

# Level Design Patterns in 2D Games

Ahmed Khalifa  
*Game Innovation Lab*  
New York University  
New York, USA  
ahmed@akhalifa.com

Fernando de Mesentier Silva  
*Independent Researcher*  
Rio de Janeiro  
Rio de Janeiro, Brazil  
fms2005@gmail.com

Julian Togelius  
*Game Innovation Lab*  
New York University  
New York, USA  
julian@togelius.com

**Abstract**—Videogame designers use tips and tricks and tools of the trade to design levels. Some of these tips are based on their gut feeling and others have been known in the game industry for the last 30 years. In this work, we discuss six of common level design patterns present in 2D videogames. The patterns under discussion are the product of an exploratory analysis of over thirty 2D games. We choose to focus on patterns that are both common and impactful for the overall player experience. We discuss in detail the rationale for and advantages of each pattern, showing examples of games that make use of such. We conclude with a discussion of the usage and understanding of these patterns from the perspective of level design and how other technical approaches can benefit from them.

**Index Terms**—Design Patterns, Level Design, Analysis

## I. INTRODUCTION

Many games depend on content variety to tailor their gameplay experience. Having repetitive challenges can lead to player boredom, so designers generally aim to create an experience that is constantly engaging. Games usually consist of many facets such as levels, mechanics, graphics, sounds, etc. One of the most unique is level design. It does not introduce a new dimension in the game, instead it is concerned with the arrangement of the existing elements. With the high demand of content in games, level design becomes a major task in game design. Its role is of pivotal importance, to the extent of being assigned to a different team than the one responsible for game and mechanics design, for games with large scopes.

When creating levels, innovation and variation alone are not enough to provide quality player experience. Having frustrating scenarios, unclear directions or unappealing challenges are reasons that could lead a level to have negative impact on players. When observing levels from existing games, it is possible to highlight common patterns that play a part in improving the quality of the design.

These patterns are simple conventions, meant to describe common practices used in level design. Level design is a subconcept of game design. As such, so are level design pattern a subset of game design patterns. However, level design presents many intricacies that standout inside the scope of game design. The major role levels plays in the experience of the game is reflected by the amount of resources put into this facet of the design during production. Levels can highlight

the story, create smaller short-term achievable goals and help shape the player experience, with the objective of enhancing player immersion. Level creation plays such an important part in a game that sometimes developers provide their audience with level design tools, so they can create and share new levels.

Although the level design patterns we present might be well known to game designers, it is important to formally document them. By doing such, we can facilitate communication between experienced and novice designers, ultimately contributing to the growth and sharing of level design practices. For this purpose, this work aims at formalizing the concept of level design patterns, providing not only valuable resource to novice designers but also groundwork for future study and design of such patterns. It is also our understanding that this work could be beneficial to the research community, especially in the fields of game design, game analytics, procedural content generation and automated level design.

In this paper, we describe in detail six level design patterns. These patterns are noticeable across a variety of 2D games. We discuss the principles and motivation behind applying these patterns and show examples of it being used across multiple games. This work does not showcase all potential or even existing level design patterns, choosing instead to focus on some of the most common, with the intent of formalizing the concept and laying the foundation for future research and discussion to extend upon.

## II. BACKGROUND

Design patterns date back to the seventies, with origins in the work by architect Christopher Alexander [1]. Alexander created these patterns to find a more formal language to communicate with. The design patterns summarized recurrent problems that could be found in the environment and their respective solutions. The solutions provided worked as guidelines, and not as definitive answers, to the problems shown. This idea was transferred to software engineering in the nineties, when Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides published a book discussing software design patterns [2]. Similar to Alexander, they summarized recurrent problems in object oriented design and suggested respective solutions to them. Kreimeier [3] introduced game design patterns in a similar manner to software design patterns as a way to solve recurrent problems. This definition has changed to be less strict so it views the game design patterns

as a common reoccurring game design elements that affect gameplay [4], [5].

Developers, designers, academics, game makers and critics have been discussing design patterns for a long time. Often they are not referred to as design patterns, but as tips, tricks, guidelines, tools of the trade, etc. They usually refer to design pattern concepts by analyzing a certain level or area [6], [7], [7]–[9], game [10]–[22], genre [23]–[31], or being as general as possible [3]–[5], [32]–[42]. That information is usually based on their experiences from designing, developing and playing games. Most notable are patterns that are recurrent across multiple games. We list similarities between previous attempts at cataloguing and classifying level design patterns and our own work in the upcoming sections whenever relevant.

In addition to finding and taxonomizing design patterns, studies have investigated how to use such patterns in procedural content generation. Thus, design patterns have been used for level generation [19], [25], [38], [43]–[48], but also in attempts to generate game rules [49], [50]. A common approach is to use design patterns as constraints during generation to limit the generative space to only include levels that make use of them [38], [43], [44], [48], [49]; this is most often seen in constructive generators [51]. Another approach is to use design patterns in fitness functions for search-based generators [52], to guide the search towards levels that utilize these patterns [25], [38], [45], [46], [50], and yet another to include patterns in a training set for a PCGML algorithm [53] to help in generating new content [19], [47].

