Analyzing the Challenge of Navigation through the Metroid Series

Keaton Hoek
University of Utah
keatonhoek@gmail.com

José P. Zagal
University of Utah
jose.zagal@utah.edu

ABSTRACT

We discuss how navigation works in video games with a focus on how it provides challenge for players. Informed by work on how humans navigate real world space we propose a framework to guide the analysis and design of games. The framework considers three steps in the process of navigation: Destination (determining where the player needs to go), Routing (determining how to get there), and Execution (traveling along the route). We further articulate our framework by showing some of the ways that difficulty in navigating game spaces is managed. Our analysis was conducted on games in the Metroid series; a hallmark of the Metroidvania genre. Metroidvania games emphasize exploration, navigation, and non-linearity in terms of how the gameworld is traversed. Although we limit our analysis to 2D games, the framework is extensible to other kinds of spaces.

Keywords

navigation, challenge, Metroid series, Metroidvania, difficulty

INTRODUCTION

Games are fundamentally about arbitrary goals and the obstacles that stand in the way of players achieving those goals (e.g. Suits 1978). The nature of those obstacles, and what players must do to overcome them, create challenges. Challenge is important to most game-playing experiences (e.g. Fullerton et al. 2008; Schell 2008; Cuerdo et al. 2023) and designers continue to innovate in ways to challenge players across the variety of games that exist e.g. testing players' ability to stay on a beat, challenging their hand-eye coordination and reflexes, and/or taxing them mentally with convoluted puzzles. Importantly, the concept of challenge is different from difficulty. When we conflate challenge with difficulty we ignore that there are different forms of challenge (Brandse and Tomimatsu 2013).

Vahlo and Karhulahti (2020) separate challenges in videogames into five types: analytical, physical, insight, socioemotional, and foresight. These categories refer to what skills are being tested via certain challenges. Denisova et. al. (2020) organize challenge into four groups: cognitive, performative, emotional, and decision-making. Meanwhile, Bowman (2018) suggests a group of five demands from games, analogous

Proceedings of DiGRA 2025

to challenge, in the form of cognitive, emotional, physical (split between the two varieties of controller and exertional), and social. Flint, Denisova, and Bowman have since merged their earlier (and separate) work to arrive at seven dimensions of challenge: performative, emotional, controller, cognitive, social, exertional and decision-making (Flint et al. 2023).

Scholars have also looked at game genres and how their game design features create challenge. For example, Smith et. al. (2008) show how levels in platforming games can be subdivided into areas of challenge while Wehbe and colleagues (2017) empirically tested how different game elements common to the genre (e.g. size of platforms, jump complexity) contributed to a game's difficulty. Aponte and colleagues (2011) studied challenge in a custom-designed first-person shooter game in which waves of enemies attacked the player. Other scholars have focused on specific types of tasks players must perform, trying to unpack how challenge is modulated within the same type of task. For example, Pusey et. al. developed a tool for analyzing the cognitive challenge that different cognitive-based puzzles in videogames provide (Pusey et al. 2021). Cuerdo and Melcer (2020) provide a framework of the ways that in-game death is represented in games and how different designer choices lead to different challenge for players. Their work led to the creation of a taxonomy to describe challenge in the context of different kinds of player failure in games (Cuerdo et al. 2023).

We build on this tradition by focusing on a specific task, navigation, and how different game designs modulate the kind of challenge they provide. Our framework for better understanding the challenge of navigation in games is based on an understanding of human navigation in real world spaces. We illustrate this framework via an analysis of several games in the *Metroid* series. By choosing a single type of task within a significant game series we aim to provide depth and nuance in our analysis. Our framework can also support game designers in how they consider the role of challenge in their games when it comes to the task of navigation.

NAVIGATION IN GAMES

The experience of navigable space is a key element of games and how we play them (Nitsche 2008). Navigation in games is the act of understanding, orienting, and moving in the game space the player is presented with, e.g. a complex maze or an open field. Most games with spatial representations provide players with some level of challenge in terms of navigating that space: reaching the end of a level, locating a hidden object, etc. For some games, navigation is the primary source of challenge. This impacts the configuration of spaces in videogames. "Video games favor maze structures since navigating them already constitutes a challenge, which can be further amplified by obstacles along the path, such as enemies, chasms or projectiles" (Fernandez-Vara 2007). Fernandez-Vara discusses mazes and labyrinths and notes how videogames favor the former – spaces that provide multiple options for traversal. As videogame technology has evolved we have seen the size and complexity of spaces increase dramatically. Thus, navigation in game spaces has become more important with designers experimenting with novel ways to enable and support players in navigating them. For example, in extremely large game spaces, navigation might be impossible without a map (Toups Dugas et al. 2019). Many techniques have been developed to support navigation: e.g. checkpoints and shortcuts (Barker 2019), breadcrumbs (Neuschwander 2019), travel companions that point out interesting places to visit (Caldwell 2019), landmarks and paths (Liszio and Masuch 2016), arrows that point towards specific directions (Moura and El-Nasr 2015), player-created and shareable maps (Horbiński and Zagata 2022), teleporters (Rollings and Adams 2003), gated paths (Gazzard 2010), different kinds of visual cues (Zhan et al. 2024), and more. Of course, there are often significant distinctions to be made depending on those spaces and how they are presented and represented to the player, i.e. whether they are graphical, text-based, continuous, discrete, 2D, 3D, etc. (Fernandez-Vara et al. 2005).

THE CHALLENGE OF NAVIGATION

Effective navigation requires that the navigator be able to "mentally manipulate spatial and contextual information" (Merabet et al. 2012). Darken et. at. (1999) note that navigation should be understood as a process and stress the importance of distinguishing between locomotion and the cognitive subtasks that drive it: "know[ing] where to go and how to get there". Darken and Sibert (1996) also identify three different types of "wayfinding tasks": naïve search wherein the navigator "has no a priori knowledge of the whereabouts of the target", primed search wherein the target's location is known (but a route to the target is unknown), and exploration where there is no target. Sometimes these tasks can occur in sequence, e.g. a naïve search may lead to a primed search once a location is discovered. Chen and Stanney (1999) also identify three processes or tasks involved in wayfinding: cognitive mapping and information-generating, decision-making, and decision-execution. Similarly, Merabet and colleagues note that navigating effectively consists of "gather[ing] relevant spatial information for orientating, route planning and path execution" (Merabet et al. 2012). Thus, the process of navigation consists of three steps: determining what the intended destination is, determining how to get to that destination, and moving or traveling to that destination.

We assume that the first two steps in that process, both cognitive in nature, are similar when comparing videogames and real life. We do not make that assumption for the third step, which is largely physical instead of cognitive. Locomotion in videogames usually requires pressing buttons on a controller, moving a thumb stick, or sliding a mouse across a surface instead of walking, running, or swimming. We can formulate the steps of navigation as challenges from the player's perspective as follows:

1. **Destination**: The challenge of determining the destination the player is required to reach (i.e. Where is the player supposed to go?)
2. **Routing:** The challenge of figuring out a route that will lead to the destination. (i.e. How is the player supposed to get there/can they get there?)
3. **Execution**: The challenge of successfully following the route leading to the destination. (i.e. Is the player able to get there?)

