most conversation challenges like this are MAZES (p. 96), although in a highly multidimensional space. Fortunately, the player has an intuitive grasp of this complex space, by virtue of understanding language and how conversations work. There are thus opportunities for wit and surprise that typical mazes lack.



Star Control II



Indiana Jones and the Fate of Atlantis



Disco Elysium

NPC chatter

NPCs supply short monologues – often one sentence – to give directions, hints, local color, or reactions to player accomplishments.

Examples

Many games have this, and in modern games NPC chatter is often just optional flavor-text, but if we go back to earlier games – especially early JRPGs like the original *Dragon Quest* (1986) or *Final Fantasy* (1987) – NPC chatter is the *primary* way of directing the player. Neither of these games even have an introductory cutscene; to figure out where you’re supposed to go and what you’re supposed to do, you have to ask around. Some NPCs tell you directly where to go, others only provide partial clues; in *Dragon Quest*, for example, one location is only clear when you’ve combined the information in three clues.

Another important part of NPC chatter is to react to the player accomplishing milestones – thanking them for saving the town, remarking on how things are better, etc.

Analysis

I’m calling this “chatter” because it’s different from “dialogue” in the sense of *DIALOGUE TREES* (p. 199). There aren’t *two* people talking here, so really this is monologue. But “monologue” has come to have the connotation of length, and these are typically quite short, so “chatter” it is.

This may seem too minor a pattern to include, but I think it only seems minor because we have gotten used to random NPC chatter being optional flavor-text. Games mostly direct and motivate the player by other means now: quest lists, map pins, etc., and players have been trained by long experience to assume it’s safe to click through NPC chatter.

If you want the player to really pay attention to NPC chatter, you have to make a special effort to retrain them. Perhaps have several



Directions like these were some of the only guidance players received in *Dragon Quest*.

Even then, many players had never played an RPG before and were baffled how to actually play. Eventually, designer Yuji Horii, who was also a writer for *Weekly Shōnen Jump*, published a detailed guide (and effective advertisement) for the game in that magazine. Ultimately, the success of *Dragon Quest* was not just because of its quality – and it *is* a much higher-quality game than prior console RPGs – but because of the magazine’s enormous cultural reach.

Incidentally, that *Shōnen Jump* connection is also how they managed to get Akira Toriyama to design the game’s iconic characters.

early roadblocks that can only be overcome through information given by NPCs, and/or overgenerously reward the player for following NPC hints about treasure. And be parsimonious with chatter otherwise; don't fill the early-game NPCs with meaningless small talk that teaches the player to mash X every time text appears.

Outer Wilds does this very well; the game is frugal with revelations and there is rarely text that does not yield important information (although often this information is only understood in hindsight). As a player, you come to crave finding a new wall-scrawl, because the game has trained you in how valuable they are. The game's combination of an open world and frugality about clues is sometimes talked about as if it's revolutionary, but really this aspect is bit of a throwback to many RPGs and adventure games of the 1980s.

On a different note, Holleman (2019) points out a common pattern he calls "NPC irony". Much NPC chatter has two meanings, the meaning it would have in-world, as delivered by that character, and the meaning that the designer intended to convey to the player:

If you're outside a dungeon and someone says, "This is the haunted temple, don't go in – it's dangerous!" the designer is basically telling you: "This is a dungeon, make sure you're ready before going in." ...What the character tells you is very clearly the opposite of what the designer is trying to communicate with you. The ironic veil is usually pretty thin... but it's fairly essential, as it would be too easy for everything in the game to become a bland manual.

Actually, many ideas about "open world" design are throwbacks. You might think, from how the word is used today, that we started talking about "open world" design for games like (say) *The Elder Scrolls: Arena* (1994). But if you go back to read contemporary reviews of these games, nobody used that term or even really described the structure of their worlds as innovative. They talked about how *large* the game worlds were, but their structures weren't actually out of the ordinary, because this is how computer RPGs had always been. Even contemporary reviews of *Grand Theft Auto III* (2001) don't use it, believe it or not; "sandbox" yes, "open world" no.

It's only once *GTA3*-inspired ideas worked their way into console action games – genres that consumers *expected* to dole out their challenges in a restricted and mostly linear order – that people started talking about "open-world" design. The first uses I can find of "open" that are clearly the way we use this term today are in reviews of *Jak II* (2003) and *Spiderman II* (2004).

Patrick Holleman, 2019. *Reverse Design: Final Fantasy VI*.

A partial version of this is online at thegamedesignforum.com/features/reverse_design_ff6_5.html

Topic-based dialogue

The player has an “inventory” of topics of conversation; broaching the right topic to the right NPC is rewarded with more information and additional topics.

Examples

The first game I can think of with a topic-based dialogue system is *Ultima I: The First Age* (1981).

This isn't a rare pattern in itself, but it's fairly rare to see it systematized to the point where the whole game revolves around it. One exception is *Nomad* (1993), an open-world space trading and combat game in the same vein as *Starflight* or *Star Control II*. While the player is free to go through the game however they choose, the “main” route through the game is by information trading. You begin with one piece of information, track down someone who needs it or who can tell you more about it, and receive further information that you take to someone else. This slowly reveals the lore and central mystery of the world.

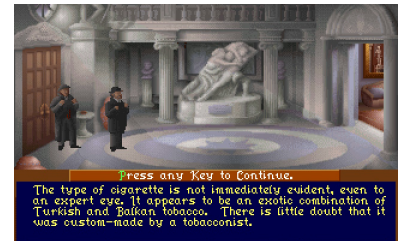
Detective-themed adventure games like *The Lost Files of Sherlock Holmes* or the *Blackwell* series also use topic-like information trading. The pieces of information may be realized as inventory objects (e.g. evidence), but unlike typical inventory puzzles, you're often not using these objects as tools, you're usually just showing them to the right people in exchange for more information.

Analysis

If DIALOGUE TREES (p. 199) are a sort of linguistic theming of MAZES (p. 96), then topic-based dialogue corresponds to LOCK-AND-KEY (p. 183) inventory puzzles. Even if both are mechanically equivalent to a non-dialogue kind of puzzle, the fact that they're expressed in text gives additional opportunities for them to flesh out lore, give hints, or establish character.



Nomad lets you ask anyone about anything or anyone in the game, or tell anyone any important fact that you've learned.



The Lost Files of Sherlock Holmes: The Case of the Serrated Scalpel



Blackwell Unbound

Nomad is nearly a forgotten game at this point, and it has some flaws, but there is a lot that it gets right and it's worth looking at its tricks in more detail.

The fundamental problem of an open-world information trading game is scale. There are hundreds of NPCs and hundreds of things to potentially ask them about. It's the basic problem of the LucasArts-style inventory puzzle: when the player gets stuck, are they going to test out every inventory "item" on everything in the game until something works? *Nomad* has to face that problem ten-fold because of how large its world is and how many different people and items are in it.

However, the designers of the game put substantial effort into the dialogue system of the game, and many NPCs are realistically helpful even if they're not the specific NPCs you need to give information to. There are only a few dozen actually-important NPCs who have unique dialogue and are important to the plot, but even the random bystander NPCs know things that it's realistic for them to know. They know political information like the location of their seat of government, their political leaders and alliances; they know other NPCs that it would be realistic for them to know, etc. "That's my cousin, actually. He's a trader like me and captains the ESS Stalwart; you'll usually find him around Newhome III."

Most NPCs don't care about most items, other NPCs, or pieces of information (and most of the alien cultures are themed as being culturally chauvinist, which makes it easier for the player to accept that most responses are some variation of "I don't know/care"). But even if they don't care, they can sometimes name someone who would, along the lines of "Oh, I don't care about all that weird foreign stuff. Maybe ask Bob-clo-cli-mar, he's probably into that." Then if you ask about Bob, they'll give you clues on how to find him. Bob is one of the actually-unique NPCs who you eventually need to meet to complete the main plot arc of the game.

The NPC dialogue thus forms a sort of graph that nudges the player towards plot-relevant nodes. This is crucial to keep a stuck player from randomly rubbing pieces of information against every NPC in the game.

This dialogue system was well-received by critics and the designers were justly proud of it, but in all it came to something like 10,000 lines of dialogue and took up much of the development time, so it's important to balance that labor against other things you might be doing. I just want to point out that *Nomad*'s detailed dialogue system isn't just for show; it's necessary within that game to scale up information-trading gameplay to a world of that size and detail.



Margins are for marginalia, so I'm using this margin for something I only learned when writing this. *Nomad* was originally meant to be a Disney-licensed game titled *Space Mountain*, in which the protagonist is riding EPCOT's iconic Space Mountain rollercoaster and discovers that one of the spacecraft inside is real.

I always felt that the introduction/backstory of *Nomad* was oddly mysterious and made little sense, and this is why: it was a last-minute addition when the Disney connection fell through.



The lobby of *Space Mountain* ride at EPCOT



The player's spacecraft in *Nomad*

Language learning

The player slowly comes to learn the language(s) of NPCs.

Examples

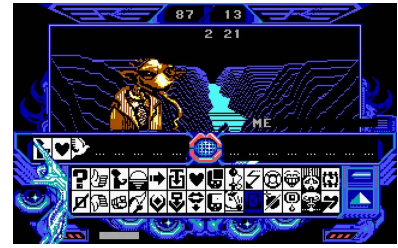
Captain Blood (1988) is one of the earliest examples of videogames where language learning is central to the game. In it, the player has a symbolic language of 150 ideograms that can be used to interrogate aliens about fugitive clones.

Although themed as learning a whole new language, videogame languages are usually just ciphers or calques of the player's language, in which letters or words are replaced in a one-to-one manner. This lets language learning be treated as a series of "collectables" and ensures that the player makes constant progress towards decipherment. *Final Fantasy X* (2001) provides an example of a simple letter cipher themed as a language, *Al Bhed*. By collecting "Al Bhed primers", each of which reveals one letter, the text slowly reveals itself as English (or Japanese in the original). *Out There* (2014) and *No Man's Sky* (2016) do the same thing at the word level.

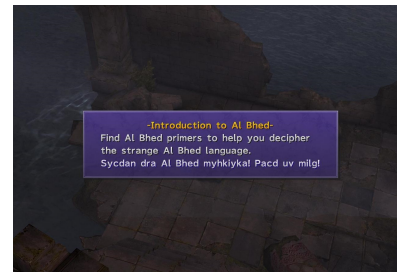
One point of difference is whether the game does the "bookkeeping" for the player and automatically replaces known elements (as in *FFX*), or whether players are expected to memorize or write it down for themselves (as in *FEZ*, 2012). An example of the latter at the word level is the Dudbear language from *Legend of Mana*. The protagonist has to sell lamps to bears that speak a simple language; the extremely limited vocabulary and structure of this language makes it more reasonable to keep in the player's head.

The interactive fiction game *The Edifice* (1997) provides a more realistic example of language learning. The player is trying to get to a mountain where a medicinal herb grows, but is blocked by a man with a spear, speaking an unfamiliar language. The goal is to learn his language well enough to talk your way past him.

In *Daikoukai Jidai/Uncharted Waters* games, TRADING (p. 102) RPGs in which the player is a ship captain in the Age of Exploration, the player can learn dozens of languages. If you interact with an NPC



Captain Blood



The designers of *Final Fantasy X* originally intended the Al Bhed language to be more linguistically complex, but decided a simple cipher worked better mechanically.



Learning the basics of the Dudbear language in *Legend of Mana*.



Uncharted Waters Origin

and none of your crew speak their language, the dialogue is obfuscated. Unlike the examples above, languages in these games are learned in one fell swoop by hiring crew or reading books, rather than on a word-by-word or character-by-character basis.

It is fairly rare that videogame “languages” force the player to wrangle with morphology (the segmentation of words into meaningful pieces like prefixes, roots, and suffixes). A rare example is seen in the JRPG *Rudra no Hihō* (1996), which I cover in more detail in *SPELLCRAFTING* (p. 206).

It is also fairly rare for a videogame language to require the player to think more deeply about the semantics of the language. Special mention here goes to the language of the Orz in *Star Control II* (1992). When first encountering the Orz, your translation computer warns you that their language is so unorthodox that it cannot fully translate all words, and some words will be represented by best-guess approximations marked by asterisks. During the game, the player comes to understand the meanings of Orz-isms like *dancing*, *bubbles*, and *fingers*, and through them gets glimpses of the real nature of the Orz. The resulting juxtaposition of silly comedy and cosmic horror makes the Orz one of the most memorable alien races in videogames.

Videogame languages are usually shallow calques on English, but it’s entirely possible to work out even more “exotic” language structures from raw data and limited clues. Check out some of the *Linguistics Olympiad* puzzles from ioling.org or naclo.org for examples of deciphering complex real-world languages with no prior background.

Mini-languages, like numeral systems, can even be embedded in games with few NPCs and little text, like the classic numeral puzzle in *Riven* (1997).

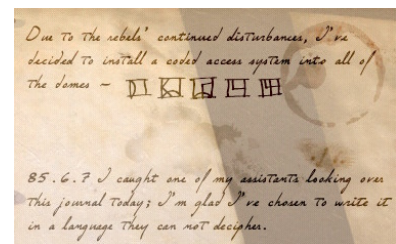
Analysis

Learning languages in an interesting progression pattern because it typically only “soft-gates” further progress. It’s different than needing to collect exactly 100 coins or 8 magic orbs before you can progress; just like in real-world foreign languages, you can often puzzle out the meaning of a sentence from context clues.

I (and, from what I gather, most players) never actually managed to find the Universal Translator to talk to the alien Shaasa in *Nomad* (1993), but by trial and error you can achieve functional proficiency in it well enough to trade, at least.



The Orz language in *Star Control 2* involves systematic mistranslations for which the player eventually deduces the true meaning.



In *Riven*, the player works out the clever D’ni numeral system.

Spellcrafting

The player uses a complex combinatoric system to execute magic spells.

This is, broadly speaking, a sort of LANGUAGE LEARNING (p. 204), but with an emphasis on constructing “sentences” that affect the game mechanically.

Examples

While this isn’t exactly a *common* feature of games, nor is it *rare*; examples include *Death Gate* (1994), *The Elder Scrolls: Morrowind* (2002), *Arx Fatalis* (2002), and *Tyranny* (2016).

Sometimes spells are just created in an ordinary menu interface, as in *Morrowind*, but often they are themed as combining runes, as in *Arx Fatalis*, or as LANGUAGE LEARNING. As mentioned in that chapter, *Rudra no Hihō* (1996) is a JRPG in which spellcasting is achieved by speaking “mantras” in an agglutinative language (as if the *Final Fantasy* spell-naming convention of *Fire* < *Fira* < *Firaga* were expanded into a richer language). As *Rudra no Hihō* was a console game, where text input is very slow, it was not possible for the player to type the spell each time; instead, players are given a “spellbook” into which they can write ten spells at a time.

Analysis

Unlike other forms of CRAFTING (where merely having the components is typically sufficient to execute a recipe), spellcrafting usually requires a more thorough understanding of the ways they components can combine. Even if the game has given you all the pieces you need, that doesn’t necessarily result in working spells any more than knowing a lot of words in a language allows you to speak it.

In *Magicka* (2011), for example, all but a few of the most powerful spell components are available to the player at the beginning of the game; all the player lacks is the understanding of how to use them, and the muscle memory to do so quickly under pressure. So



Arx Fatalis



Rudra no Hihō (Treasure of the Rudras) allows players to invent custom spells, like NAAQU here, so long as they follow the rules of the in-game spell-casting language.

progression happens not just through the game giving you the spell pieces, but by the player's growing understanding of the possibilities allowed by their combination.

Thus, like LANGUAGE LEARNING in games, spellcrafting can support an experience where the player feels like they're smarter than the game, that they're "further ahead" or "overpowered" due to their superior linguistic ability... even when it's fairly clear that the game was designed this way. The language in *Rudra no Hihō* is easy to decipher ahead of when you're "supposed" to, and just going around and talking to NPC gives you abundant clues just in case, but even so the player is made to feel like they're getting one over on the game by getting "early" access to powerful spells.