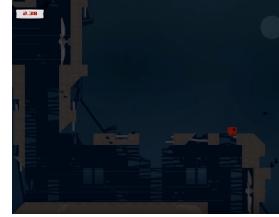
### III. PATTERNS

The patterns we discuss are elements that are present in levels across multiple games, rather than being a feature that is recurrent on the same title. To this point, we choose to showcase these with screenshots from different games. We believe that having examples from different sources supports our point that these are valuable concepts with longevity.

Patterns were chosen taking into account their relevancy (if it is often discussed in the community or other works), their recurrence (if they were observed in a substantial amount of the games that were played) and their impact (how important is the pattern in the games that have it). Over 30 Games were chosen to form a representative set, mixing older titles and new ones; published from large companies to independent developers; different gameplaying experiences (such as platforming, racing games, combat-based, puzzle-based, etc); while prioritizing games that are relevant, either from being popular, having a strong community (or cult following) or being commonly referenced as a positive example.

Before presenting the patterns we observed, we identify the vocabulary we are using throughout the paper. Such vocabulary is listed in table I. This vocabulary is extracted from Anthropy and Clark [35], Bjork and Holopainen [4] and design pattern wiki [5]. It was then adapted to fit our work.

In this work, the patterns presented are the result of the position of game elements in the level in relation to each other. The reason we do not list checkpoints as a pattern, for instance,



(a) Hallway's shape guides players on where to go.



(b) Stacked platforms guide the players upwards.

Fig. 1: The level shape used in *Super Meat Boy* and *Super Mario World* to guide players to specific directions.



(a) The bananas guide players to dive into what looks like a bottomless pit ...



(b) ... only to find the platform that has the target they have to hit to finish the level.

Fig. 2: Collectibles used in *Donkey Kong Country 2* to guide players to what looked like a dangerous path.

is that we frame it as a mechanic in the game. Nonetheless, the checkpoint mechanic can be used in conjunction with the patterns presented here. For instance, checkpoints are usually present in Safe Zones and are a commonly associated with Pace Breaking.

While the patterns are presented individually, they can be combined or modified to provide a new and fresh experience. It is recurrent that new design patterns can be generated from the combination or variation of existing patterns. For example, Thompson [30] showed that a pattern from *New Super Mario Bros U* [61] is a combination of multiple known patterns. It is also worth noting that the absence of a pattern does not represent bad design. These are guidelines and ultimately it is the designer's decision to make use of such concepts.

Not all patterns are applicable to all games or levels, so we choose to focus on those we believe to be the most common across different games. In this work, we present six patterns: Guidance, Foreshadowing, Safe Zone, Layering, Branching and Pace Breaking. In the following subsections we discuss each individually, and present examples of how games have made use of them, by presenting screenshots as visual evidence whenever possible or necessary.

#### A. Guidance

When playing a level, it is possible for players to lose track of the path they need to take, especially when exploration is an aspect of the design. The pattern we name Guidance refers to the use of non-verbal game elements to guide players in an intended direction. We rather classify verbal guidance and tutorials (verbal or not), such as a non-playable character giving instructions or tips, or a character showing you an

Name	Definition
Avatar	the game entity the players control inside the game. For instance, Mario and Luigi in <i>Super Mario Bros</i> [54].
Boss	an enemy that poses a bigger threat than most. An example is Iceman from <i>Megaman</i> [55].
Collectible	a game object that can be collected by the players. An example are the coins in <i>Super Mario Bros</i> [54].
Enemy	a hazard that takes the persona of a character. Goombas from <i>Super Mario Bros</i> [54] are one example of enemies.
Environment Tile	entities used to decorate the game, usually not interactable. Clouds in <i>Super Mario Bros</i> [54].
Hazard	game entities that hinder the player from completing their goal. An example are spikes in <i>1001 Spikes</i> [56].
Inaccessible Area	an area that players cannot reach, but later might become accessible. For instance, dungeon rooms in <i>The Legend of Zelda</i> [57] that require a key to enter.
Level	section of the game, where players act towards winning the game. An example is World 1-1 from <i>Super Mario Bros</i> [54].
Mechanic	an element in the game, visible or not, that enforces a rule. An example are the checkpoints in <i>Sonic the Hedgehog</i> [58].
Object	any entity that appears in a game scene and can change state. It includes hazards, enemies, power-ups and etc.
Power-up	a collectible that positively impacts the players. The mushroom in <i>Super Mario Bros</i> [54] is an example of power-up.
Projectiles	moving entities projected from a hazard or avatar. Hammers thrown in <i>Super Mario Bros</i> [54] are an example.
Rule	describes how the game is played and controlled. An example is the rule to collect all pellets to win in <i>Pac-Man</i> [59].
Scene	a slice of a level/world that revolves around a concept, usually a challenge.
Screen	the part of the game level/world that is currently visible to the players.
Solid Tile	a game object that prevents the players from passing through it. An example is the floor in <i>Super Mario Bros</i> [54].
World	a large game area that is built from multiple levels. For example, Brinstar in <i>Super Metroid</i> [60].

TABLE I: Game design vocabulary used in this work. Concepts are presented in alphabetical order.



(a) *Super Metroid* uses enemies to highlight the path the game wants them to follow.



(b) *Castlevania Legends* uses the enemies to guide the player towards a trap.

Fig. 3: An example of how *Super Metroid* and *Castlevania Legends* used enemies to guide players to specific areas.

action you can replicate, as a game design element of teaching or narrative purpose. This pattern encompasses the concept of guiding players both towards making direct progress in the level, as well as in the direction of a secret or collectible.