Destination corresponds with the information-gathering phase of wayfinding (Chen and Stanney 1999), as well as the naïve search from Darken and Silbert (1996). It serves as a cognitive challenge, testing the player's memory and problem-solving. When in the destination phase of navigation, the player is looking for a target location with limited knowledge of where the target is. In this phase the player must take in information from the game and gameworld to decide where their destination is. A treasure hunt is a game in which **Destination** is the primary source of challenge: clues are provided to players who must then determine the location of the hidden treasure. Mazes with clearly marked exits trivialize **Destination** as a source of challenge: "Looking in on the maze from above, the walker knows where she starts and where she is supposed to go" (Fernandez-Vara 2007). Here the challenge lies in figuring out how to get to the exit.

Routing corresponds with the "decision-making" phase of navigation (Chen and Stanney 1999) and Darken and Sibert's primed search (1996). Here, a player knows where they must go, their "target location" or destination, but must still determine how to make it there. For example, a player knows they must reach a treasure chest behind an unbreakable glass wall, but do not know how to reach it. This phase's cognitive challenge tests players' understanding of the gameworld (e.g. known locations and how they are oriented in relation to each other), what they can currently achieve (e.g. the affordances the player has for traversing the game world – how high they can jump, can they climb?, fly?, etc.), and how these combine to determine viable routes to get to the destination from their current location. This challenge includes being able to determine whether the destination is even accessible to them at that point in time.

Execution correlates to the "locomotion" phase (Chen and Stanney 1999). It also aligns with physical-controller challenge (Bowman et al. 2018) and Flint et. al.'s exertional challenge (2023). **Execution** challenges the player's ability to physically interact with the game – generally via an input device. Typically, a player must press certain combinations of buttons at the right moments in order to guide the character they control to the destination via the route planned. Even if a player knows their destination is across a pit, and they know they have to use the falling rocks as temporary platforms to jump across, they might still struggle with performing this action. A game that relies on **Execution** to provide challenge in the context of navigation would be an obstacle course. The destination is clear (usually in plain sight) with the route also well known (almost always a straight-line), so the challenge is derived from how well a player overcomes and avoids the obstacles along the way.

Cognitive Load and the Challenge of Navigation

We have examined the basic steps of the task of navigation and framed them in the context of challenge. However, challenge is also modulated by additional factors external to navigation. While it is outside of the scope of this paper to examine all of those factors (see Hoeg 2008 for an overview and study), there is one worth discussing because of how it relates to the challenges of **Destination** and **Routing**: cognitive load.

Cognitive Load (CL) theory was developed from the need to better understand and support problem solving by humans. Specifically, it looks at how human memory works and its limitations. For example, it requires greater mental effort to recall information while working on a task. While developed in the context of teaching and learning, CL theory also relates to the ease of task completion. Engström et. al. note that "CL selectively impairs… subtasks that rely on cognitive control but leaves automatic performance unaffected." (2017). While their work examined tasks related to driving, this concept is easily applied to game playing.

Simply put, a high cognitive load will lead to impaired ability to perform cognitively-dependent tasks. If the player has a lot to remember at once it may not limit their ability to move and act within the game world, but it will keep them from easily and successfully performing cognitive tasks, like decision making. This has a direct relationship with navigation, specifically destination and routing, as these tasks rely on the player's memory and mental abilities. A player suffering greater cognitive load will find it harder to succeed at these challenges. As such, CL is a direct way to scale the difficulty of destination finding and routing (e.g., providing the player with more things to keep track of as a way of increasing difficulty).

METROID SERIES

Metroid was released for the Nintendo Entertainment System (NES) in 1986 (Nintendo 1986). It is a 2D side-scrolling game in a sci-fi setting where you play as bounty hunter Samus Aran as she struggles to defeat the Space Pirate leader Mother Brain while exterminating "Metroids": artificial alien creatures that suck the life out of whatever they latch onto. The game's popularity led to several sequels on different platforms over the years.

The Metroid series stands as a hallmark for its sub-genre: the *Metroidvania*. The name is a portmanteau of the *Metroid* and *Castlevania* series in which, broadly, "players are tasked with finding a goal somewhere in the game; initially the player's access to regions of the game are restricted by their abilities, such as the height they jump to. By collecting powerups, the player's abilities change and new areas of the world become accessible" (Cook et al. 2012). Unlike open-world games, in Metroidvania games, the gameworld is generally constituted by a series of rooms and pathways that connect them (Mawhorter et al. 2022). However, the genre places an emphasis on exploration, with "a degree of non-linearity and player discretion" (Camper 2008).

As such, the *Metroid* series serves as an emblematic example of how navigation is utilized in games as a source of challenge. Iantorno's description of *Super Metroid* arguably describes the series as a whole: "instead of presenting distinct levels or explicit paths to travel, Super Metroid's sprawling world gradually opens to players as they collect upgrades and construct an understanding of how to use them" (Iantorno 2021). To be clear, our work will by no means be a comprehensive examination of all the ways that navigation provides challenge to players. For example, we limit our analysis to navigation in two-dimensional spaces.

Game	Release	Platform
Metroid	1986	Nintendo Entertainment System (NES)
Metroid II: Return of Samus	1991	Game Boy
Super Metroid	1994	Super Nintendo Entertainment System (SNES)
Metroid Fusion	2002	Game Boy Advance (GBA)
Metroid: Zero Mission	2004	GBA
Metroid Dread	2021	Switch

Table 1: *Metroid* games analyzed

METHODS

With our initial framework for examining the challenge of navigation (i.e. the steps of **Destination**, **Routing**, and **Execution**) and our choice of game series in place we had to determine which games to study. Our goal was to include those games that provided for the broadest view of the design space while limiting the time and effort of analysis. To inform our decision regarding which games to include we relied on our personal experience as players, strategy guides, game reviews, and critical commentary and discussion. We excluded games whose gameplay was significantly different from what is typically considered within the Metroidvania genre (e.g. *Metroid Prime Pinball*), games that did not use a 2D perspective (e.g. *Metroid Prime* and sequels), and re-releases, remakes, and compilations. *Metroid Zero Mission*, a remake of *Metroid*, was included because it has significant changes from the original including added items, areas, and mini-bosses. This narrowed down the list of games to examine to six (see Table 1). We then played these games extensively, reviewed online videos of others playing the games, read and examined strategy guides, and discussed our observations to highlight and understand more broadly the kinds of navigational challenge each provided to its players, and how the game's design modulated that challenge.

THE CHALLENGE OF DETERMINING THE DESTINATION

This challenge taxes a player's problem-solving capabilities while also testing their memory. Destination-finding corresponds to a naïve search when the player must determine a target destination without prior knowledge of its location (Darken and Sibert 1996). Generally, the player must make sense of information provided to them by the game and make connections between that information and what they know (and have learned) about the game world. The information provided to the player thus guides players while also modulating the difficulty of determining the destination. Guidance comes in many forms, from direct instruction provided in a tutorial to a cryptic riddle scrawled on the back of a cave wall. Our analysis found two broad strategies for guidance that are distinct, but not mutually exclusive: indirect and direct.