When a game includes a spellcrafting component, this is often central to the game, because balancing and working out the consequences of a complex combinatoric system is a lot of work! In *Death Gate*, spells are expressed as two-dimensional patterns made up of hexagonal "rune" tiles. As the player encounters spells in the game, they obtain the component runes that make it up, and can invent their own spells by re-combining runes in novel ways. According to designer Glen Dahlgren:

Each spell became its own verb, and boy did this make a mountain of work for me. One of the biggest challenges of an adventure game designer is making interesting (and hopefully funny) responses for all of the nonsensical actions the player can take. Now imagine adding a ton of new verbs you can apply to anything in the game, like Heat and Cold. What fun that was to think of hundreds of new responses spraying spells on every object in the game that likely few players would ever see.

That said, learning, building, and casting these spells was very satisfying, and magic was consistently one of the favorite features brought up by reviewers. It was a powerful tool for me as a designer, and I was happy about how it came together—especially when I got to use the runes themselves to make some interesting puzzles.

Challenge

Spellcrafting is almost a sort of programming language exercise, but I've never seen an AUTOMATION (p. 133) game or other indirect-control game take quite this route. I've mostly just seen this used for combat systems. Invent a magical automation game along the lines of *Factorio* or *SpaceChem* but in a magical/linguistic vein, where the components are driven by player-written spells in a magical language.



Death Gate

Glen Dahlgren. "Death Gate." In *The Mysterium: The Worlds of Glen Dahlgren*. <https://glendahlgren.home.blog/death-gate/>

Queries

The player constructs a query to attempt to find hidden information.

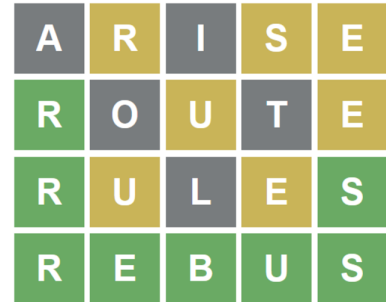
Examples

Many children's and parlor games are guessing games, dialogues in which a player "asks a question" and receives information from it, such as *20 Questions*, *I Spy*, *Hot and Cold*, *Botticelli*, and *Contact*.

Several traditional pen-and-paper games, like *Battleship* and *Bulls & Cows* (later commercialized as *Mastermind*, 1970) work this way as well: there is some secret that one player is trying to guess (e.g., the location of battleships or a secret code), and the other player responds (e.g., whether the square contained part of a ship or whether the guessed code had any correct symbols). The viral hit *Wordle* (2018) was an implementation of *Bulls & Cows* using words.

Black Box (1970s) is a *Battleship*-like boardgame inspired by the invention of the CAT scanner, in which player one secretly places imaginary particles on a board, and player two sends imaginary lasers from the sides of the boards. The lasers reflect off of particles, and player one informs player two where the laser left the board. From this, player two attempts to narrow down the location of the particles.

In videogames, *Her Story* (2015) is a notable game based on querying; you play as a detective searching through video interviews of a suspect, using a Google-like search interface that indexes her dialogue. As you learn more, you can craft better queries and find increasingly relevant interview segments.



Wordle



Black Box



Her Story

Riddles

The player is given a cryptic clue and has to guess what it refers to.

Examples

Riddles used to be fairly common in fantasy RPGs. *The Elder Scrolls: Arena* (1994) used riddles in certain dungeons, and *Betrayal at Krondor* (1993) used them to seal treasure chests.

Conquests of the Longbow: The Legend of Robin Hood (1991) even used them as copy protection; upon solving a particular set of riddles, you had to look up in the manual how to answer it using the monk's hand-sign codes.

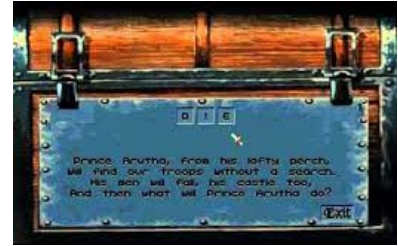
Riddles are somewhat less common these days, the only recent example that comes to mind is the riddle sidequest in *The Legend of Zelda: Breath of the Wild* (2017).

Analysis

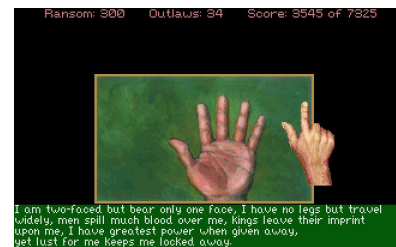
I think, like MAZES (p. 96), riddles in games have mostly died a just death. Riddles don't really work well if you're not on the same wavelength as the author; they often require a lot of cultural knowledge that a global audience doesn't have, and like WORDPLAY (p. 211) they're difficult to translate.

If you use riddles in your game, I strongly recommend either not having them on the critical path, or having them as an optional clue towards something that can be figured out another way. (E.g., the riddle might reveal the location of a treasure chest, but nothing prevents a dedicated player from simply scouring the area.)

Granted, players today have the internet and thus the possibility to simply look up solutions for any riddle (or puzzle in general), but even so I think riddles should be used with caution. Going to the internet takes the player out of the game, and also I feel like it is over-optimistic to design assuming the presence of a robust online fan-base.



Betrayal at Krondor



Conquests of the Longbow: The Legend of Robin Hood

Knights and knaves

The player is given an inconsistent set of statements, and must deduce which statement(s) must be false.

To be a true knights-and-knaves puzzle, a character must either *always* tell the truth or *always* lie, but here I'm stretching the definition somewhat.

Examples

Although earlier examples exist, this term was coined in Raymond Smullyan's 1978 puzzle collection, *What is the Name of this Book?*, p. 20.

There is a wide variety of puzzles about an island in which certain inhabitants called "knights" always tell the truth, and others called "knaves" always lie. It is assumed that every inhabitant of the island is either a knight or a knave...

According to this old problem, three of the inhabitants – A, B, and C – were standing together in a garden. A stranger passed by and asked A, "Are you a knight or a knave?" A answered, but rather indistinctly, so the stranger could not make out what he said. The stranger then asked B, "What did A say?" B replied, "A said that he is a knave." At this point the third man, C, said, "Don't believe B; he is lying!"

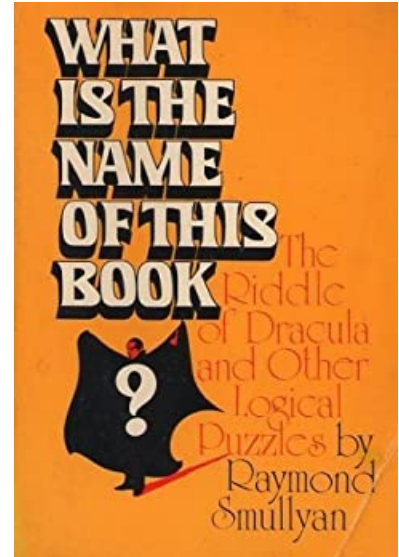
The question is, what are B and C?

The Elder Scrolls: Arena (1994) contains a Knights-and-Knaves puzzle:

If Cell 3 holds worthless brass, Cell 2 holds the gold key. If Cell 1 holds the gold key, Cell 3 holds worthless brass. If Cell 2 holds worthless brass, Cell 1 holds the gold key. Knowing this brave fool, and knowing that all that is said cannot be true, which cell contains the gold key?

Another knights-and-knaves puzzle occurs in one of *The Elder Scrolls: Morrowind's* riddle books (which also reappears in *Skyrim*), but has no gameplay consequences.

Knights-and-knaves-like challenges shade into WHODUNIT (p. 35) gameplay where you try to catch characters in lies, like the Dantooine mystery in *Star Wars: Knights of the Old Republic* (2003).



What is the Name of this Book



The Elder Scrolls: Arena



Star Wars: Knights of the Old Republic

Word puzzles and other wordplay

The player solves a puzzle based on the letter-by-letter structure of words and/or sentences, their sounds, puns, or other linguistic aspects beyond their literal meaning.

Examples

There is an enormous variety of wordplay-based puzzles and games. For an example in videogames, the classic puzzle collection *The Fool's Errand* (1987) contains many traditional and novel word puzzles.

Although it has little resemblance to actually learning a language, cryptograms are often themed as LANGUAGE LEARNING (p. 204), and examples from *Final Fantasy X* and *FEZ* are already covered there.

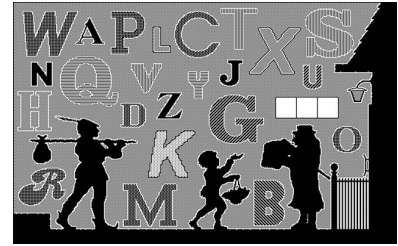
Counterfeit Monkey (2012) is a text adventure in which the player is given a magical tool that transforms objects into other objects by inserting, removing, changing, or rearranging the letters in their names (e.g. “money” → “monkey”). *Typoman* (2015) is a puzzle-platformer with a similar conceit.

Analysis

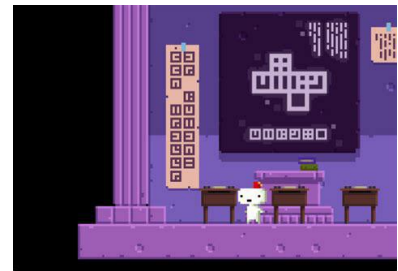
One thing to be careful of is that you can no longer truly translate/localize games with a lot of wordplay mechanics; you can only make a similar game in a different language. Changing the words can potentially change the structure and balance of the game, and some set-piece puzzles may no longer work at all.

For example, in *Typoman*, where objects in the world are made of their letters, there's a puzzle set in a rainy scene, where you drain a pool by adding a D to RAIN to DRAIN a pool of water. This puzzle – and the assets created to support it – only work in English.

An infamous puzzle in *Monkey Island 2* (1991) required the user to turn a stuck bolt, and the inventory item needed was... a monkey. Get it, “monkey wrench”? Unfortunately *MI2* was released in translation, leading to a lot of baffled Germans. Even British English speakers had trouble, preferring to call that tool a “spanner”.



The Fool's Errand



Fez



Typoman



Monkey Island 2: LeChuck's Revenge

Part XVI

Hidden information and special rules

Hidden rules

Some important rules of the game are left unexplained, and the player must figure them out, typically by attempting various actions and analyzing their rewards and punishments.

Examples

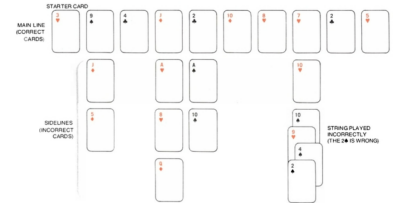
The classic card game of hidden rules is *Mao*. A particular game of *Mao* will have a variety of bizarre house rules; players familiar with those rules are not permitted to teach new players, and new players must attempt to deduce the rules by trial and error.

It is fairly common for videogames to “hide” at least some of their rules. Few RPGs, after all, tell you their exact hit/damage formula, and some, like *Pokémon*, hide entire stats. Akitoshi Kawazu’s RPGs like *Makai Toushi SaGa* (1989), released as *Final Fantasy Legend* in the West, are notable for deliberately explaining almost nothing about important game systems.

CRAFTING (p. 163) and ANIMAL HUSBANDRY (p. 63) systems often have outputs whose natures and stats depend on rules that are never fully explained, and engaging with these systems can be an exercise in deduction. The *Legend of Mana* (1999) crafting system was notorious for this, appearing to add and remove effects from equipment almost at random. Since crafting is entirely optional in this game, however, its inscrutability is forgivable. Players who enjoy black-box puzzles (cf. QUERIES, p. 208) can work out the hidden systems in detail; players who do not can avoid it or treat it as a sort of GACHA (p. 250) machine that occasionally spits out amazing equipment for unclear reasons.

Analysis

Hidden rule games are often compared with the scientific method: the player is making hypotheses, and can only test them through making plays and observing the results. *Eleusis*, a *Mao*-like query game popularized in Martin Gardner’s column in *Scientific American*



Eleusis, as illustrated in *Scientific American* 237:4, Oct. 1977



Final Fantasy Legend makes no attempt to explain how eating different kinds of meat transforms monsters in your party into other monsters.



Even the GARDENING (p. 64) system in *Legend of Mana* is arcane; which seeds result in which plants is strangely complex. And this is the game’s most straightforward crafting system.

What producer even greenlit so many crazy systems in one game? Let’s see, producer... oh. It was Akitoshi Kawazu.

Martin Gardner. 1959. “Mathematical Games”, in *Scientific American* Vol. 201, Issue 6, June 1959.

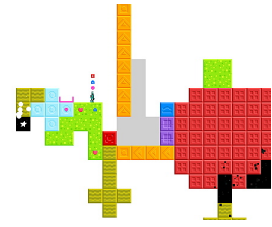
(June 1959 and Oct. 1977), has been used (in simplified form) to teach the scientific method to elementary-school students.

Hidden rules need to be used carefully; games like *Mao* have a certain cruelty to them, as the knowing players punish the unknowing for infractions that they couldn't possibly know at the outset. You the designer are in the role of the rule-knower, but keep in mind that in a videogame there's no social pressure keeping your rule-learner playing your game; after reaching their threshold of bewildering punishment, they'll just turn off the game.

Remember that your player doesn't know you personally. They won't necessarily think "My brilliant friend must have some great surprise in store for me." They may just think "This game is broken and stupid" and move on. There are more buggy, ill-conceived, and poorly-telegraphed games in the world than there are hidden masterpieces.

It's thus important when building a hidden-rule game that there be *some* manner of intermediate discovery, some part of the rules that are locally valid and can serve as a metaphorical platform on which the player can rest. You probably don't want the rules to be only discoverable in their entirety in a single flash of insight after hours of bewilderment.

One interesting aspect of hidden-rule videogames is that, once you understand the rules and structure of the game, it often becomes quite short and straightforward. In *The Witness* (2016), for example, there is actually little keeping you from the endgame except that the challenges in your way will be incomprehensible to the new player, and some will not even be recognizable as challenges. The game is primarily gated by knowledge, and the main reason to play through the game's many areas is to learn the visual "language" in which the puzzles are couched (cf. LANGUAGE LEARNING, p. 204). That is an extremely interesting structure and one that I think will receive more imitation in the future.



Playing *Starseed Pilgrim* (2013) is largely a process of figuring out the nature and goal of the game. Its simple presentation and lack of explanation would probably have caused most players to give it a miss, or play it briefly and assume there was little to it, but for a glowing endorsement by Jonathan Blow (creator of *Braid* and *The Witness*).

You have to be cautious about hiding your light under a bushel; most games don't get that initial opportunity to be lauded by a famous designer.

Hidden goals and roles

Different players (or NPCs) have goals or roles of which other players are unaware. Typically they know what goals/roles are available, but they do not know exactly which is assigned to which character.

Examples

While it's not the first example ever, hidden roles were popularized largely by the parlor game *Mafia*, also called *Werewolves*, which spread worldwide by word of mouth in the 1990s. In it, players role-play inhabitants of a small village; most villagers are innocents, but a few are traitors. The innocent villagers do not know who the traitors are, and seek to identify and execute them; the traitors pretend to be innocents and seek to lead suspicion away from themselves and their conspirators.

Influenced by *Mafia/Werewolves*, many boardgames and cardgames introduced hidden goals or roles, like *Bang!* and *Citadels*.

Shadows over Camelot (2005) deserves special mention here, because it introduced a now-common kind of semi-cooperative game. Like most cooperative games, the players are engaging in grand tasks together. Each time a quest succeeds, the players place a white sword on the table; when it fails, a black sword; the game is won if white swords outnumber black swords at the end. However, one of the players may secretly be a traitor, and they win if the black swords outnumber the white swords. The traitor must play in such a way that they sabotage the quests, but not in such an obvious way that their identity is discovered.

Another twist on this is games where one player is missing crucial information and tries to hide this fact. *A Fake Artist Goes to New York* (2011) is a cooperative drawing game where one person doesn't know what the team is drawing, but has to fake it. *Spyfall* (2014) is similar; everyone is role-playing being in a particular location, and the "spy" character doesn't know what the location is.



A sample of role cards from *Bang!*



Shadows over Camelot

Fog of war

The game map is blank or covered in fog until the player has explored nearby.

Examples

This is a very common game mechanic from *Civilization* to *Super Metroid*. But really, I added this chapter just to talk about the clever fog-of-war system in the *Merchant Prince* series of Renaissance-era TRADING (p. 102) games.

Rather than cover the world in uniform gray fog until you visit it, the player sees the world map *as understood by Renaissance-era Europeans*. Many of the basic landmasses of the Eastern Hemisphere are known to them, but they don't know exactly what shapes they have or where they are. Shapes and locations are increasingly distorted the further you go from the Mediterranean.