The easiest way to guide the player is through the level shape. In this pattern, designers use solid tiles to guide the player eyes towards the target direction. This pattern has been discussed more often in 3D games [31], [32], [42], [62] than 2D ones [17], [36]. The main reason is in 3D games players usually have more freedom of movement which require the designers to have to push them towards the next location. Figure 1 shows how *Super Meat Boy* [63] and *Super Mario World* [64] use solid tiles to convey the direction the player need to traverse the level.

One of the most common representation of Guidance is done through the use of collectibles. It can indicate the main path, as well as provide awareness to areas players are not able to see yet. Developers can use this technique to guide players away from “bad decisions” such as performing a blind jump, because they cannot perceive the correct path through another location. Collectibles can also be used to guide players into exploring a path that initially looks “dangerous” or incorrect, or to steer them into a safe location or target area when they cannot see ahead of them. An example can be seen in Figure 2. The figure shows two follow-up screens in *Donkey Kong Country 2* [65]. At the end of the second level of



(a) The cracked tile catches the attention of the player for its different texture.



(b) The cracked wall indicates a passage that opens when bombed.

Fig. 4: An example of how *Super Metroid* and *The Legend of Zelda: A Link to the Past*, respectively, use different textured tiles to have players searching for hidden paths.

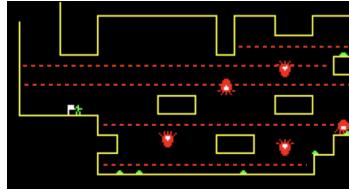
the game, if players dive where the bananas are located, as on figure 2a, in what looks like a bottomless pit, they will be greeted with the platform below, shown in figure 2b. This type of guidance has been mentioned in previous work by Anthropy [36], Smith [24], Pugh [32], Koncewicz [23], and Kremers [40]. The concept is referenced under multiple names, such as breadcrumbing [32] and signifiers [36], but the meaning remains the same.

Enemies can induce Guidance in a different fashion than most hazards. Enemies can be used to attract players attention, as they intend to fight them. This type of guidance has been discussed in the Boss Key Series [15] during the analysis of *Super Metroid* [60], it was discussed in the work Milam and El Nasr [42] during analysis of pursue AI pattern in 3D games, and in Hoeg’s work [22] during discussing the effect of movement on player choices in FPS games. The presence of enemies guides players towards their location. For instance, enemies in seemingly inaccessible locations cause the players to wonder how to reach such. Figure 3a shows how *Super Metroid* [60] uses enemies to guide players into the intended path. Figure 3b shows enemies being used to attract players to a trap (where the floor collapses), rather than a rewarding path, in *Castlevania Legends* [66].

Another common representation of Guidance is using environmental cues such as highlighted tiles (changing the color

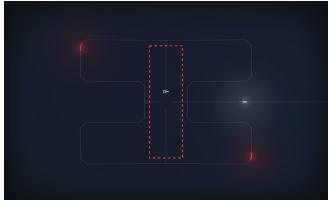


(a) *Mega Man*



(b) *Star Guard*

Fig. 5: Safe Zones as shown by the spaces where the enemy lines of fire (which are invisible in-game), represented by the red dotted lines, do not cross.



(a) *Linelight*



(b) *The Legend of Zelda*

Fig. 6: Safe Zones being highlighted by red dotted rectangles.

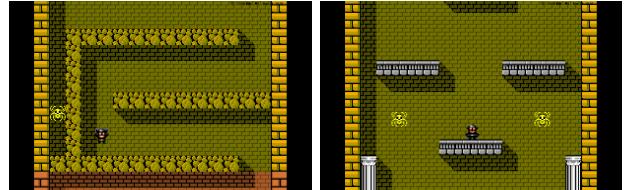
or texture of certain tiles in the scene instigates player curiosity). This technique has been discussed in several previous work [3], [10], [15], [18], [31], [41], [42], [62]. Authors discuss how the use of different environment tiles can draw players' focus to certain areas, instigating them to explore. Figures 4a and 4b show how different textured tiles are used in *Super Metroid* [60] and *The Legend of Zelda: A Link to the Past* [67], respectively, to evoke player exploration.

### B. Safe Zone

Safe Zone is the pattern that refers to one or more areas in the current scene where the players are not exposed to negative interactions (hazards, enemies, and etc). Having Safe Zones creates pockets of space where players can analyze the surroundings and safely plan their next actions. They are especially important at the starting point (spawn point) for the avatar, as not to punish players before they can become aware of the challenge ahead of them. Safe Zones are rarely effectively drawn on the screen, but rather being identified after observing the scene for a short amount of time to notice hazard reach and patterns.

In figure 5 we draw, with red dotted lines, the line of fire of the enemies present on the screen, which are invisible in the actual game. In these examples, the Safe Zone is a space where the avatar can stand in without intersecting any line of fire. In the scene from *Mega Man* [55] shown in figure 5a we can notice Safe Zones on the bottom most platforms, right above the leftmost ladder and at the top-right of the screen. In the scene from *Star Guard* [68] we can notice the Safe Zones as the empty black spaces. In these scenarios, it is common for players to try and devise strategies of how to move between Safe Zones while proceeding closer to their goal.

In figure 6 we draw dotted red rectangles to highlight Safe Zones in the scene. The level from *Linelight* [69] shown in figure 6a has a big center area where the moving hazards



(a) The Spider hazard gets introduced.