Indirect Guidance

Indirect guidance consists of instruction or hints supplied diegetically, i.e. via the game world rather than the game's interface. Importantly, this is not necessarily guidance the game's character would see or understand. In a 2D side-scrolling game the player can effectively see through walls in ways the in-game character cannot. Therefore, a room's layout and how much the player can see beyond it serves as an indirect guide towards determining a destination. Other examples include locked doors or visible collectibles that are out reach. Here the player implicitly assumes a destination: the other side of the door or the collectible's location. *Metroid* features color-coded doors that need unlocking with special abilities. Only the blue colored doors (yellow in *Dread*) can be opened by default. We call these examples positive guidance: because they direct the player toward a desirable goal. Positive guidance lowers the difficulty of cognitive tasks by freeing up mental resources: the implicit assumption operates as a hint of sorts. The lack of such guidance would therefore result in higher perceived difficulty instead.

The Metroid series also has examples of negative guidance that leads the player astray, causing them to overlook potential solutions and increasing difficulty. Both *Metroid* and *Super Metroid* start with a trivial example of negative guidance. In *Metroid* the player begins in a long corridor, while *Super Metroid* starts on the largely flat planet surface. In both games the player can move right or left. In platforming games progress generally lies to the right of the screen (Arnott 2017). If the player tries that here they quickly meet an impassable wall. Instead, progress lies in the opposite (non-typical) direction. Here the player's implicit assumption leads them in the wrong direction. While this example is trivial, it sets the scene for both games' more liberal use of negative guidance later.

Figure 1: *Super Metroid* starting area, with explorable areas to the left and right

Metroid also provides negative guidance by obscuring connections (pathways) between rooms thus excluding some rooms from consideration as destinations. These connections are hidden by using destructible blocks that are identical to their indestructible counterparts except that they break when shot. The entrance to the Varia Suit chamber in *Metroid* is one such example (see Figure 2).

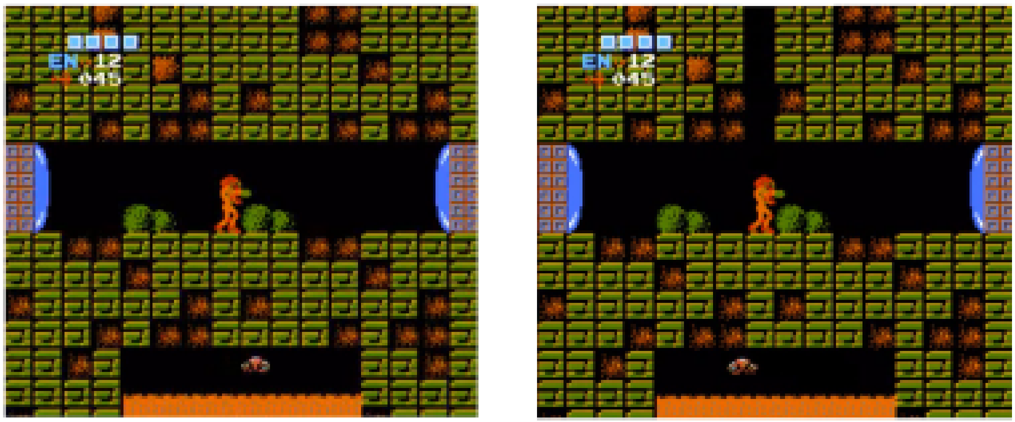

Figure 2: The entrance to the Varia Suit chamber is hidden (left) until the destructible blocks above Samus are shot (right).

However, there is also nuance in how positive and negative guidance is implemented. While in the older titles destructible blocks must be discovered through trial-and-error, from *Super Metroid* onward they could be revealed using the Power Bomb: an upgrade that identifies the location of every destructible block on the current screen and tags them with an icon indicating the ability needed to destroy them. Thus, what was once negative guidance becomes positive guidance.

Further nuance is found in tandem use of positive and negative guidance. In *Metroid Dread* there is a room where progress halted by hidden destructible ceiling blocks (see Figure 3). However, there are enemies flying above the destructible ceiling. This goads the player into shooting them. Here a player might miss an enemy and hit a destructible block instead, thus revealing that the space above is a destination. The existence of enemies in the sub-area above the player can also implicitly suggest that sub-area as a destination depending on the enemy attack possibilities. If an enemy cannot attack the player from where they are located, why would the designers have placed them there? Thus, that location implicitly becomes a destination.

Figure 3: A room in *Metroid Dread* where enemies above hint at the possibility of destructible blocks beneath them

Direct Guidance

Direct guidance most often takes the form of in-game text and other UI elements: the game (and its designers) communicate directly to the player. This can take the form of glowing waypoints and arrows (Moura and El-Nasr 2015), on-screen tutorial text, or in-game conversations with NPCs (Moura and El-Nasr 2015). In the *Metroid* series we found direct guidance usually comes in two types that vary by their specificity: high-level and low-level.

High-level instructions guide the player in the general direction without revealing the specific details of their destination. *Metroid Fusion* does this frequently. After an in-game task is completed, Samus is often forcibly returned to a central location. Then, an in-game character tells the player the general location of their next destination. By lowering the potential locations in which the player's next destination is located, the player has a lower cognitive load, and reduced challenge. In *Metroid II: Return of*

Samus (Nintendo R&D1 1991), the main shaft used for traversal is blocked by lava that recedes only after Samus has eliminated all the metroids in adjacent areas. The receding lava allows her to descend into previously inaccessible locations. When Samus eliminates the last metroid – "the screen shakes and the sound rumbles. The game design thereby signals to the player that the way is clear back at the main shaft" (Arnott 2017). Here, the player receives direct guidance, i.e. there is a new destination, but the player needs to figure out where it is.

Low-level instruction is more specific, often to the point of telling the player precisely where their next target is located (thus trivializing the challenge). In *Metroid: Zero Mission* the player finds "Chozo" statues they must interact with. Upon doing so the in-game map appears with a glowing waypoint (i.e. destination, see Figure 4) showing exactly where the player must go next. While waypoints neuter the difficulty of **Destination** they usually appear off the currently explored map grid thus maintaining the **Routing** challenge of navigation, i.e. "how do I get there?.