This isn't just historically realistic (it's not like European explorers were just heading out at random); it also helps spur the player to choose a direction in the absence of other motivations. Exploring a featureless gray fog sets up a Buridan's Ass problem (p. 95): why should I go west here rather than northwest or south or...? Even a little nudge can give the player a way out of this dilemma.

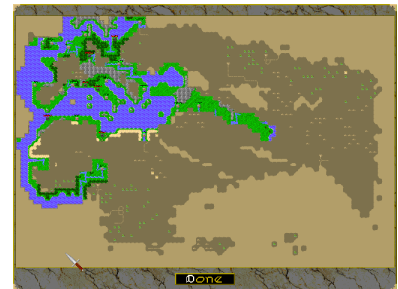
Analysis

Another thing fog-of-war brings is that it can keep the player from being overwhelmed by too much information or too many possibilities. As Hentzau points out in his review of *A Monster's Expedition*:

If you showed me a list of nine-hundred log-pushing puzzles and asked me to work through them in sequence I'd probably tell you to sod off, but the pseudo-open world structure of *A Monster's Expedition* – where the entire map is covered in clouds with only discovered islands visible so that you can't see the true scale of the game, and where you can only uncover new islands by solving current ones – keeps your attention focused onto the next two or three islands that you need to solve.

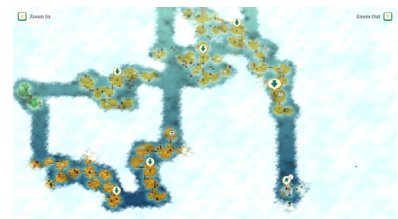


Now that I think of it, why don't the colonists of *Sid Meier's Alpha Centauri* know even the basic shape of their planet's continents? They were just in orbit a few days ago!



Machiavelli the Prince (1995), second game in the *Merchant Prince* series

Hentzau, 2020, *Thoughts: A Monster's Expedition* at scientificgamer.com



A Monster's Expedition

Hidden passages

The game obscures a passage, **SHORTCUT** (p. 104), or treasure cave.

Examples

An classic example of this is the bombable walls in *Zelda* games, which are reliably (at least in later games) marked by a distinctive pattern of cracks. If you bomb these, an entrance is revealed and treasures are usually contained within.

Similarly to **HIDDEN OBJECTS** (p. 155), secret passages like this are rarely on the critical path, at least in modern games. One exception to this is *Fairune* (2013), a minimalist retro-styled *Zelda*-like RPG, where important passages are often indicated only by slight tile variations. It's therefore quite easy to get stuck in this game, although I find it forgivable, because there isn't much else going on in the game and it's not a very challenging game otherwise.

Analysis

For an example of **CLAIRVOYANCE** (p. 218) applied to hidden passages, an important non-combat passive ability in *Final Fantasy V* is the Thief's ability to see hidden passages.

The game world of *FFV* is full of **HIDDEN PASSAGES** (p. 217), including an entire town based around them. Unlike bombable walls in *Zelda* games, hidden passages are typically unmarked and invisible until you're right up against them.

However, the Thief class has a special ability that makes them visible to the player. Thieves are probably in part *why* the game has so many hidden passages; their frequency gives the player an additional reason to train up a Thief party member, which is otherwise a somewhat weak early-game class.



A bombable wall in *The Legend of Zelda: A Link to the Past*.



In *Fairune*, tiles that subtly differ from the typical tiles are traversible; this one tree, for example, lacks a trunk.



The horizontal passage on the left side of this *FFV* screenshot is usually invisible, but having a Thief in your party reveals its presence.

Clairvoyance

The player can see hidden information, reveal hidden things, or know the results of future events before they occur.

Examples

Being able to see hidden traps, or seeing inside treasure chests before opening them, are common special abilities in RPGs (especially rogue-likes), letting the player better judge risks.

Games with hidden rules, goals, or roles (p. 213-215) often have a special ability or reward that reveals information, like the Fortune Teller role in *Mafia/Werewolves* letting a player discover another player's true identity once per night, or the scholar character in *Touhou Mystia's Izakaya* revealing the food PREFERENCES (p. 54) of other characters.

Clairvoyance can also be a hedge against randomness. In *Alex Kidd in Miracle World* (1986), bosses first play rock-paper-scissors with you before the actual boss fight. If you lose, you lose a life, which is a dire penalty in a game where lives are scarce. However, you can also find a "telepathy ball" that reveals the opponent's next move.

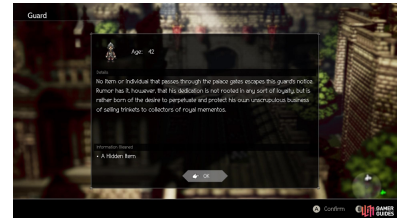
Analysis

FOG OF WAR (p. 216), HIDDEN PASSAGES (p. 217), and LIGHT (p. 219) give you the designer another kind of reward or ability to dole out to the player: the ability to see farther or reveal hidden parts of the map.

Designers of non-violent games sometimes have trouble coming up with a diverse list of rewards/upgrades to give to the player at a steady drip. But most games have some kind of hidden information, so the revelation of this information can be a reward or special ability in itself. You can even put *more* hidden information into a game just for the purpose of making these rewards more valuable.



"Item sight" in *Cadence of Hyrule* (2019) reveals the contents of treasure chests in advance, so that the player can judge whether going for it is worth the time and risk.



Cyrus in *Octopath Traveler* (2018) has the special ability "Scrutinize" that reveals NPCs' secrets.



Alex Kidd in Miracle World

Light and darkness

In-game light sources and shadows reveal or obscure level elements, or the player's vision is otherwise obscured.

Examples

A real-world use of light and shadow can be seen in the children's game *Shadows in the Forest*. It's an asymmetrical game in which one player is a "Seeker" represented by a tiny LED lantern, and everyone else is "Shadowlings" who attempt to hide from the seeker in the shadows cast by stand-up cardboard trees.

Many games like *Zelda II* have lantern items that light up dark levels. However, at least in games with fixed level layouts, obtaining lanterns is often not absolutely necessary; if you have memorized the level, you can potentially still run through it. Some games use intermittent darkness to add MEMORIZATION (p. 220) gameplay: briefly illuminating a scene and then turning off the lights, or having it periodically illuminated by a lightning flash.

One of the iconic STATUS EFFECTS (p. 146) in the Mario Kart games, first seen in *Mario Kart DS* (2005), sprays Blooper ink over the player's windshield and makes it hard to see the road ahead.

Closure (2012) is a puzzle platformer where only illuminated objects exist; if you walk on a platform in shadow you simply fall through it. Getting through the levels requires manipulating the light sources. *Lightmatter* (2020) uses a similar mechanic in a 3D world.

Ori and the Will o' the Wisps (2020) has an area in which the darkness will kill you, and lanterns last for only a limited time, meaning that you have to rush between oases of illumination to survive.

Analysis

My main warning about the use of darkness as a challenge is to keep accessibility in mind. Especially in pastel-hued "cozy" games, dark areas can become inaccessibly low-contrast.



Shadows in the Forest



Mario Kart DS



Closure



Lightmatter

Memorization

The player is tasked with repeating something, or doing some other kind of task, using information that they have only seen in the past, and cannot access in the present.

Examples

Most games stress the player's short term memory at least somewhat – “what is my current goal?”, “what button does what?”, etc. – but some games rely more heavily on memorizing arbitrary sequences and patterns.

An early arcade game based on memorization is *Touch Me* (1974), a game with four buttons with associated lights and sounds. The game plays a sequence of lights and sounds, and the player must replicate it. This same idea, given a visual and aural upgrade and released as a handheld game, became a smash hit toy as *Simon* (1978).

Memory challenges can be used to make a simpler puzzle challenge more difficult, as in the children's game *Memory*, in which a memory component supplies the difficulty to an otherwise-trivial MATCHING PUZZLE (p. 196).

One of the minigames in *Ichidant-R* (1994) is a crane game in which you must extract some gold vases from a stack of worthless ones. However, you only see which vases are golden for a moment before the lights go out, and must do the actual puzzle only in silhouette.

Carto (2020) offers TILING PUZZLES (p. 190) in which the player is rearranging *Carcassonne*-like tiles and also controlling the protagonist living in that world. While many of its puzzles are straightforward, they are complicated during the two-layer “volcano” level: you can only see one layer at a time, and can only move pieces while viewing the top layer. You thus have to remember the orientation of pieces on the lower layer while manipulating the top layer.



Simon



The *Memory* minigame in *Super Mario Bros. 3*.



Ichidant-R

Analysis

Memory challenges aren't too hard to add to games, but should be used with caution. While people do have some ability to "level-up" their memorization skills with respect to your game (e.g. by learning to remember sequences of events as "chunks"), there's simply a cognitive limit beyond which they player can't "git gud" anymore, and that limit will differ for different people. The ability to wholly devote one's memory to a game depends on circumstance, too. Is the player at home alone, or are they constantly being interrupted? Are they playing in quick bursts on public transit?

The table-waiting minigame in the *Tales* series (e.g. *Tales of Symphonia*, 2003) is a case in point. In these games, you can work as a waiter and have to memorize customers' orders. It's a highly divisive minigame among the player base, particularly as later installments increase the difficulty. It's hard enough memorizing orders, but at the orders get ridiculous (31 mapo tofus for table 2!) and difficult customers cancel orders a dozen times in a row, it grows to the point where for many players the stress of the minigame outweighed the fun to be had with it.

It's also worth noting that memory plays a crucial role in Metroidvania-style MAP EXPANSION (p. 228). Just like in the card game *Memory*, the player is finding locks (in the abstract sense) they cannot yet "match" with keys, and ideally files their locations away in memory for quick retrieval when the corresponding ability upgrade is found. As *Game Maker's Toolkit* on YouTube pointed out, a good Metroidvania assists the player using distinctive level design to aid their memorization. For example, an important "lock" might be accompanied by a grotesque statue that serves no immediate gameplay function, but serves to make the room more memorable.



Tales of Symphonia

Unidentified items

The player obtains an item, but doesn't know what it is until they either use it, take it somewhere for identification, or use some sort of identifying ability or item (e.g. a "scroll of identification").

Examples

This pattern originates in *Rogue* (1980); potions are identified by color (e.g. "beige potion") and the association between the color of the potion and its effect (e.g. invisibility) is randomized each game. The player has to either use potions at random to figure out their effects, or use a scroll of identification.

For a unique graphical theming of the same idea, the action-roguelike *ToeJam and Earl* wraps all items up like birthday presents. The same item will always be wrapped in the same way (i.e. same size, shape, and color), but this is randomized between runs.

In the *Diablo* series and similar games, magic and unique equipment all have unidentified effects, which you cannot see until you use a scroll of identification, an identification ability, or pay someone to identify them for you. In the first game, you can equip unknown items, but just equipping the item does not activate the unknown effect; only the base stats are applied when equipping unidentified items. (Equipping equipment without identifying it is sometimes a better choice, since items in *Diablo* can have negative effects.)

Analysis

Even though *Rogue* has proven to be one of the most influential games of all time, the unidentified item system has largely not survived in its modern descendants. (At least, not in many of the popular rogue-likes and "rogue-lites" on Steam; I assume that plenty of more traditional ASCII rogue-likes still feature it.) Really only the *Diablo* lineage of action RPGs still uses it, and that in a somewhat different form.



In this playthrough of *ToeJam and Earl* (1991), the player has already identified that the purple present with the star represents Bonus Hitops, but the flat green present remains unidentified.



Diablo II

I feel its disappearance is largely just; it's a simple puzzle, but one that you primarily solve by trial-and-error, and getting it wrong can kill you. As Jeremiah Reed (designer of the rogue-like *Golden Krone Hotel*), put it:

One of the hardest lessons to learn about roguelikes is that you must use your damn items. It's very common for new players to die with full inventories. Part of this can be explained by item identification. When each unidentified item could be one of 20 things and half of those things are bad, the risk of using items appears too high.

The result is that new players typically disengage from the potion system altogether. Advanced players who know a ton of item identification hacks will engage with it, but many of those hacks trivialize it; even then it doesn't exactly make a *good* puzzle.

In *Diablo* and its descendants like *Torchlight*, however, the system still serves an interesting purpose. One thing about *loot* as a reward is that it doesn't have much of an anticipation stage (cf. the Anticipation/Honeymoon/Neutral cycle discussed in *LEVELING*, p. 232).

I can anticipate getting loot *in general*, but it's a different anticipation than seeing that my EXP bar is almost filled up and I'm getting level-based rewards soon. Even if I don't know what rewards the next level brings, level-based rewards engender a more specific anticipation; I know I'm getting *something*. Unidentified loot is more like levels in this respect; like a wrapped present under the Christmas tree, it heightens my anticipation for the time in the near future when I'll open it.

Having a character "in town" who identifies items also helps reinforce a medium-term game loop between action and rest (cf. *RESTING*, p. 76). You adventure, you get unidentified loot, you take it back to town to be identified. Having the loot unidentified sets up anticipation for getting back to town, and once the loot is identified, the "honeymoon" effect motivates you to return to adventuring to try out your new toys. So if you the designer are having trouble motivating the player to take rests, or having trouble motivating them to stop resting and get back to adventuring, item identification might help grease the wheels in both directions.

Religion

Aside from the ordinary rules of the game, the player also has to follow additional rules out of commitment to a god, religion, or legal regime. The player receives rewards for following the laws and/or punishment for breaking them.

Examples

Many tabletop RPGs and roguelikes have alignment, conduct, or religion systems. The Monk class in *Nethack* (1987), for example, suffers a small alignment penalty if they eat non-vegetarian food.

Some games implement religion as a sort of RELATIONSHIP BUILDING (p. 52) – you choose a god, and build “piety” by acting according to their unique preferences. For example, if you worship the Earthmother in *Desktop Dungeons* (2010), using her signature spell nets you +10 piety, but killing a plant nets you -15 piety. Depending on the game, piety can be spent like a resource, or reaching piety thresholds can grant you new abilities.

The action RPG *Bastion* invites the player to pray to various idols, each of which makes the game more difficult in some way, but also provide multipliers for in-game rewards. Unlike many games, these gods are not jealous gods; you can invoke multiple gods at once for an extra-difficult but extra-rewarding game.

Sometimes additional gameplay restrictions are themed as laws rather than something supernatural. When these are under the control of the player (e.g. the player is inventing or influencing these laws) I cover them under RULE MODIFICATION (p. 226), but in some games like *Final Fantasy Tactics Advance* (2003), the laws are out of the player’s control and I decided to cover them here instead. The world of *FFTA* follows a cycle of arbitrary laws; each day a particular action, weapon, element, etc. will be forbidden in normal battles, and if the player breaks these laws repeatedly a referee will enact various punishments. Punishments include confiscating equipment, releasing random monsters into the battle, and removing the offender from play and sending them to jail.



The gods of *Desktop Dungeons* give out rewards or punishments based on their judgments of your actions.



The idols in *Bastion* make the game harder in exchange for in-game rewards like EXP multipliers.

Analysis

In games that the player is meant to play again and again, religion is an easy addition to motivate them to try different play-styles. For example, if players are mostly breezing through the game using magic and not using their physical abilities, one way to encourage them to shake it up and try playing physically is to add a god who prefers physical feats and disdains magic.

However, in some games the proscription against using certain game systems isn't optional or temporary, but is baked into the mechanics of the game, where engaging with particular game systems *at all* is matched by eventual supernatural punishments. I don't like this pattern very much; it feels like tempting the player with fun things to do and then punishing them when they actually do.

For example, in *Magical Vacation* (2001), the main health "potions" are gummy frogs that jump around the map; a player in need of health can chase and eat them. However, certain elemental spirits frown on frog-catching, and players who have caught too many will be unable to completely level up spells of that element. It's a strange feeling to run through an otherwise low-stress game avoiding an entire game system; encountering a field of frogs because a source of anxiety rather than fun. Part of this anxiety is that frogs are quite common and just that picking up frogs is easy to do; press the main interaction button at the wrong moment and you've caught a frog; sorry, elemental spirits! If you're going to punish the player for an action, make sure that action is hard to do by accident.

This is also a common complaint about *Dishonored*, that it gives the player a rich toolbox for violent gameplay, but if you actually engage with this part of the game, you're magically punished for it: more enemies spawn and you're locked out of the good ending. (It sounds strange for me to argue *against* the game's non-violence bias in a book like this, but I just don't like the heavy-handed way *Dishonored* handles it.)