(b) Later, it is part of a more challenging obstacle.

Fig. 7: *Mighty Jill Off* foreshadowing a hazard to prepare the players for a later challenge in the game.



(a) An enemy shoots scissors during the stage.



(b) The boss of the stage has the same scissor projectile.

Fig. 8: *Mega Man* foreshadows the boss's weapon during the stage leading up to it.

cannot reach the player. It is important for players to use the Safe Zone to maneuver the scene, as hazards move faster than their avatar. Figure 6b shows that doorways of dungeon rooms in *The Legend of Zelda* [57] are Safe Zones that allow players to become familiar with the challenge they are about to face.

Safe Zones can provide the opportunity for Pace Breaking, a pattern we describe later in this paper. They can also be used to express Guidance, guiding the player to favor certain paths over others. The same concept of Safe Zone has appeared in previous work [8], [24], [26], [34], but has usually been used to host checkpoints and save rooms [4], [5], [14], [36].

### C. Foreshadowing

Foreshadowing is a literary concept [70]–[72] where the author hints at something that will happen later in the story. The general concept can be applied to games [6], [10], [14], [15], [41]. We then call Foreshadowing the pattern of introducing the player to an element under a controlled environment, before this element is more integral to the game.

Foreshadowing is a helpful pattern to help players experience a new game mechanic or game object without an explicit tutorial. Once players learned the basics, the game can present a challenging situation around this concept. Figure 7 shows how *Mighty Jill Off* [73] introduces a new hazard. The spider is first introduced in the scene shown at figure 7a, and becomes more threatening later in the game as shown in figure 7b. Figure 8 shows how a stage hazard in *Mega Man* [55] foreshadows the boss fight in that level. During the level, players are confronted by the scissor shaped projectiles shown in figure 8a which foreshadows the stage's boss's weapon, as shown in figure 8b.

Foreshadowing can also instigate players curiosity, making them excited about future possibilities. This is a common feature in games with an exploratory nature. Figure 9 shows two



(a) *The Legend of Zelda: A Link to the Past* shows the players a stone than can be lifted later in the game.



(b) *Super Metroid* foreshadows a path that requires the Morph Ball power-up.

Fig. 9: *The Legend of Zelda: A Link to the Past* and *Super Metroid* foreshadow obstacles that can only be surpassed with power-ups available later in the game.



(a) *Lesbian Spider-Queens Of Mars*



(b) *VVVVVV*

Fig. 10: Layering the same object increases the difficulty without introducing a new object or mechanic.

examples of such. *The Legend of Zelda: A Link to the Past* [67] introduces objects that the player can only interact with later in the game, as shown in figure 9a. *Super Metroid* [60] teases paths only accessible after acquiring a power-up present later in the game. This application of Foreshadow tends to reflect a design that gives players incentives to backtrack, which is a common action associated with exploration.

#### D. Layering

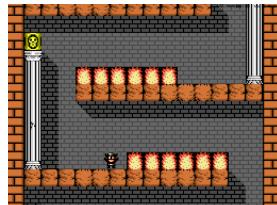
Layering is the pattern of combining multiple objects to create a new experience. It often relies on Foreshadowing to be able to present a fair challenge to the player. The most common use of Layering is to introduce new and harder challenges, without the need to present new elements to the player. It benefits from previously introducing the different elements separately, which players are familiar with their behaviors, but have to devise a new strategy to overcome them. This pattern can lead to frustrating experiences when the outcome of mixing these elements is not planned through. Many games use this pattern efficiently [9], [11], [13], [17], [20], [28], [35], so as not to have game objects that seldom appear and are deemed unimportant by the player (which is similar to Thompson’s Throw away pattern [12]).

Figure 10 shows how games can create harder experiences by combining multiple instances of the same enemy or hazard. While *Lesbian Spider-Queens of Mars* [74] achieves such by using multiple of the same enemy, *VVVVVV* [75] does so by placing several copies of a moving hazard in a specific layout.

Figure 11 shows games that create a whole new experience by combining different elements. In *Mega Man* [55] shown on



(a) *Mega Man*



(b) *Mighty Jill Off*

Fig. 11: Introducing new challenges by Layering elements. Using previously introduced elements to create new experiences.



(a) *L'Abbaye Des Morts*



(b) *The Legend Of Zelda*

Fig. 12: Branching with no restrictions. Players can freely choose which path they want to take at any point.

figure 11a the players have to wait to jump on the disappearing platforms when they become solid, but also have to pay attention to the enemy that runs from side to side on the floor they are standing. While in *Mighty Jill Off* [73] players need to crawl up the level as fast as possible without touching the flames and before the spider comes out of the yellow box at the top left of the screen to follow the player.

#### E. Branching

Branching refers to providing the players with multiple paths to reach their objective. Giving players a choice leads to a feeling of empowerment. As previous work before [4], [5], [7], [9], [12], [14], [15], [20], [24]–[28], [30], [34], [37], [39], we deduced that there are various ways in which this pattern has been leveraged to produce different experiences. Sometimes the game provides a false sense of choice [21], when to complete the level exploration of most paths is required, as is common with dungeons in *The Legend of Zelda* [57]. In *Mega Man* [55] levels were designed to be played in any order, as the players could choose to visit stages in any order. Therefore, every level had to be designed to be beatable with only the tools players are provided with at the beginning. *Super Mario Bros* [54] is famous for having pipes that players can enter to find an alternative paths through the level.