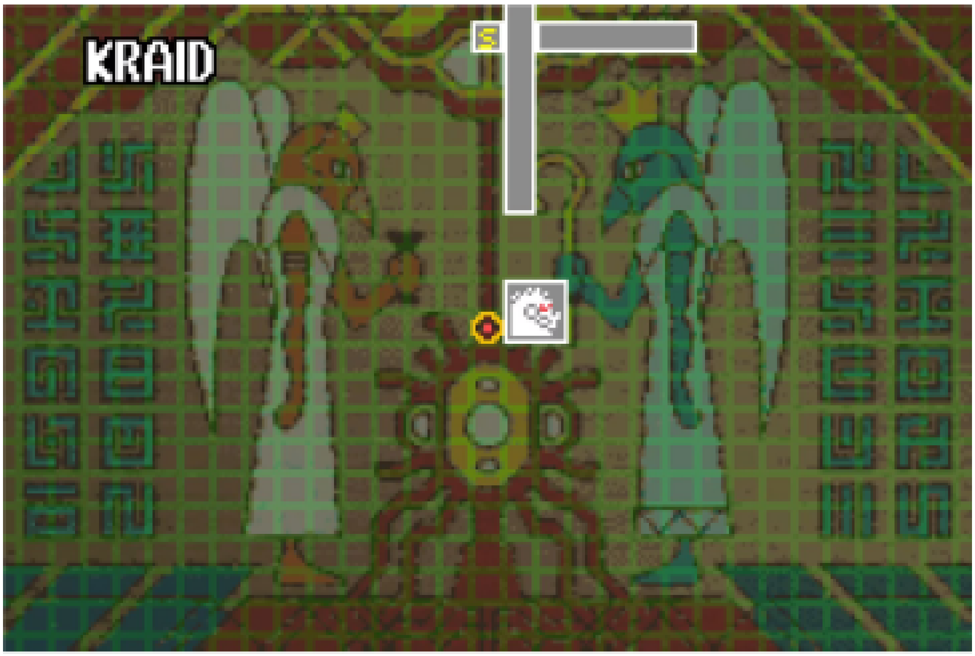

Figure 4: A waypoint (yellow circle with orange dot) shown on an incomplete map in *Metroid: Zero Mission*

Maps are a staple of the Metroid series, first appearing in *Super Metroid*. Due to being a direct UI element, the map affords direct and indirect guidance. By varying how much the map shows at once, or how much it shows beyond what the player sees directly, the challenge of destination can be modulated in difficulty. For example, in the series, the map is generally updated with locations as they are discovered making it easier to remember different locations in the game and their spatial relationship. Additionally, in *Super Metroid*, some rooms are "map rooms'" from which Samus can "download maps of (some) areas she has yet to explore" (Arnott 2017). This directly guides the player in terms of new destinations because *Super Metroid*'s map displays previously visited rooms as pink, while those unexplored are blue (Arnott 2017). Furthermore, "the maps are incomplete: Only map rooms, save rooms, and energy and missile recharge rooms are marked explicitly" (Arnott 2017). This can leave empty spaces that indirectly guide towards destinations (i.e. an empty space could be a secret room).

	Indirect	Direct
Positive	e.g. Locked doors	e.g. Waypoints on map
Negative	e.g. Hidden paths	e.g. Intentional lying

Table 2: A diagram of the two Types (Indirect and Direct) and how they interact with the two Values (Positive and Negative)

As noted, determining a destination is a cognitive challenge. The challenge's difficulty can be modulated by managing (including denying) the information the player is provided with. Metroid series game designers use two types of guidance: indirect and direct. Indirect guidance is information implied through the design and structure of the gameworld, while direct guidance often provides the player information through the UI. Guidance of any kind has a value: positive or negative. Positive guidance leads the player toward a desired destination, while negative guidance leads them astray. By altering how much guidance of either type the player receives and its valence (positive or negative), the difficulty of the challenge can be tuned by the developer (see Table 2).

THE CHALLENGE OF ROUTING

Routing is also primarily a cognitive challenge (Flint et al. 2023). Routing relies on the player's memory and problem-solving, and is affected by cognitive load. To determine a viable route to a destination the player must combine what they know about the world (e.g. locations and their spatial relationships) with knowledge of the games' rules and affordances when interacting with the world (e.g. the player character has the ability to jump, and when frozen, flying enemies stop in mid-air and can be jumped on as if they were platforms). Then, players formulate a plan.

We distinguish two types of plans: micro-plans and macro-plans. Micro-plans relate navigation through a small space, usually a single screen (e.g. how to get to the other side). Macro-planning is larger in scope, for instance by taking into consideration several screens worth of information resulting in a higher-level working plan to reach a destination.

Micro-Planning

Micro-planning almost always takes place at the scale of a single screen or room in the game world. At this scale the challenge often lies in understanding the "moving pieces" (including enemies) in an area and where potential routes to a destination may lie. The more "moving pieces", the harder it will be for the player to plan a route. This is also intensified if there is danger (i.e. there is no time to stop and plan because of moving enemies or environmental hazards). Micro-planning is susceptible to cognitive overload. In a larger the room with complex interactions between the "moving pieces" it is easier to become overwhelmed and make a mistake. This directly affects players' ability to take in and utilize information including anticipating and understanding how the "moving pieces" move, what routing possibilities they offer, and what can be done to create routes (e.g. enemies whose destruction creates spaces to move through).

As a simple example, in the Brinstar area of *Super Metroid* there is a tall multi-screen elevator shaft-like room filled with enemies traveling left to right (See Figure 5). To progress the player must use the Ice Beam ability to freeze enemies thus creating temporary platforms (Pelland et al. 1994) (see Figure 9). Here the player must visually recognize and discriminate between the gameworld's elements to determine a viable route (e.g. which elements are dangerous and where an exit is) and identify uses of their abilities and how they interact with the elements in the gameworld to create possible routes (e.g. recognize that a gap is able to be jumped over, or that the Ice Beam ability can create temporary platforms).

Figure 5: Tall room in *Super Metroid* showing Samus as well as one of the floating enemies

Time limits are also often part of micro-planning because can easily modulate the difficulty: giving the player less time to think creates stress and takes up cognitive resources. When timed, a player is more likely to reach detrimental states of cognitive load thus increasing the difficulty of a task. In *Metroid Dread* there are several zones that are sectioned off from the rest of the map. Inside them are relentless stalkers known as EMMIs that will kill Samus if they catch her. This increases the challenge of navigation across the entire zone since the player has to either plan a route while sneaking around and avoiding the EMMI, or act under a time limit if discovered.