Rule modification

The player can make changes to the rules of the game itself.

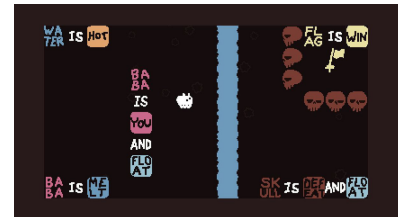
Examples

The cardgame *Fluxx* is based around this; the game initially starts with few rules (not even a win condition), but many cards in the players' hand add rules to the game, with the result of a chaotic and unpredictable game.

Baba is You (2019), widely considered one of the best puzzle games of its decade, is a BOX PUSHING (p. 193) game except that the very rules of the game are encoded into blocks that you can push, making manipulation of the game's rules a part of every puzzle.

Some strategy games where you play a ruler involve making laws and treaties that restrict your own ability to engage in gameplay systems (e.g. non-aggression treaties). *Crusader Kings II* is a particularly interesting example. Unlike many grand strategy games, the ruler is not a godlike figure existing for centuries; they are a mortal human in the world, and when they die the player controls their heir instead. Inheritance thus becomes a central system of the game, since it determines exactly what lands and titles your heir inherits. But since the inheritance laws are themselves something that can be legislated, the player's in-game actions have huge effects on the rules of the game itself.

Disgaea offers a light-hearted take on a similar idea. The player, a wannabe demon overlord, can summon a demonic parliament and propose bills – anything from “Enemies are less/more difficult” to “More expensive items are available in shops”. This latter is particularly interesting, since there is only one set of shops in the game, the ones in the hub world. Whereas in most RPGs, equipment progression happens naturally as you journey and find new towns with new shops; in this game, that aspect is replaced by interacting with parliament.



Baba is You

Part XVII

Progression and upgrades

Map expansion

The player gains the ability to go to additional places on the map, whether that's by solving a LOCK-AND-KEY PUZZLE (p. 183), getting a new ability, a new vehicle, fixing a broken bridge, etc.

Examples

This is the central progress pattern of “Metroidvanias”, a portmanteau of *Metroid* and *Castlevania*. In these games, the player is initially prevented from exploring the entire map because of barriers that they cannot overcome; as they play, they find permanent upgrades that allow them to breach these barriers and explore progressively larger portions of the map.

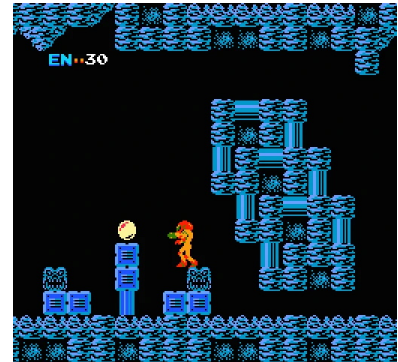
This also occurs in many other genres. A very similar ability-based progression scheme is seen in the RPG *Mario & Luigi Superstar Saga*. *Final Fantasy* games tend to gate progression by VEHICLES (p. 113), with the player starting out on foot but eventually getting the series' iconic airship.

Even things like the “broken bridge” gates in *Chrono Trigger* (1995) or *Grand Theft Auto III* (2001) would count as this, although in these cases it's just a plot-sensitive game-internal flag that opens up the new area.

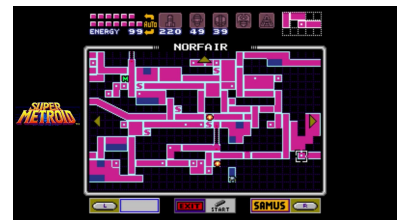
Many games boil down map expansion to a single number. “Collectathon” games like *Super Mario 64* (1996) and *FEZ* (2012) gate the map according to how many of the collectables (e.g. stars) you've found; doors are labeled with some number of stars and are locked until you've found that many. *A Short Hike* (2019) is set primarily on a single mountain, and higher regions are inaccessible until you upgrade your CLIMBING (p. 110) stamina.

Analysis

Map expansion an extremely powerful pattern for an exploration-focused game, because it keeps the player to a small part of the map



Samus in *Metroid* cannot fit through 1-tile passes like the one to her right until she picks up the “morph ball” upgrade to her left.



A small section of the complex map of *Super Metroid*. Its full complexity is only revealed slowly (cf. FOG OF WAR, p. 216), as Samus finds the upgraded suit parts that allow her to bypass various kinds of barriers.

while they're still learning, and expands the world as their skill and confidence grows.

When designing Metroidvania-style map progression, it's important to keep in mind that this pattern naturally comes with a lot of backtracking, as the player explores and re-explores the world in search of places to use their new abilities. To avoid player tedium, it's good to take care in designing the world so that re-traversals are easier, faster, or at least *different*, especially by utilizing the player's new abilities.

I mention in *SHORTCUTS* (p. 217) the importance, particularly in platforming-oriented Metroidvanias, of being diligent in opening up shortcuts after a player's first time through a platforming gauntlet.

Puzzle-oriented Metroidvanias come with a different danger: the player spending hours on a puzzle with many moving parts (e.g. *BOX PUSHING*, p. 193), never realizing that they simply haven't found the tool that makes it solvable. In this case, it's good to use the level design to ensure that complex puzzles are gated by smaller, easier obstacles that require all the same tools – as simple as a *LOCK-AND-KEY* (p. 183) puzzle. The player may still knock their head against the small puzzle, but hopefully for only a moment or two before realizing it's unsolvable with their current arsenal.



When I get an ability that completely changes the way I traverse levels – like the ability to launch off projectiles in *Ori and the Blind Forest* – backtracking is actually quite enjoyable, because it forces me to see familiar objects and places in a new light (cf. *VARIANT FORMS*, p. 144).

Expanding hubs

As the player progresses through the game, their hub world expands with new buildings and NPCs.

Analysis

In adventure/RPG games, this is particularly associated with the *Suikoden* series. In each of these games, there is a hub location (usually a castle), and 108 recruitable characters. As you recruit more characters, they show up in your castle and slowly turn it from an empty building to a bustling small town.

In *Suikoden* this expansion happens outside of your control; when you recruit the blacksmith, a particular room is turned into a smithy automatically. In other games, hub expansion shades into a sort of MAP-BUILDING (p. 81) gameplay. *Ni no Kuni 2* (2018), for example, has you building the buildings and staffing them with recruitable characters; it's halfway between *Suikoden* and *The Settlers* (1993).

Hub expansion also provides some of the META-PROGRESSION (p. 259) in rogue-like games like *Rogue Legacy* (2013), *Loop Hero* (2021) and *Cult of the Lamb* (2022). Even if each discrete playthrough starts from scratch in a randomized world, the rewards from it can go towards upgrading a permanent hub-world.

Analysis

Especially in games where the town is rather small and concentrates on individual people, this provides a way to give the player a feeling that they're shepherding a community of people, and by their actions improving people's lives. It also provides a way to keep favorite NPCs relevant and active throughout the course of the game, rather than having the player meet them once and move on.

For this pattern to work, the player needs to be frequently returning to the same place. This may not make sense in a picaresque journey where the protagonist is forever traveling onward. On the other hand, if the game structure is such that the protagonist is fre-



Suikoden II



Ni no Kuni 2: Revenant Kingdom



Ni no Kuni 2 put a lot of effort into making distinct characters with strong Ghibli-like visual designs. Rather than limit the appearance of these characters to single plot arcs in particular towns, having the player invite them to the main settlement lets the game get more use out of them.

quently returning to a central hub, this pattern can shine. Rather than have the hub be a static asset that the player will eventually get bored with, have it grow along with the player into a vibrant town.

The growth of the town can mirror the mechanical progression of the player; a new system might be introduced not when the player reaches level 10, but when a particular NPC is recruited and builds a relevant shop in the hub town. *Suikoden* manages weapon progression this way. In a typical RPG, the player accesses progressively-more-powerful equipment as they travel; in a fortunate coincidence each town sells slightly better equipment than the last. In *Suikoden*, on the other hand, you happen to meet three increasingly-skilled blacksmith NPCs, who, upon recruitment, return to your castle and increase the level to which your smithy can upgrade weapons.

Leveling up

The player gain experience points from achievements and actions, once they reach particular experience thresholds (“levels”), the receive rewards like increased stats or new abilities.

Examples

RPG-style leveling has become part of the DNA of nearly every game genre.

Sometimes the rewards for leveling are fixed (e.g., you get particular stat upgrades and new abilities upon level up depending on a character’s “class”); many early RPGs worked like this, and games where leveling is only a minor aspect of progress.

It’s becoming more common, however, that leveling up is associated with a choice of rewards – it’s more of a treat, and focuses the player’s attention on the reward. In *Fallout 3* you get one point per level that you can spend on a new ability. In *XCOM: Enemy Unknown* (2012), your soldiers, on each level up, get the choice between two special abilities; but unlike in *Fallout*, you cannot go back and “spend” this on a previous ability; unchosen abilities are locked away for good.

Analysis

A frequent question on design forums is what leveling is *for* and what we can replace it with instead. Can it be dispensed with? Could we replace leveling altogether with the player periodically finding stat-boosting collectables, like the permanent stat-boosting “tabs” of *Chrono Trigger*? Well, of course we could, but keep in mind that two of the crucial functions of leveling would be lost.

I’ll illustrate the following with combat, but it applies to any challenge. Leveling obviously works with a wide variety of challenges or it wouldn’t have infected almost every genre of game.

One, leveling provides an implicit difficulty dial that adjusts naturally as the player plays. Bold, reckless, or expert players who speed



Dragon Quest (1986)

towards the local boss will lose out on a lot of experience points that they would have gained if they had explored more thoroughly, and thus engage the boss under-leveled and have a harder fight on their hands. Meanwhile, more conservative, timid, or inexperienced players can explore thoroughly and “grind out levels” in order to make the upcoming fight easier.

This dial is not infinite – level thresholds tend to increase triangularly or exponentially, so fighting the same enemies forever has diminishing returns. However, there’s still a wider dial than (say) the power-up collectables suggested above, which are typically finite.

RPG players have been complaining about “grinding” for 40 years now, but “grinding” is just what we call it when this system is unbalanced and bosses are too hard for the average player. If this system really didn’t work, it would have died out... but instead it’s survived and spread far beyond RPGs, because it allows games to cater to a wider variety of player temperaments and skill levels.

The other thing about leveling is that a level-up is a reward that you can *anticipate*. Finding a “tab” in *Chrono Trigger* or new suit in *Super Metroid* is a great reward, but I can’t anticipate it in advance.

Backing up, when a game has levels, big purchases, research, etc. towards which the player is working – a progress bar filling up, however we want to theme that – we can split the player’s time into three periods:

- **Anticipation:** the player knows there’s some sort of upgrade coming soon. They’re unlikely to turn off the game right now, because they know they’re about to be rewarded.
- **Honeymoon:** the player has received their upgrade. They’re also unlikely to turn off the game right now, because they’re having fun trying out the new possibilities it offers.
- **Neutral:** The fun has worn off from getting the upgrade, but the next upgrade is too far off to really be anticipated. This is the period where it’s more likely that the player turns off the game.

Of course, this is just one of the things keeping the player playing; hopefully the game is enough fun on its own. But through decades of experience, designers have learned that adding in leveling (or a similar big progress bar) adds another layer of compulsion to a game.

This is seen most notably in the *Civilization* games. The Civilization games are *full* of progress bars; not only is there your research progress, but every city has a progress bar towards its next building or unit. The player is constantly a turn or two away from something they want, or a turn or two after getting something new; they are only rarely in “neutral”. This underlies the “One more turn” phenomenon that keeps the player awake until 4 in the morning.

Leveling by use

Characters become more proficient in actions that they perform repeatedly.

Examples

Although it has precedents in table-top RPGs, leveling-by-use in videogames is particularly associated with *Dungeon Master* (1987), *Final Fantasy II* (1988), and the *Elder Scrolls* series. Physical fighters improve their physical stats and abilities (sometimes in weapon-specific ways), magic users improve magical stats and abilities; in some games, players increase health and defense by being attacked.

Analysis

RPG-style LEVELING (p. 232) is sometimes derided as unrealistic. If I'm a magic user who defeats a monster by poking it repeatedly with a spear, how does this "experience" contribute to me learning new spells? So some games implement a more psychologically/physiologically realistic progression system, in which practice with spears only increases your proficiency with spears.

However, this pattern should be used with caution. Especially because this it tends to reward *actions* rather than *results*, it can incentivize the player to repeatedly engage in useless actions as a means of leveling up. It may be *realistic* that doing the same thing a thousand times improves your skill, but is it *fun*? There's a reason that movies express training sequences as montages.

Simpson (1999) explored the unintended consequences of this pattern in the economy of *Ultima Online*:

As explained earlier, skills improve through use. For a manufacturing profession such as tailoring or smithing, players are motivated to improve skills because it will allow them to make better items for themselves on-demand in the future. Thus players, in the process of training, produce huge numbers of item regardless of whether there is a market for them. This, combined with the effects of macroing and the



Dungeon Master



Final Fantasy II is notorious for players repeatedly attacking their own party members to build up their HP.

Zachary Booth Simpson. 1999. "The In-game Economics of Ultima Online." Presented at the *Computer Game Developer's Conference*, San Jose, CA, Mar. 2000

fact that some players create items for the sheer fun of it, conspire to shift the supply curve to the right. This results in overproduction and deflated prices of many basic goods.

It's also important to understand how leveling-by-use can "type-cast" players – if you have sunk dozens of hours into improving your archery, you're incentivized to remain an archer because switching to a spear carries with it a substantial power setback. Again, this is realistic, but is it fun? Game designers generally want their players to try out a wide variety of their game's systems during play, and this tends to pull in the other direction.

By bringing up these problems, though, I don't mean to say that leveling-by-use is always a bad idea. If the game is short and meant to be played many times, or if the player upgrades many characters like in *XCOM: Enemy Unknown* (2012), what harm is there that a particular character becomes typecast?

Or, you could simply limit the amount of times the player can level up their ability through repeated use, until some result is achieved or some event happens. I can't think of a game that does this exactly, but it's similar to the way that *Chrono Cross* makes level grinding impossible. Characters randomly receive small stat upgrades after encounters, but there are actually relatively few of these available at any given point, and after a handful of battles the characters simply stop improving. However, after major events like boss battles, the "growth level" is incremented for all characters in the game. This grants all characters (even ones you're not using) stat upgrades, but also "resets" the ability to gain small stat upgrades after battles. This system means that the player still feels rewarded for engaging in *some* encounters, and can somewhat improve their most-used characters over their peers, but grinding a character until they're over-powered is simply impossible.



Chrono Cross

Research

The player chooses an upgrade in advance, to which they'll devote experience/research/building points until the upgrade is achieved.

Examples

The first game I'm aware of that had "research" in this sense was *The Sumerian Game* (1964), the first game where the player shepherds a nation through the ages by making decisions. In it, you could allocate workers towards the development of certain crafts.

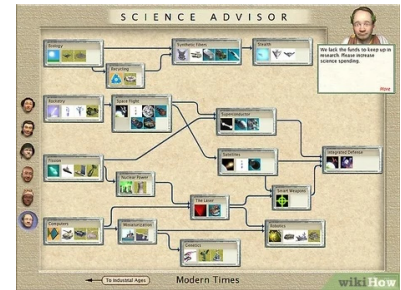
The prototypical example, however, is the research system in the *Civilization* series. The player's civilization generates "research points" every turn, and the player decides what research goal to allocate them towards: pottery, atomic energy, etc. Once the gauge fills up with the required number of points, that technology is unlocked.

Research goals are often themed as being in a tree (a "tech tree") but this isn't necessary for the way I'm using the term. Building queues in 4X games would also count as "research" here, since they are chosen in advance and your production towards them can often not be repurposed. Even something like the EQUIPMENT-BASED ABILITIES (p. 178) of *Final Fantasy IX* would count.

Analysis

I call this "research" here because of its classic implementation in *Civilization*, but more abstractly this might be called "upgrade pre-selection". Unlike money or EXP, the tokens used to "purchase" research are usually not fungible; if you cancel your gunpowder project, you cannot re-purpose the unspent points into navigation.