Levels can have Branching with no restrictions, meaning that all paths are open for players to select from. This reinforces the feeling of exploration during gameplay. Figure 12 shows two scenes that use Branching with no restrictions. In *L'Abbaye Des Morts* [76] shown in figure 12a and *The Legend of Zelda* [57] shown in figure 12b, the players are allowed to freely choose which path to take.

Conditional Branching presents the players with multiple paths, but only allows them to access certain paths after



(a) *Super Metroid* requires that players find power-ups to open certain doors.



(b) In *Super Mario Land*, taking the power-up makes specific paths inaccessible.

Fig. 13: Conditional Branching where players depend on having achieved specific conditions to take certain paths.



(a) *Excitebike*



(b) *Speedrunners*

Fig. 14: Skill-reward Branching. Players can choose a safer path with a small reward or a path that requires more skill, but offers bigger rewards.

meeting specific criteria. This form of Branching stimulates player curiosity and might require backtracking or repeated attempts to explore more than 1 path. Figure 13 shows 2 examples of Conditional Branching. As shown in figure 13a *Super Metroid* [60] has doors scattered throughout levels that are locked until players find the weapon that can open them. In the case of *Super Mario Land* [77], the avatar can be in 1 of 2 states, small or big. Figure 13b shows a level in *Super Mario Land* where players can only access one of the paths if their character is small. This leads players to interesting decisions, as being small makes the player more fragile, but only then can they reach certain paths [7].

Lastly, Branching can be used to create a risk-reward scenario [4], [5], [12], [14], [26], [30], [34], [37]. When presenting players with multiple paths, one can be a safer alternative with a small reward, while another require higher skill, but has a bigger payoff. This gives incentive for players to spend more time with the game, as they are rewarded for their investment. Figure 14 shows the racing games *Excitebike* [78] and *SpeedRunners* [79] applying risk-reward Branching. In both games players can chose to take a path the requires them to be more skilled at the game, but if successful puts them ahead of the competition.

#### F. Pace Breaking

Pace breaking refers to purposely changing the dramatic arc of the game from one scene to the next. It is commonly used to either increase or decrease tension, in order to make players more invested in the overall experience. Audio and visual cues are some of the most powerful tools to convey Pace Breaking to the players. Layering is one pattern commonly coupled with Pace Breaking as the presence of multiple elements tends to

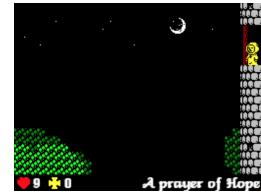


(a) Fighting with multiple enemies occurs regularly...

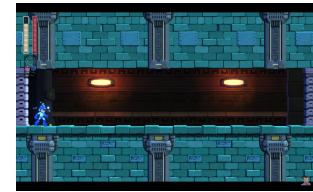


(b) ... until reaching a much stronger foe at the very end.

Fig. 15: *Final Fight* uses boss fights to create tension.



(a) *L'Abbaye Des Morts* reduces the stress just before the climatic challenges.



(b) *Mega Man 11* reduces the stress just before the climatic fight of the level.

Fig. 16: *L'Abbaye Des Morts* and *Mega Man 11* use empty areas to relax and break tension from players.

generate natural tension, especially when creating a harder challenge [24], [25]. Safe Zones, on the other hand, can be used to have the opposite effect and break the pace to reduce the tension [24], [25].

A common feature that creates Pace Breaking is the introduction of a hazard of conspicuous difficulty. Whether it is considered a boss fight or not, recognizing that they are in the presence of a bigger threat usually draws players attention, increasing the tension. Figure 15 shows the difference between normal gameplay and boss fights in *Final Fight* [80]. In the transition to the boss fight scene, the game changes the scenario and background music, it then reveals an enemy bigger in size, wielding a weapon and with a different health bar, all which indicate this to be a new, harder challenge.

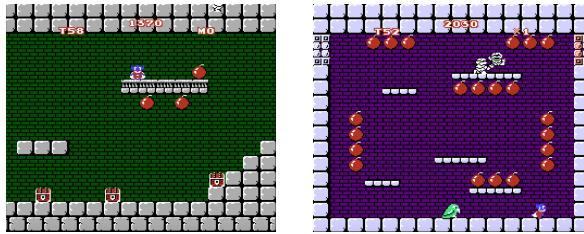
Decreasing the pace of the game can lead to interesting results. It can be used to give players time to relax and enjoy other aspects of the game, such as new environments or abilities. In other times, it is interesting to give players a moment of peace right before an intense scene to generate a climax (the calm before the storm) [35]. Figure 16 shows two examples of Pace Breaking to reduce players tension. In *L'Abbaye Des Morts* [76] shown in 16a, players are given time to relax after a sequence of stressful challenges and enjoy scenery of the night sky. In turn, *Mega Man 11* [81] introduces a completely empty room right before the climax of a stage, the boss fight, as shown in 16b. This creates a calm sequence right before the height of the dramatic arc for the level.

Pace Breaking can be a powerful tool to introduce players to a new experience, feeling or mechanic in the game [36]. In *Super Metroid* [60] when players acquire a new power-up, it is common for them to have to make use of such in a very simple scenario to proceed. These usually come right after a climatic moment in the game, as the power-ups in *Super Metroid* are usually given as rewards for beating hard



(a) The most common challenge is capturing the enemies before they can reach you...  
(b) ...but some levels change the pace by having to collect all static enemies, before they become a threat.