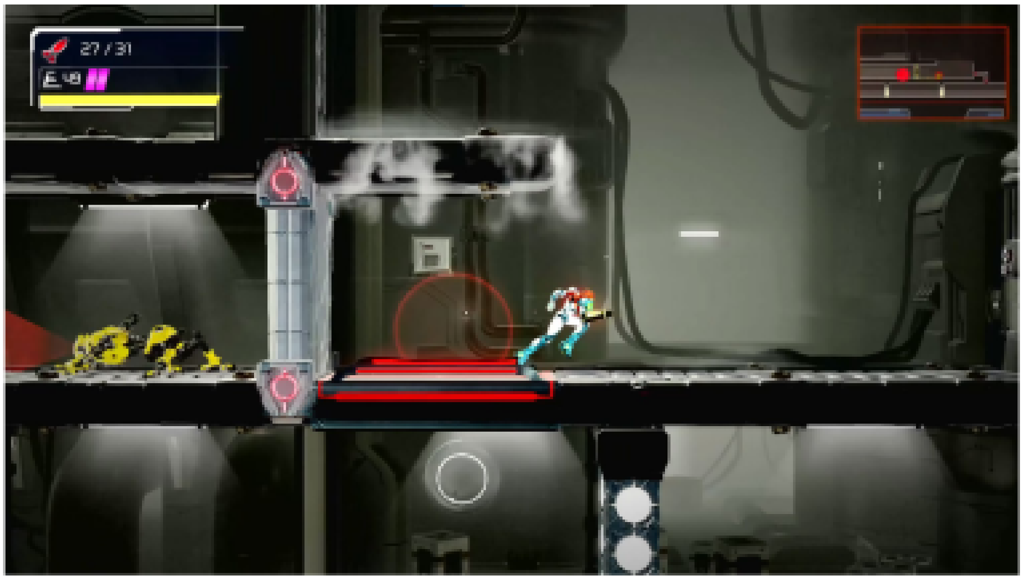

Figure 6: An EMMI Zone, showing the yellow-colored EMMI to the left

Macro-Planning

Macro-planning is analogous to micro-planning (e.g. consider abilities, knowledge of the gameworld) but at a larger scale and generally without the real-time component (e.g. consider moving objects for planning purposes). Again, the challenge is primarily cognitive with the player needing to think about various areas at once all across the gameworld to plan a route from, say, room A to G. In addition to how the rooms are spatially related to each other, players must consider the status of obstacles: perhaps the player recently obtained a key or ability allowing access to a previously inaccessible area or new routing opportunities now exist across the gameworld.

Metroid's designers provide different forms of guidance to assist (or hinder) a player's macro-planning. For example, making an area distinctive, including notable visual landmarks, means it is likelier to stick out in the player's memory, thus making it easier to work into a route. Consider the entrance to Kraid's lair in *Super Metroid* (see Figure 7). The first time the player encounters this grotesque doorway they are unlikely to be able to reach it, as it is too high off the ground. This changes once the player has found the Hi-Jump boots, located in a completely different part of the gameworld (Pelland et al. 1994). Once the boots are obtained, the player must recall how to get to the entrance – something that is easier due to the doorways distinctive art.

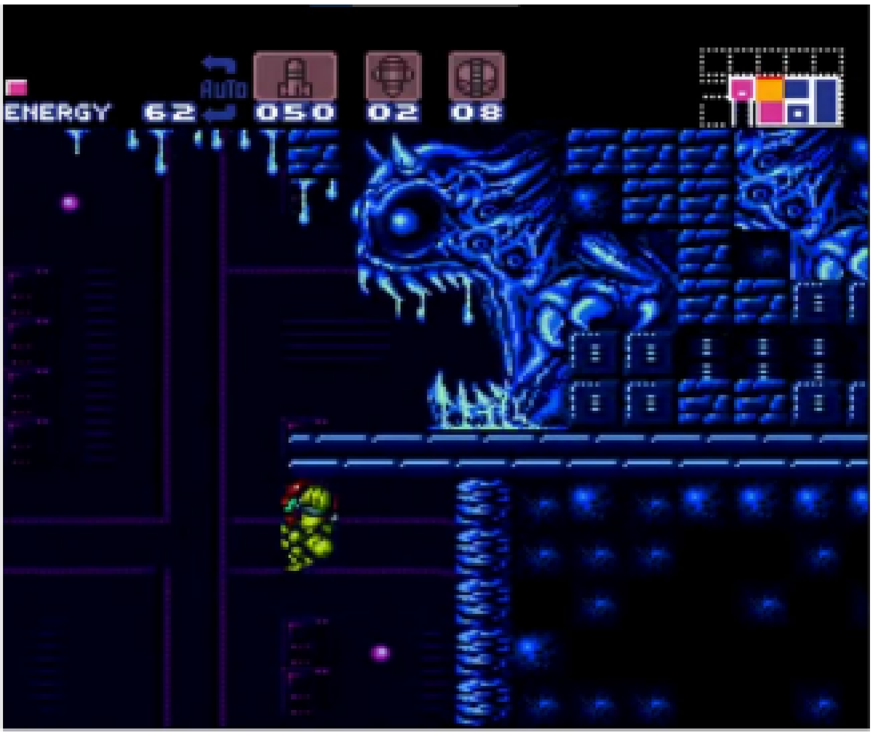

Figure 7: Kraid's Lair entrance as it appears in *Super Metroid*

While every game in the Metroid series divides its world into various zones (thus reducing cognitive load via compartmentalization), two games take this a step further. *Metroid Fusion* employs a hub, with the other zones acting as spokes from the central area. *Metroid II: Return of Samus* (*Metroid II*) features a central tunnel the player must repeatedly visit to access other areas. Laying out the gameworld in this way is an expansive form of positive indirect guidance since players know they always must travel through the central zone before heading anywhere else. Having to keep track of fewer entrances and exits (e.g. nodes in a potential route) lowers the demands on working memory, therefore reducing cognitive overload.

THE CHALLENGE OF EXECUTION

Execution is distinct from **Destination** and **Routing**, as overcoming it is not primarily a cognitive task. Execution corresponds with the locomotion phase of navigation (Darken et al. 1999) and represents the actual travel to a desired location. For videogames, this generally means pressing buttons on a controller - the correct ones and at the correct time, so the player-controlled character follows the desired route to the intended destination. Under Flint et. al.'s model (2018) **Execution** combines physical and control challenge: physical challenge relates to "player speed and reaction time in response to on-screen elements" (Flint et al. 2023) while control challenge relates to "the physical input required to interact with the game", i.e. basically how challenging it is to engage with the game itself (Flint et al. 2023).

Physical Challenge in Execution

The physical portion of the execution task directly deals with the player's proficiency with using the character's movement and abilities to get around. This varies in a number of ways, but nearly all physical tasks like this can be modulated via the required precision. Precision describes the latitude the player has in timing their button presses in order to succeed. For example, if a game character can jump 3 meters, jumping over a 1-meter pit does not require much precision: i.e. there is greater latitude in terms of when the player must press the "jump" button to still succeed. The character could jump within 2 meters of the edge of the pit and still succeed. However, jumping over a 2.9-meter pit requires significantly more precision: the player must press the jump button when the character is close to the edge of the pit (within 10cm).

Figure 8: A simple, low difficulty, high forgiveness platforming example from *Metroid II: Return of Samus*

These kinds of challenges are used all throughout the Metroid series. Every game we analyzed includes numerous examples of jumping over or onto something. Figure 8 shows an example from the central shaft area of *Metroid II* where Samus can easily jump to the platform on the right. In Figure 9, to reach the top, the player must use the Ice Beam to freeze the flying enemies, then use the frozen enemy as a platform to shoot the next one in sequence. This continues until the player has reached the top (the area is much larger than a single screen). This is further complicated by the fact the frozen enemies do not stay frozen forever. As such, there are two aspects of precision jumping at play here, properly landing on a frozen enemy and timing the next jump so the player does not fall from their thawing perch. This challenge can be tuned difficulty-wise by altering the location of the enemies thus making the jump (once the enemies are frozen) harder or easier. Similarly, altering how long the enemies stay frozen would also affect the challenge.