Compared to LEVELING (p. 232), Research heightens the "Anticipation" stage of the leveling cycle. It's rare in RPGs that I actually look ahead to which specific upgrades I'm be receiving upon my next level-up. This pattern shows me what my upgrade options are ahead of time, and makes me choose between them in advance; this strengthens my anticipation, although it lessens my surprise.



Civilization III

Money

The player is rewarded in some kind of currency, which can be spent on (typically) a wide variety of purchases including tools, upgrades, consumable items, etc.

It might not be themed as currency; if the only thing you can spend it on is abilities/spells/etc., it might be themed as experience points. The important point here is that you just have a big pot of it and you get to spend it however you choose, or save it, as opposed to LEVELING (p. 232) (where you get rewards when you reach a certain threshold) or RESEARCH (p. 236) (where you choose in advance what you're working towards).

Examples

Amassing money is obviously the goal of most business sims. To give just one example, the goal of *Recettear* is to pay off your debt; indeed, if you do *not* manage to service this debt you lose the game. It can also be the central motivator even in games that are not primarily themed as business; paying off debt is also the goal in the *Animal Crossing* games.

As an intermediate reward, money is so common in games that it's difficult to pick any particular one to talk about. But *Yakuza Zero* is a thematic example; the game is set in a red-light district in a consumeristic 1980s Japan, where money is power and the nouveau riche flaunt their cash and possessions. It's wholly on-theme, then, that the game eschews "EXP" and has you beating literal cash out of thugs and simply buying stat and skill upgrades.

Analysis

Beyond its use as a primary win condition, earning money for doing things in a game is completely ubiquitous; along with "experience points" and "loot" it's the typical way the world rewards the player for doing what they do. And it's less arbitrary feeling than awarding



The Legend of Zelda: Link's Awakening



Yakuza Zero

unthemed VICTORY POINTS (p. 33) since it's an ordinary real-world thing to which people are accustomed.

However, there's a danger to leaning too heavily on monetary accumulation was a win condition, in that it can incentivize players to "cheese" the game by repeatedly exploiting the same profitable strategies rather than explore the full range of systems the game offers.

I think this is more a danger when the monetary goal is difficult, to the point where the player worries if they don't grind selling Premium Cakes to Old Lady #4 that they might fall irreparably behind and lose the game. When the monetary goal is relatively lightweight, it can serve to keep the player reasonably on track overall (e.g. not sleeping all day, not over-spending on stupid things), while still allowing the player to explore the game freely.

In the other direction, because in many games money accumulation increases exponentially, it's often the case that a savvy player has much more money in the mid/late game than they know what to do with. This can be alleviated somewhat with money sinks (items or activities into which the player can pour lots of money), but either way it's often the case that late-game money accumulation challenges are trivialized. (It's not like, if an upcoming debt payment is 20k gold and I currently have 200k gold, I'm going to sink all 200k into a money sink. Obviously I'll keep 20k around just in case.)

Because of this, many games that start out with a "service your debt" goal transition to other goals in the late game. *Recettear* becomes more about the dungeon crawling, *Animal Crossing* about SELF-EXPRESSION (p. 39), *Touhou Mystia's Izakaya* about learning NPC PREFERENCES (p. 54).

That's why most real-world jobs are boring! It's more profitable for your employer to keep you doing the same things again and again, than to change up your job every day. Your employer cares a lot about maximizing profit and relatively little about how much fun you're having. As a game designer, you need to remember that the fun is the goal, not the profit.

Class systems

Each character can specialize in classes or jobs, for which there's a class-specific leveling/upgrade path.

Examples

RPGs have had classes since the original *Dungeons and Dragons* (1974); characters could choose to be a fighter, magic user, or cleric. Subsequent RPGs have tended to have many more; it's typical for RPGs to have about a dozen, and games with MULTICLASSING (p. 241) can effectively have hundreds of possible class combinations.

Despite this greater variety, we can still see the shadow of *D&D*'s original three classes in the modern triumvirate of *Tank* classes (tough, high HP characters who draw most of the enemy attacks), *DPS* classes ("damage-per-second", often lower-HP characters who hang back and concentrate on damage), and *Support* classes.

Analysis

Class systems are an important aspect of party-oriented RPGs, because they encourage the player to use their party members in different ways rather than treat them as interchangeable. Even in single-character RPGs, classes can increase the replay value by encouraging different playstyles on subsequent runs.

Despite the growing prevalence of non-violent RPG-like games centered around crafting or cooking or farming or medicine, I'm drawing a blank on one that offers a full class system. *Monaco: What's Yours is Mine* (2013) has different roles, but you don't progress in those roles the way you do in RPG classes.

I suppose, in 50 years of making combat RPGs, we've thought of many different valid ways for characters to approach goblin massacre, and classes encapsulate those solutions. We haven't thought of many different ways for characters to approach medical or culinary or agricultural challenges – we're happy when we find *one* way to make a compelling game out of these! Hopefully in the near future,



The Elder Scrolls: Oblivion (2006)



XCOM: Enemy Unknown (2012)



Monaco: What's Yours is Mine

as we develop more ways for characters to approach challenges non-violently, we also start to see these approaches combined into single games and encapsulated into classes.

I also want to mention classes' role in minimizing the number of decisions a player must make at any one point. Imagine a game that treats all upgrades like a supermarket: several hundred are available from the beginning, and you only have to spend the requisite EXP to purchase it. You might think this would be awesome – choice is good, so more choice should be great! – but the supermarket approach leads to two problems.

First, it can lead to analysis paralysis, a sudden stoppage of the smooth flow of the game as the player has to analyze hundreds of possible shopping carts of abilities.

Secondly, it can lead to the “tyranny of choice”: the counterintuitive result that, past a certain point, offering people more choice tends to make them *less* happy with whatever they choose. Particularly for people who seek to maximize utility with every choice they make (“maximizers”), the reminder that they had hundreds of *other* things they could have chosen leads to less satisfaction with whatever they *did* choose. “I guess that was pretty good, but was it really the best possibility out of hundreds?”

Classes help mitigate these issues by combining abilities in ways that the designer thinks lead to fun gameplay (e.g., putting stealth upgrades and archery upgrades into the same class), and makes it so at any point the player is only looking at a few possible next directions. (Even if the player is permitted to change classes, it at least cognitively *chunks* the abilities together – you might be looking at 5 possible next upgrades in your own class, or considering switching to one of 7 other classes, but you're not considering 200 different abilities every time it's time to buy an upgrade.)

This isn't the only way to do this – many games use LEVELING (p. 232), a few use UPGRADE BOARDS (p. 242) rather than discrete classes, etc. – but the point is that classes are one of many structures that designers use to avoid overwhelming the player with choices.

Challenge

What fields of human endeavor require sub-specialization in a same way as military matters? Medicine is the first one to my mind, but there are many more. Choose one, and think of how you might implement a *party*-based non-violent RPG with those specializations as classes.

Barry Schwartz. 2004. “The Tyranny of Choice.” *Scientific American* 290(4): pp. 70-75.

Multiclassing

In addition to their primary or initial class, characters can specialize in a “subclass”, combine two classes, or restart as a new class without losing all their previous class’s abilities.

Examples

A notable multiclassing system in videogames is the “Job system” from *Final Fantasy* games, appearing in primordial form in *FFIII* (1990) but reaching full fruition in *FFV* (1992). It returns in various ways throughout the series, and especially in *Final Fantasy Tactics* (1997).

In the Job system, each character can specialize in any class, and can change classes at any point. Gaining experience in a class eventually causes the player to learn that class’s skills permanently, even if they switch to another class (although there are a limited number of skill slots a character can utilize at once). This allows the player to invent creative hybrid classes with surprising synergies.

Analysis

Multiclassing greatly expands the possible playstyles of a game; if there are 12 classes, allowing the player to choose two means that there are actually 144 class possibilities. Practically, you have to throw “balancing” 144 classes out of the window; some of these combinations will have powerful synergies and others will not.

Final Fantasy Tactics is rather gloriously broken in this regard. You might think that *FFT* is meant to be a serious and reasonably-balanced strategy game, but in a way it’s an odd sort of sandbox puzzle game, where you find bizarre job combinations that trivialize ordinary encounters, play with them until bored, and eventually re-train your characters in new jobs to explore what new combinations offer. It’s a combinatoric puzzle in the guise of a tactical RPG. This is a powerful pattern, but just understand what you might be getting into when allowing multiclassing in this way; I would be cautious with this if I were designing a game meant for competitive play.



Final Fantasy Tactics



Final Fantasy V

Upgrade boards

Player characters' upgrade progress is represented as a sort of board game, often with different characters on the same board, but in different locations.

Examples

The classic example here is the "Sphere Grid" of *Final Fantasy X*. The seven characters of *FFX* do not have CLASSES (p. 239) per se, but they still have archetypes; Lulu is a black mage, Kimarhi a blue mage, etc. But instead of progressing in class-specific upgrade paths/trees, each character is represented by a pawn on a giant board-game-like graph. Lulu's being-a-black-mage is not a "class" attribute she has, but is only due to her pawn starting in the section of the board with the elementary black magic spells. You can spend "spheres" (experience points) to unlock stat bonuses, spells, and abilities, and even venture into upgrade paths "meant for" other characters.

Final Fantasy XII's "license boards" used a similar system, although in this case without the use of pawns; you could simply unlock any square adjacent to a previously-unlocked square.

Analysis

The interesting thing here, and the reason I don't consider this just another way to envision a "skill tree", is that different characters share all (or most) of the same board.

The first-time player will probably follow the path of least resistance and develop Lulu into a powerful, specialized black mage. With the right spheres, however, the experienced player can lead her to (say) Yuna's part of the board instead and develop her into a white mage in addition, or instead. This is a very different way of reifying the idea of classes and MULTICLASSING (p. 241); there is no sense that Lulu has anything as her "primary" class or "secondary" class, or rules about how many classes she can have or how she switches between them.



Final Fantasy X



Path of Exile adopted the Sphere Grid system, pretty much verbatim, although as a single-character game the distinction between this and a more conventional "skill tree" is blurred.



Final Fantasy XII

Upgradable items

Some items have upgrade paths of their own, with the player spending money/experience to develop them into more powerful versions.

Examples

Many RPGs have items that you can sharpen/strengthen, but I'll concentrate here on item upgrades that are central to progression.

The "Materia" system in *Final Fantasy VII* is a classic example. Equipment in *FFVII* has slots for orbs called "materia" which grow in experience much like a player does. As they grow, they give the equipment and/or player access to powerful abilities and spells. Characters gain levels as well, but ultimately the materia system is the most important upgrade system in the game.

The distinctive trademark of the *Disgaea* series of tactical RPGs is that the player can "enter" any item to find an "Item World" dungeon inside. The further you make it into this dungeon, the more powerful the item becomes.

Analysis

Unlike a lot of the previous patterns in this chapter, upgradable items are usually not tied to any one character – as items/equipment, they can be traded between characters. This can be helpful when the player can't effectively level all characters; an under-leveled character can still be useful if given well-leveled equipment.

One thing upgradable items give you is an enormous "experience sink". The player will typically get to a point where upgrading their characters is a matter of diminishing returns; either the next level gets too far away, or the rewards for leveling just aren't noticeable anymore. But *items* are effectively infinite, there will always be plenty of items to upgrade. So if you find you have a core game loop that players already love, to the point where they are accumulating so much EXP that they're running out of character-oriented upgrades to spend it on, consider item upgrades as a nearly-limitless EXP sink.

Part XVIII

Randomness and gambling patterns

Ability checks

Success in an endeavor is represented as a numerical threshold, and the player rolls dice (or similar) to try to meet or exceed that threshold.

Examples

Although ability checks are indelibly associated with *Dungeons and Dragons*, they were actually not included in early editions; instead, the dungeon master simply decided whether a proposed action worked by applying their common sense or making up an ad-hoc roll that would decide the outcome. The first ability check system that was intended to apply across many different kinds of situations was first proposed in an article in 1976. The *Traveller* RPG system also had ability checks in 1977. The idea of ability checks evolved only slowly in *D&D* itself and did not appear in their most familiar form – roll, add a modifier, check if it meets or exceed a threshold – until the third edition in 2000.

Like almost all the mechanics of *D&D*, ability checks can now be found in too many RPGs to count, and even in non-RPG games.

Analysis

Zooming out, ability checks are perhaps the purest example of what's sometimes called “output randomness”.

Output randomness is noise injected between the player's decision and the outcome. Examples would be the dice roll combat in *Risk* or *Memoir '44*, or the random number generation combat in *X-Com* or *FTL*.

This is contrasted with “input randomness”, which is where the randomness occurs *before* the player's decision. Drawing three cards and then choosing one would be input randomness, as would PROCEDURALLY GENERATING (p. 251) a random playfield.

Ability checks have never been a favorite pattern of mine, although part of that is that I have relatively little experience with TTRPGs, and mostly only know these through videogames.



Disco Elysium

Wesley D. Ives. 1976. “How to Use Non-Prime-Requisite Character Attributes.” In *The Dragon*, vol. 1.

I learned about this article and the more general history of ability checks from DM David's site at dmdavid.com.

Keith Burgun, 2014. “Randomness and Game Design.” At gamedeveloper.com, October 15.

In TTRPGs, these fill an important role (albeit one about which there's still debate), putting an objective check on the omnipotence of the DM to decide whether a course of action succeeds. Meanwhile, the primary downside – that players are locked out of a path due to chance – is mitigated by that DM's presence. Even in the face of successive failures locking out many options, the DM's human imagination can still weave an interesting story out of it.

In videogames, that latter mitigation cannot happen; there are only so many paths we can pre-program into an RPG. As a player, I usually don't have the confidence that the alternative path will be as interesting or rewarding as the one I initially chose – the alternative path is often just falling back on goblin-punching until the problem is punched away.

However, when the designers have instilled in me the trust that every path will contain some interesting story (as the designers of *Sorcery!* or *Disco Elysium* did), then I find myself fully willing to accept bad rolls and continue.

Wagering

The player commits money to a random or unknown outcome, and if they outcome they committed to occurs, they receive a reward.

Examples

This need not be themed as gambling, of course. In the board game *Settlers of Catan*, the player places settlements on the vertices between hexagonal tiles. Each tile has a number on it, and every time that number is rolled by any player, adjacent settlement owners receive a resource. Settlements in *Catan* are themselves wagers, on particular numbers being rolled.

Similarly, many “worker placement” games are based on wagering. The next turn’s rewards may be random, but placing workers is placing wagers on possible rewards.

The PROVISIONING (p. 170) pattern (stocking up on items to offset anticipated setbacks) is an example of the player wagering on their own future performance. E.g., “How many potions do I think I’ll need for the next section?”

Analysis

Wagering is probably as old as human culture, and there’s an inherent cognitive enjoyment to taking risks and being rewarded.

Wagering can also level the playing field between competitors or allow people who otherwise couldn’t compete a stake in the outcome. I might only last five seconds in a sumo match, but wagering on someone else’s match puts me into the game in another way.

The trivia game *Wits and Wagers* uses wagering in this way. The trouble with trivia is that it’s rarely a level playing field; some people are absolute fiends at it. Why would I bring a trivia game to the table if the victor is a foregone conclusion? *Wits and Wagers* doesn’t just let players answer questions, it lets them bet on other’s players answers. It’s entirely possible that a trivia-poor player wins, not by knowing trivia but by correctly predicting who *else* knows trivia.



Settlers of Catan



Wits and Wagers

Press your luck

The player can repeatedly gamble on a series of random rewards, but one of the possible outcomes loses them everything. The most common version of this bet is “double-or-nothing”: an even chance of doubling your wager, or losing everything you’ve gained so far.

Examples

To see an example of pressing-your-luck expanded into an entire game (beyond, of course, the classic TV game show *Press Your Luck*), I recommend the board game *The Quacks of Quedlinburg*. In *Quacks*, the players are doctors making medicines by drawing ingredients out of bags, but a wrong draw destroys the potion entirely.

The *Flip Cards* minigame in *Yoshi's Island* (1995) uses a press-your-luck mechanic. The player can flip over any of eight cards and reveal items and powerups beneath them. At any point, the player can choose to leave with their winnings. However, if the player flips over a Kamek card (representing the game's villain), the game ends and all rewards are lost.

Pressing one's luck in an action game is illustrated well by the minigame *Monkey Target* in *Super Monkey Ball*. In it, players must maneuver gliders and try to hit a small target in the distance. If you were to simply try to land on it, it would be straightforward, although not exactly *easy*. However, there are also many useful powerups (like being able to stick to the target) and score multipliers floating just out of reach. If you attempt to change course to go for these, it could mean a big payoff... but it's also *very* likely that doing so means you get off course and cannot make it to the target at all.