Fig. 17: *Lesbian Spider-Queens of Mars* uses Pace Breaking to present the player with a new challenge to overcome.



(a) Spacious corridors provide freedom of movement.  
(b) Tight spaces make hazards much harder to avoid.

Fig. 18: *Mighty Bomb Jack* alternates between sections.

challenges. Figure 17 shows a similar approach in *Lesbian Spider-Queens of Mars* [74]. In figure 17a we can see a regular level from the game, where the players have to use their laser to capture the other characters. Meanwhile in figure 17b there is the introduction of a level where the players are trying to reach the other characters before they break free and become stronger enemies. This change of pace breaks the normal gameplay of the game, that requires reaction and dexterity, to introduce a level with a run against the clock (timer) challenge.

Another example of Pace Breaking are levels designed to have choke points [4], [5], [28], arenas [29], or collision points [26]. These take players from a space where they have more freedom to move and escape hazards to a tighter space where interaction with hazards is usually unavoidable. Figure 18 shows this concept in *Mighty Bomb Jack* [82]. On the longer section of levels, as shown in 18a players control their avatar in a long corridor collecting power-ups and collectibles, while avoiding hazards. Meanwhile on the bonus stage, as in 18b, players are trapped in a small room where hazards are much harder to avoid while having to collect all bombs to proceed.

#### IV. DISCUSSION AND CONCLUSION

In this paper, we introduced 6 level design patterns extracted from observing multiple 2D games. These are meant as conventions that can improve player experience and not requirements to create a quality game. The formalization of these patterns contributes to create an universal language for developers, helping knowledge share and expansion.

The patterns are presented in this work individually, but it is common for them to be used in conjunction. Layering multiple

hazards can create tension, resulting in Pace Breaking. Locked doors can foreshadow future branches to pursue. It is important to experiment around these patterns and not be restricted by them, as the solution to a design problem might require such.

Level design patterns are generally used to have a positive impact on the gameplay experience. But it is worth noting that consciously subverting such concepts can provide an unique experience. For instance, *Syobon Action* [83] and *I Wanna Be The Guy* [84] use the element of surprise in their level design. By presenting features common to famous 2D platform games, players are caught off guard with their unexpected behaviour, such as having deadly traps where power-ups would usually be and having enemies spawn on top of the player character with no prior warning.

The level design patterns show in this work are not restricted to 2D games. They were presented as the result of studying 2D games, but all are applicable to 3D games as well. The definitions are not restricted to 2D space, changing the context does not invalidate their application. Zombie hordes on *Left 4 Dead* [85] are a result of Layering and create tension with Pace Breaking. The concept of open world games, such as *Red Dead Redemption* [86], are an extrapolation of the Branching pattern. *Super Mario 64* [87] constantly uses coins to guide players and foreshadow to instigate player curiosity.

The patterns in this paper can have a more technical application as well. They can be used as guidelines to procedural level generators, as part of fitness functions for search based generator, as training sets for machine learning algorithms or to guide towards having intentional level generators.

Ultimately, level design patterns have always been used in games. Having guidelines on how to improve player experience is a very valuable tool. A formal taxonomy facilitates wider dissemination of these concepts, stimulating expansion and contribution, helping improve level design practices.

#### ACKNOWLEDGMENTS

Ahmed Khalifa acknowledges the financial support from NSF grant (Award number 1717324 - "RI: Small: General Intelligence through Algorithm Invention and Selection").

#### REFERENCES

- [1] C. Alexander, *A pattern language: towns, buildings, construction*. Oxford university press, 1977.
- [2] E. Gamma, *Design patterns: elements of reusable object-oriented software*. Pearson Education India, 1995.
- [3] B. Kreimeier, "The case for game design patterns," [https://www.gamasutra.com/view/feature/132649/the\\_case\\_for\\_game\\_design\\_patterns.php?print=1](https://www.gamasutra.com/view/feature/132649/the_case_for_game_design_patterns.php?print=1), 2002, last Accessed: May 23, 2019.
- [4] S. Bjork and J. Holopainen, *Patterns in game design (game development series)*. Charles River Media, 2004.
- [5] S. Bjork, "Gameplay design patterns collection," [http://virt10.itu.chalmers.se/index.php/Main\\_Page](http://virt10.itu.chalmers.se/index.php/Main_Page), 2009, last Accessed: May 30, 2019.
- [6] A. Anthropy, "Level design lesson: To the right, hold on tight," <http://auntiepixelante.com/?p=465>, 2009, last Accessed: March 5, 2018.
- [7] ———, "Level design lesson: In the pyramid," <http://auntiepixelante.com/?p=459>, 2009, last Accessed: March 5, 2018.
- [8] ———, "Star guard," <http://auntiepixelante.com/?p=499>, 2009, last Accessed: March 5, 2018.
- [9] ———, "Level design lesson: Low overhead," <http://www.auntiepixelante.com/?p=1022>, 2011, last Accessed: March 5, 2018.