Figure 9: Samus climbing the "Red Tower" by standing on the frozen hovering enemies

While the challenge of execution is largely physical, stress and low concentration easily lead to mistakes and failure. This leads us to the concept of "forgiveness": how much, or little, the player's progress is set back for failing. High forgiveness means the setback is minor, while the setback for low forgiveness is high. In *Metroid*, setbacks are usually minor – the player simply attempts the challenge in the same room again. But, in more dire situations, Samus could lose health, and as a result, die. Dying in Metroid games generally results in restarting from the room last saved in, potentially a major setback based on hard it is to navigate to where they were when they died.

Figure 10: Samus falling through the floor to land on the ground below, showing a high *forgiveness* execution challenge

Most platforming in *Metroid* games has high forgiveness. For example, in *Super Metroid* there is a room Samus must dash across while the floor falls beneath her. Should the player fail, Samus safely lands on the floor below. This has high forgiveness, since the player can merely walk back to the start of the room. However, there are game areas with low forgiveness. *Metroid Fusion* has an aquatic area called Sector 4 where "if you touch the water, you'll be electrocuted" (Cassady et al. 2001, 100) (See Figure 11). There are platforming sections that, when failed, result in Samus falling in water. These sections are less forgiving because of the loss of health (and possible death) making the area's platforming more stressful and difficult. As the official strategy guide notes "[l]anding in the water is almost certain death" (Cassady et al. 2001).

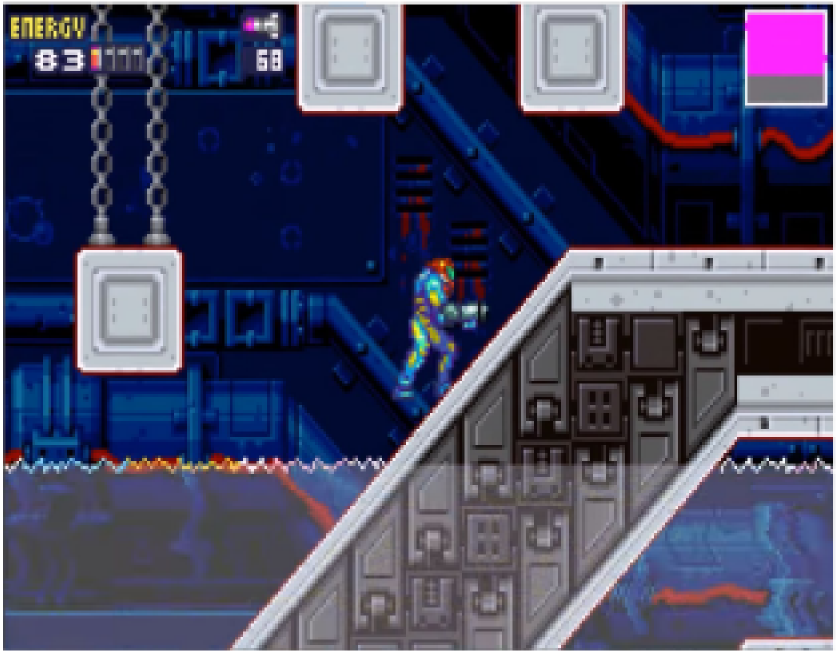

Figure 11: A platforming section in *Metroid Fusion* above the crackling electrified water below (shown by the jagged edge of the water)

Control Challenge in Execution

While game feel, including how "good" or "poor" game controls feel to a player (and thus, how easy or hard it is to perform actions in the game based on those controls), is an important part of game design (Swink 2008), a deep examination of the *Metroid* series' game feel is outside the scope of this study. Similarly, we will not address how, in some games players can re-map the controls potentially making things easier (or harder) for themselves, use alternate controllers, and more. Instead, our focus will be on control complexity: a higher number and complexity of specific actions to recall at any time will increase the difficulty of executing a task. At face value, *Metroid Dread* is more difficult than *Metroid* because there are more buttons to press (and remember what they do) to accomplish different actions (see Figures 12 and 13).

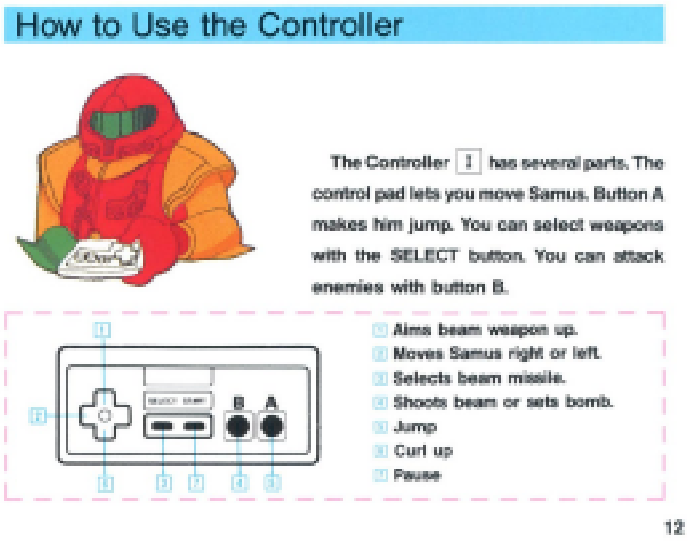

Figure 12: The controls page from *Metroid*'s game manual

Figure 13: The basic control scheme for *Metroid Dread* (before upgrades are obtained)

Furthermore, pressing two buttons at once, or two in quick succession is more challenging than pressing one button. *Metroid Fusion*'s strategy guide advises: "To shoot a Sciser that is above Samus, jump up, then fire as she comes down past the level of the target" (Cassady et al. 2001, 100), implicitly suggesting that actions requiring many buttons to be used in tandem and/or quickly after each other will be more difficult to perform. Button-combination actions are sometimes called techniques. *Metroid Fusion* has several including the "wall jump" and "shinespark". The wall jump requires the player synchronize a second jump press at the moment they collide with a wall, while executing a shinespark allows Samus to gather energy by running a long distance, crouching to store that power, then, by pressing a button and a direction, Samus forcefully launches in that direction. Segments that require use of these techniques will have a higher challenge.

Broadly, the Control aspects of **Execution** become more challenging over the Metroid series since, within each game, the player finds upgrades that provide additional abilities. *Metroid* features nine upgrades for the player to find, *Super Metroid* has 16 and *Metroid Dread* has 18. More abilities mean more to learn and master, and therefore a higher complexity in the controls portion of execution.