Analysis

Pressing one's luck can be a stressful pattern to participate in, but I think it's especially enjoyable pattern for spectators. Watching another player's anxiety at choosing whether to press further or play it



The Flip Card minigame in *Yoshi's Island*



Super Monkey Ball 2

safe, and then watching as hubris loses them everything, is a Shakespearean drama played out in a few seconds.

This is a particularly useful pattern in multiplayer games where players take turns. What is there to keep the other players paying attention? When player turns are long, other players often tune out while the current player is effectively “playing solitaire”. A press-your-luck mechanic can keep players attentive to others’ turns, eagerly awaiting their hubristic downfall.

However, it’s also important to consider the risks that pressing-one’s-luck adds to your game. Say you have a mechanic that when upgrading equipment, the player can try to do so beyond normal “levels”, but each time the player does so there’s a chance that the equipment breaks forever. From the designer’s point of view, this might just feel like a way to add some risk/reward to an otherwise boring crafting system. But from the player’s point-of-view, you’re potentially taking away something that they may have grown attached to; it’s the kind of penalty that can lose you a player permanently. Losing something completely is a major penalty and you should always be cautious with major penalties.

Gacha

The player spends a currency (often a special, limited one) to receive a random selection from a set of rewards.

Examples

This term comes from *gachapon*, the random toy vending machines of Japan and elsewhere in East Asia. The customer puts in a coin or token, and receives a small toy in a capsule. There are usually 5-8 different toys available, but if the customer wants to collect them all, they must be prepared to spend an unbounded amount of money because there's no guarantee you get the one you're looking for.

Gacha has become extremely prevalent in free-to-play games like *Genshin Impact* – the player has some collection they're working on (equipment, characters, etc.), some collection of those is available each day, and the player can spend some sort of currency to get a random one.

Analysis

A random reward, combined with the completionists' desire to "catch 'em all", together forms a particularly addictive pattern and can easily shade into abusing players' compulsive tendencies. Ethical designers should be cautious in how they use this pattern. However, the pattern doesn't make the abuse; only your implementation does, especially whether you're allowing players to dump life-ruining amounts of real-world money into this system.

If you want to ensure that the player eventually finishes their collection in a finite amount of time, consider using dependent randomness rather than independent. That is, consider all your possible rewards as a shuffled deck of cards, rather than a roll of a many-sided die. Eventually the player will work through every card in your deck and receive every kind of reward, whereas it's possible to roll that giant die thousands of times without ever happening to roll a 3.



A bank of *gachapon* toy vending machines in Japan, photo © matcha-jp.com



Each machine offers a random draw from a small collection of toys: in this case, the player receives one of six mutant parrot-banana hybrids.



Gacha has become a sufficiently widespread term that it's even presented in the UI. Here, the player is offered a "premium gacha" with a particular minimum guarantee in *Princess Connect Re:Dive*.

Procedural generation

The world is randomly created, often at the beginning of every run.

Examples

Randomizing levels is nothing new – many early arcade games had this, after all – but procedural generation (“procgen”) typically refers to the randomized creation of worlds/levels and other content that look like they could have been designed by humans: dungeons, towns, etc.

Rogue (1980) is the classic example; the terms “rogue-like” and “rogue-lite” refer to games that combine this procedural world generation with perma-death, from *Nethack* (1987) to *Angband* (1992) to *ADOM* (1994). Around 2010 it became popular to extend this formula to action genres; e.g. *Spelunky* (2008), *The Binding of Isaac* (2011), *Ziggurat* (2014), and *Nuclear Throne* (2018).

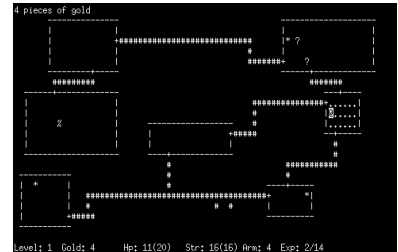
But procgen is, of course, larger than roguelikes; *Elite* (1984), *SimEarth* (1990), *Dwarf Fortress* (2002), and *Minecraft* (2009) generated highly detailed worlds for other sorts of gameplay.

Analysis

A lot has already been written about procgen and you have probably already given significant thought to whether your game is going to have a hand-crafted world or be randomly generated each time. Here, I just want to point out that this choice lies on a spectrum.

Games like *Spelunky* or *Rogue Legacy* are not random down to the individual tile; rather, the algorithm is assembling handcrafted set-pieces into larger units. And in the other direction, even human-designed worlds may not be hand-crafting every single tree.

Rather, the question to ask is “Which aspects of my game should be handcrafted, and which randomized, to support the experience I want the player to have?” You might have the early game content fixed and the late-game content randomized, the towns fixed but the overworld randomized, etc.



Rogue



Dwarf Fortress

Rando

While the gameworld remains mostly intact, aspects of the quest (like the locations of important items) are randomized.

Examples

Randomizer mods of metroidvania games, that randomize the locations of upgrades and other items, are popular among speedrunners, especially for bringing novel challenges to very-commonly run games like *Super Metroid* and *The Legend of Zelda: A Link to the Past*.

Even some commercial metroidvanias like *Bloodstained: Ritual of the Night* (2019) are including official randomizer modes.

These are making their way to RPGs as well; the *Free Enterprise* mod for *Final Fantasy IV* randomizes many aspects of the world (items, shops, etc.) and allows you to begin with the airship that makes the entire world accessible. This shortens and diversifies *FFIV* to the point where it becomes a viable “spectator sport” and tournaments are now common in speedrunning venues.

Final Fantasy V Four Job Fiesta is a randomizer challenge (and charity event) that does not even require modding the game; instead, a Twitter bot (@FF5ForFutures) randomly selects four out of *FFV*’s many job classes, and players simply police themselves to only use those four jobs during the game.

Another popular kind of self-policed rando is “Bingo”, where players are given a *Bingo* board full of random in-game challenges (e.g. “Collect 5 <whatevers>” or “Defeat <miniboss>”), and compete to complete lines before other players can. Like the item randomizers above, this can add some novelty and an additional layer of strategy to games like *Super Mario 64* (1996) that speedrunners and audiences already know very well.

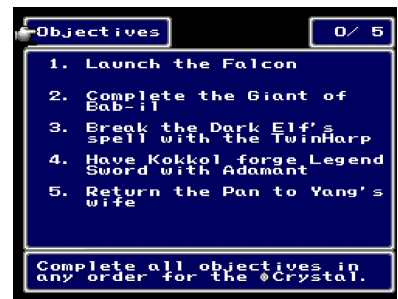
80 Days provides an interesting example of randomization within the Interactive Fiction genre. Structurally, the game is a sort of choose-your-own-adventure, where the main decision is where to go next in a global network of cities connected by various transportation methods. Alongside the authored content, however, there’s also



Speedrunners Andy and ChristosOwen race to complete a randomized version of *The Legend of Zelda: A Link to the Past*.



Bloodstained: Ritual of the Night randomizer options



You can play *Final Fantasy IV Free Enterprise* to defeat the final boss, or the randomizer can generate a set of random objectives for you.

a simple trading mechanic, where you might find a valuable telescope for sale in Berlin, and be told that it should fetch a good price in Lahore. These items are randomized each time, meaning that on each playthrough the player has reasons to try new routes. As simple as this is, I find it a very elegant solution to making the player re-play a text adventure many times and see the wealth of authored content the game offers.

Analysis

Even if you've decided that your world needs to be handcrafted to support the gameplay or story you envision, consider (especially for short games) what aspects might be randomizeable to promote replayability. Something like an item randomization mode does not necessarily take very much coding to add to your world.

However, remember that it is not *trivial*; you can't just randomly shuffle elements in your world. *Bloodstained* players complained about the randomizer creating unwinnable situations. Even though randomization may be easy to *code*, it requires some deep thought about the structure of your game and how this maps to a randomization procedure that ensures playthroughs that are neither trivial nor impossible.

Part XIX

Replay and metagame patterns

Challenge collectables

Optional collectable items are placed in such a way to provide additional challenge for more experienced players.

Examples

These are very common in *Mario* and *Yoshi* series games since roughly *Super Mario World* (1990); they're an important pattern for ensuring that the games can be completed by children and novice players, but still pose challenges for adults and experienced players.

It's also common for casual/mobile puzzle games (e.g. *Cut the Rope*) to structure puzzles so that there is one necessary win condition (in this case, feeding the candy to the monster) plus three optional challenges (collecting the stars) that contribute to a star score.

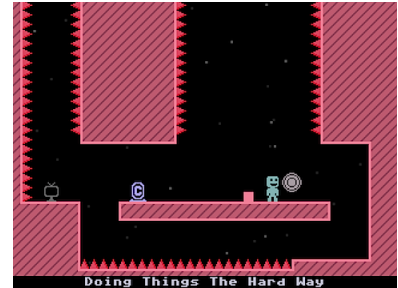
On the other end of the difficulty spectrum, *VVVVVV* (2010) is not an easy game in the first place, but gathering all 20 challenge collectables poses a significantly harder challenge.

Analysis

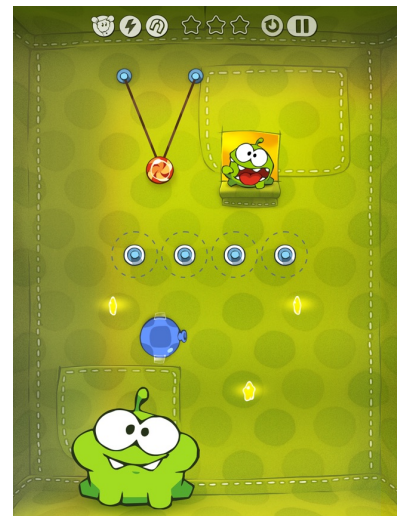
One thing to ask yourself, when putting challenge collectables in the game, is what rewards (beyond the satisfaction of completionism) to associate with them. We don't necessarily want to associate them with rewards that like health upgrades or potions that make the game easier, since these are for the players who want *greater* challenge.

Yoshi's Island (1995) uses them, in part, to unlock special, more challenging levels – an appropriate reward for a player who's proven they can handle more challenge.

As mentioned in *FAST TRAVEL* (p. 105), one of the difficult optional challenges in *Spelunky*, getting a particular object to a particular exit, unlocks fast travel to that point. Fast travel is an appropriate reward for demonstrating mastery; by letting advanced players skip easy content that they've already mastered, they get to spend more of their time on more difficult challenges.



You think it might be easy to get over this small bump in VVVVVV to obtain the challenge collectable on the other side. It is not.



Cut the Rope

Second quests

Upon completing the game, the player is offered a variant playthrough, often with a harder difficulty level, different item locations, and sometimes even different art assets.

Sometimes the player retains their stats and abilities from the previous playthrough, in which case this is also a `NEWGAME+`.

Examples

Super Mario Bros. is an early example of this. Upon beating the game, you can replay the game in a hard mode, in which some enemies are replaced by more difficult versions, all enemies are sped up, and some platforms are shortened.

The Legend of Zelda (1986) offers a second quest with even further changes – not just with upgraded enemies, but with substantial changes to the world map.

In a different direction, *Super Mario World* (1990) offered a second quest, but only changed the assets (to an Autumn/Halloween theme).

Although it was not presented as a second quest exactly, the inverted castle in *Castlevania: Symphony of the Night* (1997) can be seen as this too. By “completing” the game meeting certain criteria, the player discovers that the game is only half done, and the game continues in an inverted and more challenging version of the original world map.



Super Mario Bros.

NewGame+

Some of the character's stats or abilities carry through to the next playthrough.

Examples

This term was coined in *Chrono Trigger* (1995). One of the draws of the game was that it offered 13 different endings... but should the player have to complete the game from scratch 13 times to see them all? Instead, upon completing the game once, the player can take their endgame stats with them when starting a new game, in order to breeze through early levels and experience new endings more quickly.

Some NewGame+ modes are also SECOND QUESTS (p. 256), but not all. *Chrono Trigger* just offers the same quest again, and the variety is offered by the fact that endings that were previously too difficult to attempt in a scratch run are now available.

There's also an interesting interaction between NewGame+ mechanics and LANGUAGE LEARNING (p. 204). In some games like *Final Fantasy X*, some NPCs speak a different language that you gradually learn through the game. When the player is allowed to carry language knowledge through to a new game (or as it's done in *FFX*, load language progress from a previous save file), the player's second playthrough can be a little richer than the first story-wise, because previously-incomprehensible dialogue is now clear.



Chrono Trigger

Prestige/Ascension

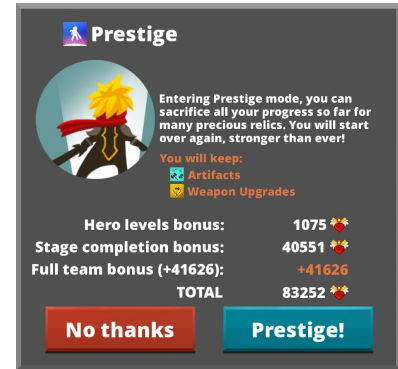
The player can restart the game in exchange for rewards like stat multipliers, increased experience growth, or new abilities and game-play systems.

Examples

Endless/idle/clicker RPGs like *AdVenture Capitalist* or *Tap Titans* have a sort of NEWGAME+ particularly tailored for endless progress. These are not games that the player can complete; they simply level up quasi-infinitely. Nonetheless, the player will inevitably reach a point where the next reward seems impossibly far away. Instead of slogging through or quitting, the player can decide to restart from level 1 but with an incremented “prestige” level that comes with its own rewards. These rewards typically allow the player to progress even further on this new playthrough, until they again inevitably stagnate, and ascend again.

Analysis

This differs from an ordinary NEWGAME+ in that it usually doesn’t require completing the game. Rather, you’re cashing in your partial progress in return for (typically) a sort of special currency, which can be used to buy permanent rewards à la META-PROGRESSION (p. 259).



The “Prestige” reset in *Tap Titans*

Meta-progression

In each playthrough, the player starts from scratch in a randomized world, but can upgrade some aspects of their character or world permanently.

Examples

This is a pattern most associated with modern rogue-likes/lites. In *Rogue Legacy* (2013), for example, you play as a series of characters in a long family lineage of adventurers. While your new character starts from scratch each time, you can spend some of your rewards on upgrading your family castle.

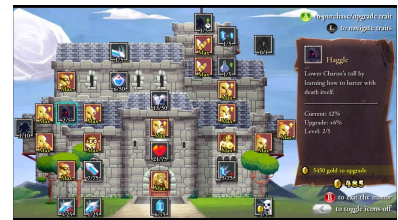
Sometimes the meta-progression is more limited. In *Spelunky* (2008), completing certain challenges in-game (like carrying an item all the way from the first levels to a later level) opens up FAST TRAVEL (p. 105) to later levels. This lets the player skip early levels with which they're already very familiar.

Analysis

RPG-style LEVELING (p. 232) is the original and most pervasive implicit difficulty slider, letting novice and cautious players exchange playtime for stat/ability upgrades.

Traditional rogue-likes, however, lack this difficulty slider. Even though they have leveling within single runs, the only thing the player can bring to a new run is their increased knowledge of the game systems; there is no purely mechanical way to convert playtime to an easier game. This is all well and good, but it does mean that more novice/conservative players are missing a familiar difficulty “knob”, and this relegates the genre to a more niche audience.

So we're increasingly seeing popular rogue-likes implement some kind of meta-progression, so that players still feel they can make progress in the long run, even if they've reached their personal skill/knowledge limit.



Individual characters in *Rogue Legacy* are short-lived, but your castle is permanent.

Leaderboards and histograms

The player is shown other player's scores, or learns where they sit in the distribution of player scores.

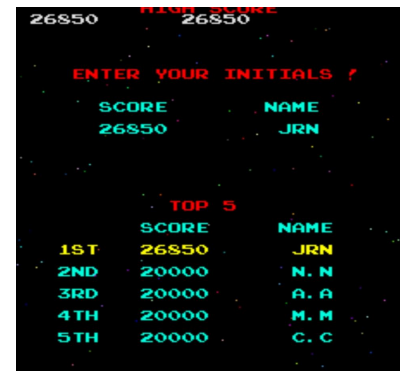
Examples

High score tables were a fixture of arcades, compelling players to compete for local dominance (and thus spend more money) even in single-player games.