- [10] J. Penner, "Breaking the law of miyamoto," *The Gamer's Quarter*, no. 7, 2006.
- [11] A. Anthropy, "Knytt syndromes," <http://knytlevels.com/levels/dessgega-KnyttSyndromes.knytt.bin>, 2011, last Accessed: Jan 4, 2019.
- [12] T. Thompson, "Researching super mario bros. level design," [https://youtu.be/t\\_zXpKlcRE](https://youtu.be/t_zXpKlcRE), 2015, last Accessed: Jan 4, 2019.
- [13] M. Brown, "Super mario 3d world's 4 step level design," <https://youtu.be/dBmIkEvEBtA>, 2015, last Accessed: Jan 4, 2019.
- [14] G. Soup, "Cave story's masterful game design," [https://youtu.be/c\\_0sjTBzCW0](https://youtu.be/c_0sjTBzCW0), 2015, last Accessed: Jan 4, 2019.
- [15] M. Brown, "Boss keys series," [https://www.youtube.com/playlist?list=PLc38fcMFcV\\_uI4D6OChdWhsNsYY3NA5B2](https://www.youtube.com/playlist?list=PLc38fcMFcV_uI4D6OChdWhsNsYY3NA5B2), 2018, last Accessed: Jan 4, 2019.
- [16] M. Thorson, "Level design workshop: Designing celeste," <https://youtu.be/4RlpMhBKNr0>, 2018, last Accessed: Jan 4, 2019.
- [17] M. Brown, "How mega man 11's levels do more with less," <https://youtu.be/nYxHMZX6lN8>, 2018, last Accessed: Jan 4, 2019.
- [18] G. Soup, "Crosscode's brilliant level design: Analysis," <https://youtu.be/Dk9OEt02aDE>, 2017, last Accessed: Jan 4, 2019.
- [19] M. Guzdial, J. Reno, J. Chen, G. Smith, and M. Riedl, "Explainable pcgml via game design patterns," in *EXAG Workshop*, 2018.
- [20] S. Dahlskog and J. Togelius, "Patterns and procedural content generation: revisiting mario in world 1 level 1," in *Design Patterns in Games Workshop*. ACM, 2012.
- [21] M. Stout, "Learning from the masters: Level design in the legend of zelda," [https://www.gamasutra.com/view/feature/6582/learning\\_from\\_the\\_masters\\_level\\_.php?print=1](https://www.gamasutra.com/view/feature/6582/learning_from_the_masters_level_.php?print=1), 2012, last Accessed: Jan 8, 2019.
- [22] T. Hoeg, *The invisible hand: Using level design elements to manipulate player choice*. Guildhall at Southern Methodist University, 2008.
- [23] R. Koncewicz, "What made those old, 2d platformers so great?" <http://www.significant-bits.com/what-made-those-old-2d-platformers-so-great/>, 2009, last Accessed: Jan 4, 2019.
- [24] G. Smith, M. Cha, and J. Whitehead, "A framework for analysis of 2d platformer levels," in *SIGGRAPH symposium on Video games*. ACM, 2008.
- [25] K. Compton and M. Mateas, "Procedural level design for platform games," in *AIIDE*. ACM, 2006.
- [26] S. Larsen and E. Aarseth, *Level design patterns*. IT-University of Copenhagen, 2006.
- [27] S. Dahlskog, S. Björk, and J. Togelius, "Patterns, dungeons and generators," in *FDG*. ACM, 2015.
- [28] K. Hullett and J. Whitehead, "Design patterns in fps levels," in *FDG*. ACM, 2010.
- [29] G. Smith, R. Anderson, B. Kopleck, Z. Lindblad, L. Scott, A. Wardell, J. Whitehead, and M. Mateas, "Situating quests: Design patterns for quest and level design in role-playing games," in *ICIDS*. Springer, 2011.
- [30] T. Thompson, "The fine line between rehash and sequel: Design patterns of the super mario series," in *Design Patterns in Games Workshop*, 2015.
- [31] G. J. Winters and J. Zhu, "Guiding players through structural composition patterns in 3d adventure games," in *FDG*, 2014.
- [32] T. Pugh, "Level design tips and tricks," [http://www.gamasutra.com/blogs/TomPugh/20181022/329044/Level\\_Design\\_Tips\\_and\\_Tricks.php](http://www.gamasutra.com/blogs/TomPugh/20181022/329044/Level_Design_Tips_and_Tricks.php), 2018, last Accessed: Jan 4, 2019.
- [33] R. Terrell, "Folded level design," <http://critical-gaming.com/blog/2008/8/5/folded-level-design.html>, 2008, last Accessed: Jan 4, 2019.
- [34] A. Stamos, "Level design primer," <http://renegadesector.com/level-design-primer/>, 2014, last Accessed: Jan 4, 2019.
- [35] A. Anthropy and N. Clark, *A game design vocabulary: Exploring the foundational principles behind good game design*. London, United Kingdom: Pearson Education, 2013.
- [36] A. Anthropy, "Game dev 1," <https://docs.google.com/presentation/d/11-uRILM6jjQ1vwyb-QPphk-EIAAEiOfayMrvZwEQOQ>, 2017, last Accessed: March 5, 2018.
- [37] M. Sharif, A. Zafar, and U. Muhammad, "Design patterns and general video game level generation," *IJACSA*, vol. 8, 2017.