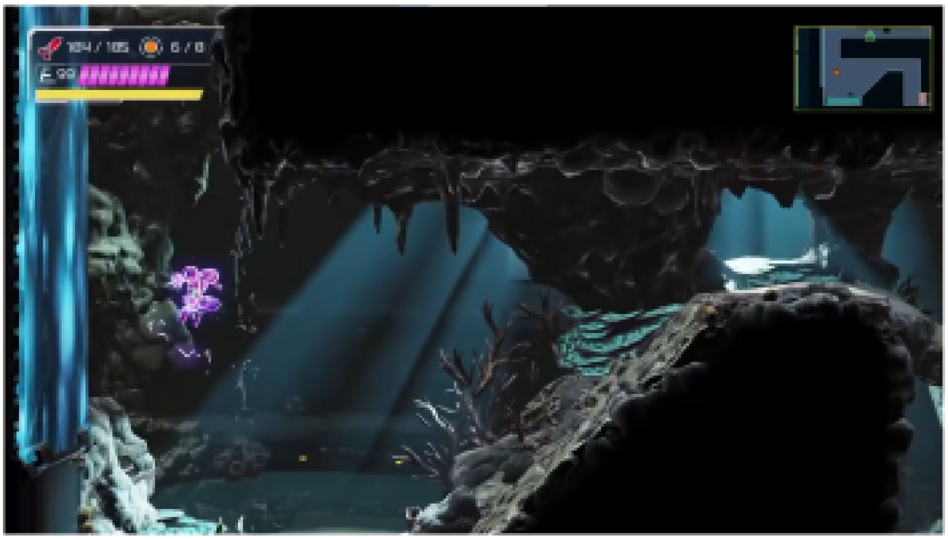

Figure 14: A room in *Metroid Dread* where the player must use "shinespark" several times in a row

CONCLUSIONS

We have observed that when it comes to the task of navigation, important design decisions are made in terms of which parts of the challenge of navigation to emphasize. Recall that navigation consists of determining a destination, planning a route to that destination, and then executing on that plan. Designers effectively regulate a games' difficulty by making these steps harder (or easier). Different Metroid titles emphasized different steps in navigation, i.e. a game emphasizing the challenge of **Execution** will not necessarily also emphasize the challenge of **Destination** or **Routing**. For example, *Metroid Fusion* and *Zero Mission* both feature a high amount of guidance for both destination finding and routing, with their directives and waypoints directly leading the player. This guidance is "offset" with demanding platforming and complicated controls that emphasizes **Execution**. *Metroid* for NES is the inverse, the game's simpler controls and basic movement de-emphasize **Execution** challenge. But, its lack of positive guidance emphasizes **Destination** and **Routing**. *Metroid II* is similar, with simple **Execution** and little guidance to assist in locating destinations. However, with its gameworld designed around a large shaft the player must repeatedly return to, **Routing** is simplified over its predecessor. *Metroid Dread* has a stronger focus on micro-planning than other games in the series, with its EMMI Zones acting as particularly high-intensity challenges for both **Routing** and **Execution**, while leaving destination finding manageable. Finally, *Super Metroid* emphasizes **Destination**, **Routing**, and **Execution**. It lacks direct guidance for locating destinations, frequently forces the player to manage large amounts of information for routing tasks, and has difficult platforming that requires the player to master the game's somewhat complicated controls.

Understanding the differences between the steps in navigation and the ways that difficulty can be regulated within each step can help in the design of games with notably different player experiences even within the same genre and series. If we think of a game as having a "difficulty quota" (a desired overall level of difficulty), a designer could "spend" parts of this quota across different steps: e.g. in this game, **Routing** and **Destination** will be a significant source of difficulty thus **Execution** will not. The same can apply within sections of a game – knowing that **Routing** within an area is easy allows the designer to consider how to provide difficulty via **Execution** and/or **Destination**. Of course, the highest difficulty in navigation would come from having all three "dialed up" as it were.

The challenge games provide is varied and while there is research into how challenge in games is composed, the exploration of individual tasks is less studied. We focused narrowly on the navigation challenge in the *Metroid* series, but it is easy to imagine performing similar analyses for other games and genres. Doing so could reveal insights and nuances in terms of how different games implement the challenge of navigation. The design of levels in *Super Mario Bros* might prioritize providing challenge via **Execution** rather than, say, **Routing** and do so in different ways than other games. Similarly, we did not examine navigation in 3D spaces or first-person perspective games including virtual reality. The challenge of **Routing** is significantly different for the player when they cannot see what is around a character in the same way you do in a 2D side-scrolling game. We look forward to expanding our initial framework to incorporate more types of games and are excited to also examine its usefulness in guiding the process of game design.

REFERENCES

Aponte, Maria-Virginia, Guillaume Levieux, and Stéphane Natkin. 2011. "Difficulty in Videogames: An Experimental Validation of a Formal Definition." In *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*. ACE '11. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2071423.2071484.

Arnott, Luke. 2017. "Mapping Metroid: Narrative, Space, and Other M." *Games and Culture* 12 (1): 3–27. https://doi.org/10.1177/1555412015580016.

Barker, Benjamin. 2019. "Use Shortcuts as Checkpoints to Give Players a Sense of Discovery with Each Death." In *Game Design Snacks: Easily Digestible Game Design Wisdom*, edited by J. P. Zagal, 37–38. Pittsburgh, PA, USA: Carnegie Mellon University: ETC Press. https://press.etc.cmu.edu/student/game-design-snacks.

Bowman, Nicholas David, David Wasserman, and Jaime Banks. 2018. "Development of the Video Game Demand Scale." In *Video Games*, edited by Nicholas David Bowman, 208–33. New York: Routledge. https://doi.org/10.4324/9781351235266-13.

Brandse, Michael, and Kiyoshi Tomimatsu. 2013. "Empirical Review of Challenge Design in Video Game Design." In *HCI International 2013 - Posters' Extended Abstracts*, edited by Constantine Stephanidis, 398–406. Berlin, Heidelberg: Springer Berlin Heidelberg.

Caldwell, McKenna. 2019. "Give Players a Traveling Companion to Help Introduce Them to Your Open World." In *Game Design Snacks: Easily Digestible Game Design Wisdom*, edited by J. P. Zagal, 39–40. Pittsburgh, PA, USA: Carnegie Mellon University: ETC Press. https://press.etc.cmu.edu/student/game-design-snacks.

Camper, Brett. 2008. "Retro Reflexivity: La-Mulana, an 8-Bit Period Piece." In *The Video Game Theory Reader 2*, edited by Bernard Perron and Mark J. P. Wolf, 191–218. New York and London: Routledge.

Cassady, David, Mark Cohen, and Tri Pham. 2001. *Metroid Prime / Metroid Fusion: Prima's Official Strategy Guide*. Roseville, CA: Prima Games.

Chen, Jui Lin, and Kay M Stanney. 1999. "A Theoretical Model of Wayfinding in Virtual Environments: Proposed Strategies for Navigational Aiding." *Presence* 8 (6): 671–85.

Cook, Michael, Simon Colton, and Alison Pease. 2012. "Aesthetic Considerations for Automated Platformer Design." In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 8:124–29.

Cuerdo, Marjorie Ann M., Anika Mahajan, Josh Mao, and Edward F. Melcer. 2023. "Try Again?: A Macro-Level Taxonomy of the Challenge and Failure Process in Games." In *2023 IEEE Conference on Games (CoG)*, 1–8. https://doi.org/10.1109/CoG57401.2023.10333199.

Cuerdo, Marjorie Ann M., and Edward F. Melcer. 2020. "'I'll Be Back': A Taxonomy of Death and Rebirth in Platformer Video Games." In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–13. CHI EA '20. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3334480.3382863.

Darken, Rudolph P, Terry Allard, and Lisa B Achille. 1999. "Spatial Orientation and Wayfinding in Large-Scale Virtual Spaces." *Presence: Teleoperators and Virtual Environments* 8 (6): 3–6.