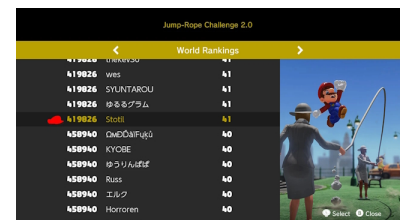
In the internet age, leaderboards are often global (and can thus feel unattainable for the average player), so some games (e.g. *Tap Titans*) also show the player local leaderboards.

It was interesting to see leaderboards brought back in *Super Mario Odyssey* (2017) for minigames like jump-rope. These simple activities would not be especially compelling on their own, but some players really will put in significant effort when it's framed as a competition.

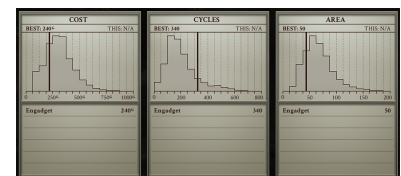
Special mention goes here to the histograms common in Zachtronics games, which show the player's position in a histogram of player scores. Since Zachtronics games are frequently AUTOMATION/PROGRAMMING (p. 133) games, there are a near-infinite number of ways to solve each puzzle, and different players might have different goals for their "code": the lowest cost, the highest speed, the fewest elements, etc. To acknowledge this diversity, the player is shown multiple histograms (e.g. Cost, Cycles, and Area in *Opus Magnum*). Having multiple histograms is an excellent way to acknowledge that there are many different things people strive to do in these games, and increases the replay value as you can strive to optimize something different next time.



Galaga



The jump-rope leaderboard in *Super Mario Odyssey*



In *Opus Magnum* (2017), the player is shown how their inventions compare to other players', in their cost, how many cycles they take to complete, and how large they are. This player is ahead of the curve in cost and area, but behind the curve in terms of cycles.

Timelapses and highlight reels

The player sees a replay of their game as a sped-up-timelapse or as a series of gameplay highlights.

Examples

The first example I can think of was in *Civilization* (1991); when the game ended you got to view a historical timelapse on a zoomed-out global map. Many strategy games have imitated this since, even bite-sized strategy games like *Dice Wars* (2006).

Replay videos and highlight reels also make frequent appearances in racing games, for example in *Mario Kart* games since *Mario Kart: Super Circuit* (2001). Replay videos are particularly important for building-focused racing games (cf. LEVEL EDITORS, p. 83) like the *Trackmania* series, since they're a way to show off your creative designs to other players.

Outside of these genres, *Overwatch* (2016) is a notable example, with their "Play of the Game" highlighting a play their algorithm thought was particularly important.

Analysis

Timelapses are particularly satisfying when the game has an aspect of AREA CONTROL (p. 25), where the player's area of influence (hopefully) expands as the game progresses. During the moment-to-moment of gameplay, the player's point of view is often too limited, or the change is happening too slowly, for the player to directly experience their overall progress. Timelapses thus allow the player to experience the grand sweep of gameplay in a way they didn't the first time around.



Civilization



Mario Kart 8 (2014) even lets you edit the highlight reels, e.g. choosing what racers to follow.

Alternate endings

There are multiple possible ways for the story to end.

Examples

The first thing people will probably think of is multiple-ending RPGs like *Chrono Trigger* (1995), dating sims like *Tokimeki Memorial* (1994), or raising sims like *Princess Maker* (1991), which had a whopping 74 endings. Collecting these endings – good and bad – is a sort of meta-game in itself.

Readers who grew up with *Choose Your Own Adventure* books will probably remember using a finger as a bookmark and turning to a choice you *knew* would be wrong, just to see what gruesome end would befall the main character. This is also the largely-forgotten pleasure of the easy deaths and snide narration in *Sierra* graphical adventures like the *King's Quest* or *Space Quest* series.

In retrospect, it was not the *deaths* that made these games cruel, but the hassle of manually saving and the loss of progress if you forgot. When playing Inkle's adaptations of the *Steve Jackson's Sorcery!* gamebooks, I was surprised how much I enjoyed finding all the failures, due to their forgiving auto-save and time-rewind systems.

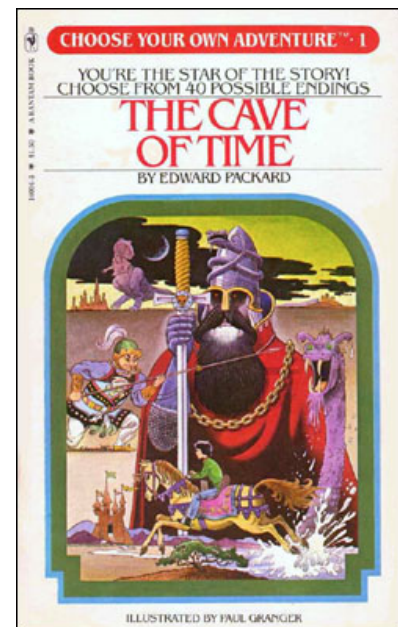
Analysis

Obviously, if every plot decision leads to a branching path with a different ending, any reasonably-sized game will end up with an infeasibly exponential number of endings.

But linear games can have lots of alternate endings too – even aside from the *Sierra* route where linearity is achieved by all wrong choices leading to death. *Chrono Trigger* is mostly linear, and the ending is determined by when the player chooses to tackle the final boss. *Star Ocean: The Second Story* (1998) is also mostly linear and the main plot only ends in a fixed way, but the ending is stitched together from many possible little vignettes determined by the game's friendship/romance system (cf. MATCHMAKING, p. 61).



In *Chrono Trigger*, defeating the final boss before securing the dominance of ancient humans leads to an ending in a world populated by dinosaurs instead.



CYOA #1: *The Cave of Time*



Space Quest III

Index of Games

- 20 Questions, 208
 3D Crazy Coaster, 116
 80 Days, 102, 167, 252

 A Fake Artist Goes to New York, 215
 A Good Snowman is Hard to Build, 193
 A Monster's Expedition, 216
 A Short Hike, 20, 37, 103, 110, 111, 117, 228
 Abalone, 44, 123, 149
 Abzû, 111, 114
 ActRaiser, 81
 Adventure, 183
 AdVenture Capitalist, 258
 Adventures of Lolo, 130
 AER: Memories of Old, 111
 Air Hockey, 136
 Air Inferno, 59
 Airport Mania: First Flight, 56
 Alba: A Wildlife Adventure, 79, 161
 Alchemy, 190
 Alex Kidd in Miracle World, 218
 Amidar, 25, 27, 128
 Among Us, 13
 Ancient Domains of Mystery, 251
 Angband, 251
 Angry Birds, 88, 135
 Animal Crossing series, 74, 84, 237, 238
 Animal Crossing, 86
 Animal Crossing: New Horizons, 86
 Anno series, 166
 Anno 1800, 166
 Anno 2205, 166
 Ape Escape series
 Ape Escape, 49, 158, 159
 Ape Escape 2, 29
 Archery, 135
 Artillery Duel, 135
 Arx Fatalis, 206
 Assassin's Creed, 110
 Assembly Required, 130
 Asteroids, 41, 119
 Atelier series, 152, 163, 164, 180
 Escha & Logy, 164
 Atomic Punk, 170
 Auditorium, 138
 Auro: A Monster-Bumping Adventure, 44, 149
 Automachef, 133
 Avalanche, 46
 Azkend, 196

 Baba is You, 193, 226
 Back to Bed, 128
 Backgammon, 24, 103
 Bang, 215
 Banjo Kazooey, 30
 Baseball, 135

- Bastion, 224
- Baten Kaitos, 171
- Batman: Arkham series, 108
- Battleship, 208
- Bear and Breakfast, 84
- Bejeweled, 38, 196
- Betrayal at Krondor, 209
- Billard japonais, 136
- Billiards, 136
- Binary Land, 122
- Bingo, 252
- Bioshock, 28, 172, 190
- Black Box, 208
- Blackwell series, 156, 202
- Blockade, 41
- Bloodstained: Ritual of the Night, 165, 252
- Bloons, 38
- BomberBoy, 170
- Boom Blox, 88
- Botticelli, 208
- Bouncing Babies, 46
- Boxxle, 193
- Braid, 214
- Brave Firefighters, 59
- Breakout, 46, 136
- Bridg-It, 28
- Bridge, 36
- Bridge Constructor, 87
- Building blocks, 87
- Bulls & Cows, 208
- Bunmei Korokoro Game: Egg, 27
- Burning Rangers, 59
- Burnout Paradise, 45
- Bust-a-Move, 196
- Buzz Wire, 44

- Cadence of Hyrule, 150, 218
- Caesar 3, 131
- Cafeteria Nipponica, 57
- Canabalt, 38
- Capoeira, 46
- Captain Blood, 204
- Captain Toad: Treasure Tracker, 122
- Carcassonne, 33, 34, 81, 190, 220
- Carmen Sandiego, 35
- Carto, 161, 190, 220
- Castle Wolfenstein, 49
- Castlevania series, 31, 165, 228
 - Castlevania: Symphony of the Night, 104, 256
- ChainShot, 196
- Chickenfoot, 36
- Chinese Checkers, 24
- Choose Your Own Adventure series, 262
- Chrono Cross, 235
- Chrono Trigger, 71, 99, 127, 228, 233, 257, 262
- ChuChu Rocket, 64, 128
- Circuit Puzzle, 190
- Circus Atari, 46
- Circus Circus, 117
- Citadels, 215
- Cities: Skylines, 82
- City Connection, 26
- Civ series
 - Civilization, 261
- Civilization series, 60, 216, 233, 236
 - Civilization III, 236
 - Civilization IV, 32
- Clash of Clans, 83
- Closure, 219
- Cloudpunk, 101
- Clue, 35
- Clue/Cluedo, 35
- Collosal Cave Adventure, 96
- Connect-the-dots, 90
- Conquests of the Longbow: The Legend of Robin Hood, 209
- Contact, 208
- Continuity, 186
- Contraption Maker, 134

- Cook, Serve, Delicious, 56, 57
- Cooking Mama, 57
- Cosmic Soldier, 32
- Counterfeit Monkey, 211
- Crane games, 120
- Crazy Taxi, 100, 138
- Creative Contraptions, 134
- Crusader Kings series, 61
 - Crusader Kings II, 60, 226
- Crusaders of Centy, 62
- Crush the Castle, 88, 135
- Crypt of the Necrodancer, 108, 168
- Cryptarithm, 34
- Crystal Quest, 30, 118
- Cult of the Lamb, 230
- Cut the Rope, 255
- Cute Knight, 60

- Dōkyūsei, 52, 70
- Dad's Puzzler, 185
- Daikoukai Jidai series, 204
- Dance Dance Revolution, 108
- Dark Cloud series, 82
 - Dark Cloud 2, 91
- Dark Souls, 31
- Dave the Diver, 56
- Day of the Tentacle, 71, 99, 183
- Dear Esther, 16, 20
- Death Gate, 206
- Densha de Go, 116
- Desktop Dungeons, 224
- Diablo series, 222
 - Diablo II, 179, 222
- Dice Wars, 261
- Dig Dug, 112
- Digit, 192
- Digital Devil Saga: Megami Tensei, 32
- Diner Dash, 13, 56, 68
- Dino Run, 45, 48
- Dinosaur Fossil Hunter, 85
- Diplomats, 149
- Disco Elysium, 199
- Disgaea, 24, 149, 226, 243
- Dishonored, 50, 225
- Doki Doki Panic, 124
- Dolphin Olympics, 109
- Dominoes, 36, 134, 190
- Donkey Kong Country II, 95
- DOOM, 183
- Dorfromantik, 34, 81, 190
- Dr. Mario, 58
- Dragon Age series, 52
- Dragon Quest, 35
- Dragon Quest series
 - Dragon Quest, 177, 200, 232
 - Dragon Quest III, 181
- Dragon Slayer IV: Draslay Family, 124
- Drive Mobile, 103
- Drug Wars, 102, 173
- DSY: Don't Shoot Yourself, 41
- Dungeon Master, 234
- Dungeon Siege, 62
- Dungeons and Dragons, 239, 245
- Dwarf Fortress, 251
- Dysmantle, 165

- Earthtongue, 65
- Easter egg hunt, 29
- Eek the Cat, 128
- Eggerland series, 130
- Eight-ball, 136
- Einstein's Puzzle, 34
- Elden Ring, 159
- Elder Scrolls series, 172, 234
 - Arena, 209, 210
 - Morrowind, 105, 206, 210
 - Oblivion, 239
 - Skyrim, 49, 210
 - The Elder Scrolls: Arena, 201
- Eleusis, 213
- Elite, 102, 173, 251
- Else Heart.Break(), 133

- Emergency: Fighters for Life, 59
- Empire Builder, 28
- Enen Angel, 186
- English Country Tune, 193
- Etrian Odyssey, 96, 123
- Eulerian path, 98
- Euro Truck Simulator, 13
- Euro Truck Simulator 2, 101
- Eurorails, 28
- Excitebike, 83
- Factorio, 133, 166, 189, 207
- Fairune, 217
- Fallout series, 62, 172
- Fallout 3, 152, 155, 172, 232
- Fallout 4, 154
- Fallout: New Vegas, 57
- FarCry series, 114
- FEZ, 16, 48, 159, 204, 211, 228
- FIFA 23, 15
- Fifteen Puzzle, 185
- Final Fantasy Legend, 213
- Final Fantasy series, 113, 143, 150, 206, 228, 241
- Final Fantasy, 17, 200
- Final Fantasy Crystal Chronicles, 48
- Final Fantasy II, 146, 234
- Final Fantasy III, 241
- Final Fantasy IV, 252
- Final Fantasy IX, 97, 100, 143, 158, 178, 236
- Final Fantasy Tactics, 24, 241
- Final Fantasy Tactics Advance, 81, 224
- Final Fantasy V, 98, 155, 217, 241, 252
- Final Fantasy VI, 156
- Final Fantasy VII, 63, 181, 243
- Final Fantasy VIII, 174, 177
- Final Fantasy X, 56, 204, 211, 242, 257
- Final Fantasy X-2, 180
- Final Fantasy XII, 133, 161, 242
- Final Fantasy XIV, 75
- Final Fantasy XV, 76, 78
- Lightning Returns: Final Fantasy XIII, 180
- Fire, 46, 59, 121
- Fire Emblem series, 61
- Fire Emblem: Awakening, 61
- Flappy Bird, 111
- Fluxx, 226
- Fortnite, 180
- Freeze Tag, 147
- Fresh Start Cleaning Simulator, 66
- FTL, 245
- Gabriel Knight 3, 183
- Galaga, 141
- Gals Panic, 27, 90
- GAMES Magazine Giant Book of Games, 188
- Gazillionaire, 102
- Genshin Impact, 92, 250
- GeoGuessr, 160
- Getting Over It with Bennett Foddy, 110, 120
- Gin Rummy, 154
- GIRP, 110, 120
- Go, 25, 28
- Goblins 2: The Prince Buffoon, 189
- Golden Krone Hotel, 223
- GoldenEye 007, 127
- Golf, 16, 135
- Gone Home, 20
- Goonies II, 99
- Gradius, 141
- Grand Theft Auto series, 45, 79, 100, 113, 114
- Grand Theft Auto III, 201, 228
- Grand Theft Auto Online, 45

- Grandia series
 - Grandia, 146, 147
 - Grandia II, 76
- GROW Cube, 189
- Grow Home, 64, 110, 117
- Guess Who?, 35
- Guild of Dungeoneering, 83
- Guitar Hero, 92, 108, 187
- Gwent, 174
- Gyromite, 128

- Habbo Hotel, 84, 86
- Hack, 143
- Hack-n-Slash, 133
- Halo series, 113
- Hamiltonian path, 98
- Harvest Moon series, 11
 - Harvest Moon, 16, 52, 64
- Hashi, 90
- Hatoful Boyfriend, 52
- Heaven and Earth, 138
- Heiankyo Alien, 130
- Her Story, 35, 208
- Hex, 28
- Hidden Folks, 13, 29, 155, 156
- Hide and Seek, 29
- Hitman, 181
- Hot and Cold, 208
- Hot Springs Story, 56, 84
- House of the Dead, 91
- Humpty Dumpty, 136
- Hunt the Wumpus, 29, 160