- [38] S. M. Beaupre and T. G. Wiles, *General Video Game Level Generation*. Worcester Polytechnic Institute, 2018.
- [39] E. Adams and J. Dormans, *Game mechanics: advanced game design*. New Riders, 2012.
- [40] R. Kremer, *Level design: concept, theory, and practice*. CRC Press, 2009.
- [41] S. Rogers, "Everything i learned about level design i learned from disneyland," <https://www.gdcvault.com/play/1305/Everything-I-Learned-About-Level>, 2009, last Accessed: March 24, 2019.
- [42] D. Milam and M. S. El Nasr, "Design patterns to guide player movement in 3d games," in *SIGGRAPH Symposium on Video Games*. ACM, 2010.
- [43] A. Khalifa, D. Perez-Liebana, S. M. Lucas, and J. Togelius, "General video game level generation," in *GECCO*. ACM, 2016.
- [44] A. Khalifa and M. Fayek, "Automatic puzzle level generation: A general approach using a description language," in *CCGW Workshop*, 2015.
- [45] S. Dahlskog and J. Togelius, "Patterns as objectives for level generation," in *Design Patterns in Games Workshop*. ACM, 2013.
- [46] A. Baldwin, S. Dahlskog, J. M. Font, and J. Holmberg, "Mixed-initiative procedural generation of dungeons using game design patterns," in *CIG*. IEEE, 2017.
- [47] S. Dahlskog, J. Togelius, and M. J. Nelson, "Linear levels through n-grams," in *International Academic MindTrek Conference*. ACM, 2014.
- [48] G. Smith, J. Whitehead, M. Mateas, M. Treanor, J. March, and M. Cha, "Launchpad: A rhythm-based level generator for 2-d platformers," *T-CIAIG*, 2011.
- [49] A. Khalifa, M. C. Green, D. Perez-Liebana, and J. Togelius, "General video game rule generation," in *CIG*. IEEE, 2017.
- [50] T. S. Nielsen, G. A. Barros, J. Togelius, and M. J. Nelson, "General video game evaluation using relative algorithm performance profiles," in *EvoStar*. Springer, 2015.
- [51] N. Shaker, J. Togelius, and M. J. Nelson, *Procedural content generation in games*. Springer, 2016.
- [52] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, "Search-based procedural content generation: A taxonomy and survey," *T-CIAIG*, vol. 3, 2011.
- [53] A. Summerville, S. Snodgrass, M. Guzdial, C. Holmgård, A. K. Hoover, A. Isaksen, A. Nealen, and J. Togelius, "Procedural content generation via machine learning (pcgml)," *TOG*, vol. 10, 2018.
- [54] Nintendo, "Super Mario Bros," NES, 1985.
- [55] Capcom, "Mega Man," NES, 1987.
- [56] Nicalis and 8bits Fanatics, "1001 Spikes," PC, PlayStation 4 and Xbox One, 2014.
- [57] Nintendo, "The Legend of Zelda," NES, 1986.
- [58] Sega, "Sonic the Hedgehog," Sega Genesis, 1991.
- [59] Namco, "Pac-Man," Arcade, 1980.
- [60] Nintendo, "Super Metroid," SNES, 1994.
- [61] ———, "New Super Mario Bros. U," Wii U, 2012.
- [62] C. W. Totten, *An architectural approach to level design*. CRC Press, 2014.
- [63] Team Meat, "Super Meat Boy," Xbox 360 and PC, 2010.
- [64] Nintendo, "Super Mario World," SNES, 1990.
- [65] Rare, "Donkey Kong Country 2: Diddy Kong's Quest," SNES, 1995.
- [66] Konami, "Castlevania Legends," Game Boy, 1997.
- [67] Nintendo, "The Legend of Zelda: A Link to the Past," SNES, 1991.
- [68] Loren Schmit, "Star Guard," PC, 2010.
- [69] Brett Taylor, "Inelight."
- [70] B. Cipliauskaité, "Foreshadowing as technique and theme in "one hundred years of solitude"," *Books Abroad*, vol. 47, 1973.
- [71] F. Muecke, "Foreshadowing and dramatic irony in the story of dido," *The American Journal of Philology*, vol. 104, 1983.
- [72] B.-C. Bae and R. M. Young, "A use of flashback and foreshadowing for surprise arousal in narrative using a plan-based approach," in *ICIDS*. Springer, 2008.
- [73] Anna Anthropy, "Mighty Jill Off," PC, 2008.
- [74] ———, "Lesbian Spider-Queens of Mars," PC, 2011.
- [75] Terry Cavanagh, "VVVVVV," PC, 2010.
- [76] Lococomito, "L'Abbaye Des Morts," PC, 2010.
- [77] Nintendo, "Super Mario Land," Game Boy, 1989.
- [78] ———, "Excitebike," NES, 1984.
- [79] DoubleDutch Games, "SpeedRunners," Multi-platform, 2013.
- [80] Capcom, "Final Fight," Arcade, 1989.
- [81] ———, "Mega Man 11," Nintendo Switch, 2018.
- [82] Tecmo, "Mighty Bomb Jack," NES, 1986.
- [83] Chiku, "Syobon Action," PC, 2007.
- [84] Michael "Kayin" O'Reilly, "I Wanna Be The Guy," PC, 2007.
- [85] Valve, "Left 4 Dead," PC and Xbox 360, 2008.
- [86] Rockstar Games, "Red Dead Redemption," Playstation 3 and Xbox 360, 2010.
- [87] Nintendo, "Super Mario 64," Nintendo 64, 1996.