Darken, Rudolph P., and John L. Sibert. 1996. "Wayfinding Strategies and Behaviors in Large Virtual Worlds." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 142–49. CHI '96. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/238386.238459.

Denisova, Alena, Paul Cairns, Christian Guckelsberger, and David Zendle. 2020. "Measuring Perceived Challenge in Digital Games: Development & Validation of the Challenge Originating from Recent Gameplay Interaction Scale (CORGIS)." *International Journal of Human-Computer Studies* 137:102383. https://doi.org/10.1016/j.ijhcs.2019.102383.

Engström, Johan, Gustav Markkula, Trent Victor, and Natasha Merat. 2017. "Effects of Cognitive Load on Driving Performance: The Cognitive Control Hypothesis." *Human Factors* 59 (5): 734–64. https://doi.org/10.1177/0018720817690639.

Fernandez-Vara, Clara. 2007. "Labyrinth and Maze." *Space Time Play*, 74–77.

Fernandez-Vara, Clara, Jose Zagal, and Michael Mateas. 2005. "Evolution of Spatial Configurations In Videogames." In *Changing Views: Worlds in Play, Selected Papers of DIGRA 2005*, edited by Suzanne de Castell and Jen Jenson, 167–75. Vancouver, Canada.

Flint, Alex, Alena Denisova, and Nick Bowman. 2023. "Comparing Measures of Perceived Challenge and Demand in Video Games: Exploring the Conceptual Dimensions of CORGIS and VGDS." In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. CHI '23. New York, NY, USA:

Association for Computing Machinery. https://doi.org/10.1145/3544548.3581409.
Fullerton, T., C. Swain, and Mark Hoffman. 2008. *Game Design Workshop: A Playcentric Approach to Creating Innovative Games, 2nd Edition*. Burlington, MA: Morgan Kaufman.
Gazzard, Alison. 2010. “Paths, Players, Places: Towards an Understanding of Mazes and Spaces in Videogames.” PhD Thesis.
Hoeg, Thomas. 2008. “The Invisible Hand: Using Level Design Elements to Manipulate Player Choice.” *Masters of Interactive Technology in Digital Game Development with a Specialization in Level Design, Guildhall at Southern Methodist University*.
Horbiński, Tymoteusz, and Krzysztof Zagata. 2022. “View of Cartography in Video Games: Literature Review and Examples of Specific Solutions.” *KN - Journal of Cartography and Geographic Information* 72 (2): 117–28. https://doi.org/10.1007/s42489-022-00104-8.
Iantorno, Michael. 2021. “See You Next Mission: An Analysis of the Super Metroid VARIA Randomizer.” *ROMChip: A Journal of Game Histories* 3 (2). https://romchip.org/index.php/romchip-journal/article/view/116.
Liszio, Stefan, and Maic Masuch. 2016. “Lost in Open Worlds: Design Patterns for Player Navigation in Virtual Reality Games.” In *Proceedings of the 13th International Conference on Advances in Computer Entertainment Technology*. ACE ’16. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3001773.3001794.
Mawhorter, Peter, Indira Ruslanova, and Ross Mawhorter. 2022. “Representing Exploration in Metroidvania Games.” In *Proceedings of the FDG Workshop on Procedural Content Generation*. Vol. 3.
Merabet, Lotfi B., Erin C. Connors, Mark A. Halko, and Jaime Sanchez. 2012. “Teaching the Blind to Find Their Way by Playing Video Games.” *PLoS ONE* 7 (9): e44958. http://dx.doi.org/10.1371/journal.pone.0044958.
Moura, Dinara, and Magy Seif El-Nasr. 2015. “Design Techniques for Planning Navigational Systems in 3-D Video Games.” *Comput. Entertain.* 12 (2). https://doi.org/10.1145/2701657.2633421.
Neuschwander, Lee. 2019. “Let Experienced Players Turn Off Breadcrumbs.” In *Game Design Snacks: Easily Digestible Game Design Wisdom*, edited by J. P. Zagal, 113–14. Pittsburgh, PA, USA: Carnegie Mellon University: ETC Press. https://press.etc.cmu.edu/student/game-design-snacks.
Nintendo. 1986. *Metroid [Nintendo Entertainment System]*. Nintendo.
———. 2021. *Metroid Dread [Nintendo Switch]*. Nintendo.
Nintendo R&D1. 1991. *Metroid II: Return of Samus [Game Boy]*. Nintendo.
Nitsche, Michael. 2008. *Video Game Spaces: Image, Play, and Structure in 3D Worlds*. The MIT Press. https://doi.org/10.7551/mitpress/9780262141017.001.0001.
Pelland, Scott, Leslie Swan, and Jeff Bafus, eds. 1994. *Super Metroid Player’s Guide*. Redmond, WA: Nintendo.
Pusey, Megan, Kok Wai Wong, and Natasha Anne Rappa. 2021. “The Puzzle Challenge Analysis Tool. A Tool for Analysing the Cognitive Challenge Level of Puzzles in Video Games.” *Proc. ACM Hum.-Comput. Interact.* 5 (CHI PLAY). https://doi.org/10.1145/3474703.
Retro Studios. 2002. *Metroid Prime [Gamecube]*. Nintendo.
———. 2023. *Metroid Prime Remastered [Nintendo Switch]*. Nintendo.

Rollings, Andrew, and Ernest Adams. 2003. *Andrew Rollings and Ernest Adams on Game Design*. Indianapolis, IN: New Riders Publishing.

Schell, Jesse. 2008. *The Art of Game Design: A Book of Lenses*. Burlington MA: Morgan Kaufmann.

Smith, Gillian, Mee Cha, and Jim Whitehead. 2008. “A Framework for Analysis of 2D Platformer Levels.” In *Proceedings of the 2008 ACM SIGGRAPH Symposium on Video Games*, 75–80. Sandbox ’08. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/1401843.1401858.

Suits, Bernard. 1978. *The Grasshopper: Games, Life and Utopia*. Toronto Buffalo: University of Toronto Press.

Swink, Steve. 2008. *Game Feel: A Game Designer’s Guide to Virtual Sensation*. CRC Press.

Toups Dugas, Phoebe O., Nicolas Lalone, Sultan A. Alharthi, Hitesh Nidhi Sharma, and Andrew M. Webb. 2019. “Making Maps Available for Play: Analyzing the Design of Game Cartography Interfaces.” *ACM Trans. Comput.-Hum. Interact.* 26 (5). https://doi.org/10.1145/3336144.

Vahlo, Jukka, and Veli-Matti Karhulahti. 2020. “Challenge Types in Gaming Validation of Video Game Challenge Inventory (CHA).” *International Journal of Human-Computer Studies* 143:102473. https://doi.org/10.1016/j.ijhcs.2020.102473.

Wehbe, Rina R., Elisa D. Mekler, Mike Schaekermann, Edward Lank, and Lennart E. Nacke. 2017. “Testing Incremental Difficulty Design in Platformer Games.” In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 5109–13. CHI ’17. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3025453.3025697.