- I am Level, 117, 136
- I Love Hue, 194
- I Spy, 208
- Ice Cold Beer, 138
- Ichidant-R, 220
- Ico, 76, 124, 127
- Ikaruga, 43, 108
- Indiana Jones and the Fate of
 - Atlantis, 96, 98, 158, 199
- Invisible, Inc., 49
- Ishidō, 190

- Jak series, 116
 - Jak II, 116, 201
- Jenga, 87
- Jet Set Radio, 25, 116
- Jetpack Joyride, 30, 111
- JezzBall, 27
- Jigsaw Puzzles, 190
- Jigsaw puzzles, 90
- Journey, 20
- Joust, 111
- Juggling, 46

- Kaboom, 46
- Katamari Damacy, 30, 95, 121
- King's Quest series, 156, 262
 - King's Quest V, 96, 184
- Kingyo-sukui, 120
- Kirby series, 140, 143
 - Kirby and the Amazing Mirror, 143
 - Kirby's Epic Yarn, 84, 85
- Kuru Kuru Kururin, 44
- Kwirk, 193

- Labyrinth (board game), 186
- Labyrinth (physical toy), 138
- Legacy of the Wizard, 124
- Legend of Dragoon, 108
- Legend of Grimrock, 123
- Legend of Heroes/Trails
 - series, 124
- Legend of Mana, 78, 81, 83, 93, 204, 213
- LEGO, 39
- Lemmings series
 - Lemmings, 24, 28, 126, 128, 134
 - Lemmings 2: The Tribes, 64
- Lemonade Stand, 169
- Lethis: Path of Progress, 131
- Life and Death, 58
- Lightmatter, 219
- Limbo, 146
- Line Knight Fortix, 27

- Linguistics Olympiad, 205
- Link-a-Pix, 90
- Little Healer, 58
- Lode Runner, 83, 130
- Lolo series, 130
- LOOM, 92
- Loop Hero, 83, 230
- Lunar Lander, 46

- Madden series, 15
- Mafia, 215, 218
- Magic: The Gathering, 174
- Magical Vacation, 225
- Magicka, 109, 206
- Mahjong, 154
- Mahjong solitaire, 196
- Makai Toushi SaGa, 213
- Maniac Mansion, 156
- Mao, 213
- Marble Drop, 188
- Marble Madness, 118
- Mario & Luigi series, 122
 - Mario & Luigi Superstar Saga, 228
- Mario & Sonic at the Olympic Games, 121
- Mario Kart series, 114, 146, 261
 - Mario Kart 8, 104, 261
 - Mario Kart DS, 219
 - Mario Kart: Super Circuit, 261
- Mario series, 15, 64, 141, 150, 255
 - New Super Mario Bros., 48
 - Super Mario 64, 37, 39, 103, 150, 228, 252
 - Super Mario Bros., 24, 26, 118, 140, 256
 - Super Mario Bros. 2, 118, 124
 - Super Mario Bros. 3, 220
 - Super Mario Galaxy, 33, 37, 48, 140, 144, 159
 - Super Mario Galaxy 2, 33, 115, 118
- Super Mario Maker, 83
- Super Mario Odyssey, 140, 141, 144, 181, 260
- Super Mario Sunshine, 66
- Super Mario World, 30, 115, 255, 256
- Massive Chalice, 61
- Master of Pottery, 85, 163
- Mastermind, 172, 208
- Matchstick puzzles, 192
- Memoir '44, 245
- Memory, 196, 220
- Mercenary Force, 123
- Merchant Prince series, 216
 - Machiavelli the Prince, 216
- Metal Gear, 49
- Metal Gear Solid, 49
- Metroid series, 178, 228
 - Metroid, 147, 228
 - Metroid Prime, 157
 - Metroid Prime II: Echoes, 99
 - Super Metroid, 148, 216, 228, 233, 252
- Microcosm, 58
- Microsoft Flight Simulator, 111
- Might and Magic, 123
- Minecraft, 57, 82, 84, 112, 133, 165, 194, 251
- Minesweeper, 160
- Mini Metro, 28
- Mini-Golf, 136
- Minit, 72
- Minit Fun Racer, 72
- Mirror's Edge, 115
- Monaco: What's Yours is
 - Mine, 49, 181, 239
- Monkey Island series
 - Monkey Island 2: LeChuck's Revenge, 211
 - The Secret of Monkey Island, 36
- Monster Hunter, 161
- Mouse in the Maze, 83

- Mousetrap, 134
 Moving Puzzle, 185
 Mr. Do, 112
 Mr. Driller, 41, 112
 My Singing Monsters, 86, 93

 NANACA+CRASH, 135
 Need for Speed series, 45
 Nethack, 172, 224, 251
 Network, 190
 Ni no Kuni 2: Revenant
 Kingdom, 82, 230
 NiGHTS into DREAMS, 30
 Ninuki-Renju, 27
 No Man's Sky, 86, 157, 204
 No Place Like Home, 66
 Nomad, 202, 205
 Nonograms, 90
 Nuclear Blaze, 59
 Nuclear Throne, 251

 Octodad: Dadliest Catch, 120
 Octopath Traveler, 218
 ODDADA, 92
 Oekaki Logic, 90
 Okhlos, 123
 One Way Heroics, 48
 Operation, 44, 58
 Opus Magnum, 133, 260
 Oregon Trail, 170
 Ori series
 Ori and the Blind Forest, 48,
 229
 Ori and the Will o' the
 Wisps, 219
 Othello, 27, 32
 Out There, 204
 Outer Wilds, 70, 72, 104, 115,
 158, 201
 Overcooked, 57
 Overwatch, 261

 Pac-Man, 25, 96, 144
 Pachinko, 136
 Pachisi, 103
 Paint-by-numbers, 90
 Paper Mario: The Origami
 King, 186
 Papers, Please, 173
 PaRappa the Rappa, 108
 Parodius, 46, 141
 Path of Exile, 242
 Pazzak, 174
 Peggle, 136
 Pente, 27
 Pentominoes, 192
 Persona series, 32
 PGA Tour 2K23, 16
 Pharaoh, 131
 Pic-a-Pix, 90
 Pick-it, 192
 Picross, 90
 Pikmin, 124, 131
 Pilotwings, 46
 Pinball, 24, 46, 136
 Pinball Construction Set, 83,
 86
 Pipe Dream, 28, 172, 190
 Pipe Mania, 190
 Pipeline: The Oil Game, 28
 Pipes, 190
 Planetscape: Torment, 199
 Plant Cat: First Blossom, 64
 Player Unknown's
 Battlegrounds, 48
 Please, Touch the Artwork, 188
 Plinko, 137
 Pokémon series, 32, 62, 121
 Diamond/Pearl, 21
 Gold/Silver, 63
 Red/Blue, 21
 Pokémon Snap, 91
 PolyPlay, 114
 Pong, 136
 Pontifex, 87
 Portopia Serial Murder Case,
 35
 PowerWash Simulator, 25, 66

- Press Your Luck, 248
 Princess Connect Re:Dive, 250
 Princess Maker, 189
 Princess Maker series, 262
 Princess Maker, 60
 Princess Maker 2 Refine, 60
 Princess Remedy in a World of
 Hurt, 58
 Prismatica, 185
 Project Hospital, 58
 Project Makeover, 180
 Proteus, 20, 23
 Pushover, 134
 Puzzle Bobble, 196
 Puzzlescript, 193
 Pyramid solitaire, 196

 QIX, 27
 Quest for Glory II, 96
 QWOP, 120

 Race the Sun, 37
 Raid on Bungeling Bay, 81
 Raiden Fighters, 45
 Raiden series
 Raiden DX, 45
 Raiden Fighters, 45
 Railroad Ink, 190
 Railroad Tycoon, 28
 Rampage, 26
 Ratchet & Clank series, 31
 Ratchet & Clank, 172
 Ratchet & Clank Future: A
 Crack in Time, 115
 Ratchet and Clank, 116
 Real Golf 2011, 135
 Recettear: An Item Shop's
 Tale, 52, 169, 237
 Resident Evil 4, 167
 Reus, 191
 Reversi, 27, 32
 Rhythm Heaven, 108
 Rimworld, 61
 Risk, 245

 Riven, 205
 RobotWar, 133
 Rock Band, 92
 Rogue, 222
 Rogue Legacy, 230, 259
 RollerCoaster Tycoon 3, 116
 Roundabout, 44
 Rubik's cube, 185
 Rudra no Hihō, 205, 206
 Rummy, 154

 Saboteur, 28
 Saints Row series, 45
 Sandcastles, 88
 Scorched Earth, 135
 Scotland Yard, 29, 160
 Scrabble, 36
 Settlers of Catan, 33, 247
 Shadow of Memories, 71
 Shadow of the Colossus, 110,
 113
 Shadows in the Forest, 219
 Shadows over Camelot, 215
 Shannon Connection Game, 28
 Shape Sorter, 194
 Shapez, 166, 189
 Shariki, 196
 Shermue series, 11
 Shiritori, 36
 Shogi, 32
 Sid Meier's Alpha Centauri,
 216
 Sid Meier's Pirates, 102, 159
 Sim series
 SimCity, 39, 81, 83, 191
 SimCity 2000, 79
 SimCopter, 83
 SimEarth, 65, 251
 SimLife, 65
 The Sims, 61
 SimCity series, 166
 Simon, 220
 Sky: Children of the Light, 20
 Sleepwalker, 128

- Sly Cooper and the Theivious Raccoonus, 49
- Smushi Come Home, 114
- Snail Sprint, 115
- Snake, 41
- Snakes and Ladders, 24
- Snipperclips, 192
- SNOLF, 137
- Snooker, 136
- Sokoban, 193
- Sokobond, 193
- Sonic series, 116, 117, 137, 148
 - Sonic Adventure 2, 116
 - Sonic Spinball, 137
- Soreyuke: Amida-kun, 128
- Space Panic, 130
- Space Quest series, 262
 - Space Quest III, 262
 - Space Quest IV, 71
- Space Taxi, 46
- SpaceChem, 13, 133, 207
- Speedway, 103
- Spelunky, 105, 118, 172, 251, 255, 259
- Spice of Life: Carrot Edition, 165
- Spiderman II, 201
- Splatoon, 25
- Splitter Critters, 128
- Spot-the-difference, 155
- Spyfall, 215
- Spyro the Dragon, 30, 31
- Squids, 44
- Star Control series
 - Star Control II, 199, 202, 205
 - Star Control III, 81
- Star Ocean series, 52, 61
 - Star Ocean, 53
 - Star Ocean: The Second Story, 163, 262
- Star Trader, 102
- Star Wars: Knights of the Old Republic, 174, 210
- Stardew Valley, 11, 15, 18, 54, 60, 62, 64, 68, 73–75, 79, 85, 93, 105, 152, 154, 168
- Starflight, 199, 202
- StarFox series, 114
- Starseed Pilgrim, 64, 214
- Steamworld Dig, 112
- Steins;Gate, 71
- Stephen’s Sausage Roll, 193
- Steve Jackson’s Sorcery series, 262
- Stix, 192
- Strange Attractors, 138
- Subnautica, 84, 111, 157
- Subwar 2050, 158
- Subway Surfers, 116
- Sudoku, 90
- Suikoden series, 230
 - Suikoden, 177, 230
 - Suikoden II, 54, 230
- Sumo, 44, 46, 149
- Super Mario Maker, 93
- Super Mario RPG, 108, 116
- Super Meat Boy, 148
- Super Metroid, 30
- Super Monkey Ball series, 119
 - Super Monkey Ball, 248
 - Super Monkey Ball 2, 248
- Superman 64, 114
- Surgeon Simulator, 58
- Tabletop Simulator, 28
- Tafl, 27
- Tag, 29
- Tales of Symphonia, 221
- Tangrams, 192
- Tank Wars, 135
- Tap Titans, 258, 260
- Taxi Ball, 138
- Tempest, 172
- Tenchu: Stealth Assassins, 49
- TerraFirmaCraft, 57
- Tetris, 13, 192
- Tetrominoes, 192
- The 7th Guest, 72

- The Binding of Isaac, 251
- The Boy and his Blob, 62
- The Castle Doctrine, 83
- The Crew: The Quest for Planet Nine, 36
- The Edifice, 204
- The Factory, 133, 188
- The Firemen, 59
- The Floor is Lava, 46
- The Fool's Errand, 211
- The Forgotten City, 72
- The Horde, 81
- The Incredible Machine, 134
- The Island of Dr. Brain, 192
- The Last of Us, 127
- The Legend of Zelda: Majora's Mask, 72
- The Lost Files of Sherlock Holmes, 35, 202
- The Lost Vikings, 124, 126
- The Nightmare Cooperative, 122
- The Price is Right, 137
- The Quacks of Quedlinburg, 248
- The Settlers, 131, 166, 230
- The Sims, 13
- The Sumerian Game, 236
- The Ultimate Noah's Ark, 197
- The Witcher III, 174
- The Witness, 214
- Theme Hospital, 56, 58
- Theme Park, 169
- Thief: The Dark Project, 49, 172
- Thurn und Taxis, 25
- Ticket to Ride, 28
- Time Crisis, 91
- Tiny Wings, 95
- ToeJam and Earl, 222
- TOEM, 91
- Tokimeki Memorial, 13, 52, 262
- Tokyo Highway, 87
- Tony Hawk's Pro Skater, 116
- Tony Hawk's Underground 2, 109
- Top Management Decision Simulation, 169
- Torchlight, 62, 223
- Touch Me, 220
- Touhou Mystia's Izakaya, 52, 54, 170, 218, 238
- Touhou Project
 - Double Spoiler, 43, 91
 - Shoot the Bullet, 43, 91
- Touhou series, 45
- Townscaper, 82
- Trackmania series, 261
- Tradewinds, 102
- Treasures of the Aegean, 37
- Triominoes, 190
- Triple Triad, 174
- Tumbleseed, 138
- TwinBee, 46, 141
- Twitch Plays Pokémon, 121
- Twixt, 28
- Typoman, 211
- Tyranny, 206

- Ultima series
 - Ultima I: The First Age, 202
 - Ultima Online, 168, 234
- Uncharted Waters series, 204
 - Uncharted Waters Origin, 204
- Under the Moon, 104
- Undertale, 11, 43, 108
- Unicorn Overlord, 61, 133
- Uno, 36
- Unpacking, 194
- Urbanix, 27

- Vertex, 90
- Virtual Villagers series, 61
 - Virtual Villagers: A New Home, 61

- Virtual Villagers: New Believers, 32
- Viscera Cleanup Detail, 66
- Volleyball, 46
- VVVVVV, 255
- Waking Mars, 65
- Warcraft: Orcs and Humans, 126
- Wario Land 3, 140, 144
- Wario Land II, 37, 140, 144, 148
- Werewolves, 215, 218
- What is the Name of this Book?, 210
- Where in the World is Carmen Sandiego?, 29, 35
- Where's Waldo, 29, 155, 156
- Wilmot's Warehouse, 194
- Windows Solitaire, 15
- Wing Commander: Privateer, 102, 173
- Wipeout, 44
- Wire Loop, 44
- Wits and Wagers, 247
- Wizardry, 32, 96, 123
- Wordle, 172, 208
- World of Goo, 87
- Worms, 135
- Woten, 16, 140, 143
- Wrecking Crew, 26
- XCOM series
 - X-COM: UFO Defense, 167, 245
 - XCOM: Enemy Unknown, 24, 232, 235, 239
- Y, 28
- Yakuza series, 11
 - Yakuza Zero, 120, 237
- Yeti Sports: Pingu Throw, 135
- Yoku's Island Express, 117, 136
- Yonder: The Cloud-Catcher Chronicles, 18
- Yoshi's Island, 115, 150, 248, 255
- You Have to Win the Game, 104
- Zebra puzzles, 34
- Zelda series, 19, 98, 113, 183, 217
 - A Link to the Past, 99, 118, 252
 - Breath of the Wild, 110, 152, 171, 209
 - Four Swords, 122
 - Link to the Past, 217
 - Link's Awakening, 193, 237
 - Oracle of Seasons, 99
 - The Legend of Zelda, 155, 158, 256
 - The Legend of Zelda: Majora's Mask, 70
 - The Legend of Zelda: Ocarina of Time, 92
 - The Legend of Zelda: Twilight Princess, 113
 - Twilight Princess, 159
 - Zelda II, 219
- Zen Bound, 27
- Ziggurat, 251