Towards Effectively Adapting Games: What needs to be Conquered to Achieve Adaptation

Phil Lopes and Micaela Fonseca
Human-Environment Lab (HEI-Lab)
Lusófona University
Campo Grande 388, Libson
Portugal
E-mail: phil.lopes@ulusofona.pt
E-mail: micaela.fonseca@ulusofona.pt

KEYWORDS

Game Adaptation, Artificial Intelligence, Machine Learning, Digital Games, Game Orchestration, Procedural Content Generation

ABSTRACT

Digital games are unique in the sense that they are one of the few mediums that allow users to directly interact and engage with a virtual world. These worlds are often complex and filled with intertwining types of media such as: audio, 3D/2D visuals and intricate narratives; which in turn react to what the player is doing in-game. This interaction however, is often hand-crafted by human designers who meticulously determine the exact parameters of how the player interacts and how the game should react. One of the emerging fields within the digital game research community has been on the concept of game content adaptation, where the field consists of the development of methods that can adapt a game towards the preference of the player. Despite a lot of work existing in this field, the focus often tends to be exclusively on the construction of a system that can generate a wide variety of specific virtual artifacts (Procedural Content Generation), or the development of statistical models which provide powerful methodologies capable of profiling and classifying players (Player Modeling). This paper argues that these methods can be used for other purposes that can directly benefit the player experience, where the game would tailor the game-play to each individual or group of players. Although this is not a relatively new idea, a lack of work and methods is still prevalent when it comes to the specification of how this game-play can change in such a way that it either benefits the player experience or some other factor, such as games for rehabilitation that go beyond entertainment. More precisely, this paper argues that an abundant amount of research is being conducted on one part of the problem, but not the other: "How does a Digital Game Adapt?".

Introduction

Modern games are complex interactive audio-visual experiences, providing a set of strict rules of how players interact and how this virtual experience responds to this interaction. Often referred to as game mechanics, they consist of a series of rules that determine the methods of how a player interacts with the game and how this game given a specific interaction will respond to the player. Depending on the complexity of a game, these mechanics can often influence or intertwine with each other leading to a complex virtual ecosystem leading to emergent situations unique to each specific player (Hokkanen et al. 2018), and here lies the strength and uniqueness of digital games as a medium. Despite games already having the ability to provide unique situations for players, these often come at a cost of game-complexity, randomization or even unintended features (e.g. bugs). Most games also do not have the ability to tailor specific mechanics or the content around each individual user, whereas currently games are designed around a general audience.

Game content adaptation as proposed by (Pedersen et al. 2010) was the culmination of two popular fields of digital game research at the time: Player Modeling (PM) and Procedural Content Generation (PCG). Specifically, PM consists of classifying either players or playing behaviours, whilst PCG consists of developing algorithms capable of generating digital game content (e.g. levels) autonomously. (Pedersen et al. 2010) argued that through the collection of player experience data an association can be made between the in-game level parameters and the different types of playing experiences (e.g. Challenge, Frustration, Fun, Anxiety). By leveraging this player experience model the level generator can subsequently search the parameter space (i.e. generates a level) that ideally optimize towards an intended experience (e.g. generate a level that optimizes or minimizes challenge/fun/frustration/anxiety).

Since (Pedersen et al. 2010) the concept of game adaptation has evolved over the years, and with the current popularization of Machine Learning (ML) the idea of PM for games has only grown (Drachen et al. 2012, Yannakakis et al. 2013, El-Nasr et al. 2016), and has began to include concepts of affective computing (Yannakakis and Togelius 2011, Chanel et al. 2011, Lopes et al. 2015). Despite a lot of research being devoted to the construction of methods and models capable of recognizing and categorizing players, the "content adaptation" aspect has fell behind with a lot of the methodologies focusing on PM rather than constructing PCG algorithms capable of using PM data accordingly, thus the question still remains **"how can we tailor and personalize a game that fits both designer and player expectations?"**.

It is unsurprising that this concept has fallen behind, given that games are complex systems with intertwining digital artefacts who work in tandem creating an immersive audio-visual interactive feedback loop (i.e. the visuals and mechanics of a game responds to player interaction further stimulating the player). Thus, it can be difficult to pin-point exactly **what** and **how** content can be adapted towards a specific objective, given the difficulty of specifying exactly what changes influence player experience and the knock-on effects that adaptation can incur on the overall game. Currently, such systems are handcrafted to respond to specific triggers and gameplaying contexts defined by the game designers so as to elicit certain experiences and emotions. Thus, when dealing with an autonomous adaptation system, such handcrafted sequences must be dynamic and have the ability to respond logically to both the player interaction and the experience/emotion recognition model.

Given the varying complexity of digital games, there are multiple ways that the system can respond such as adjusting: the difficulty, different digital game facets (e.g. audio, visuals, level environments) or a combination of both. This paper will touch upon these different methods explored within the literature, which ideally will provide an outline of the different difficulties and methodologies that need to be solved to achieve dynamic game adaptation.

What is Adaptation?

The "player-experience loop" as it is often referred, consists of the process that goes from the player stimulus (i.e. the player interprets the audio-visual and sensory feedback received from the game), which ideally provokes a response from the player either mechanically (e.g. reacts to changes in the game) or physically (e.g. emotional response) which are captured through input devices (e.g. controllers) and sensors (e.g. physiological recordings, eye tracking, facial expression analysis). All this data is interpreted by a statistical model which outputs a set of values that corresponds to the players current experience (or emotional state) - using game difficulty as an example where a model outputs a value between 0 and 1, the closer the value is to 0 or 1 suggests that the player is finding the game too easy or hard, respectively. Using the information obtained by the model the game is expected to *adapt*, by either making the game harder if it is too easy (or vice-versa), which in turn restarts the *loop*. Although the concept of difficulty is easy to understand in theory, mechanically it can provide a series of design challenges that can be quite cumbersome even for human designers (Aponte et al. 2011). More precisely, the complexity of a game and the increase of different mechanics can influence the perceived challenge of a game, for example the challenge in a game of *Tetris* (Pajitnov, 1984) is widely different than a game of *Super Mario Bros.* (Nintendo, 1985).

Digital games come in a variety of complexities, which have only increased over the years due to the modernization of hardware, the wider availability of development that ease the development process allowing for more complexity and more intricate input devices that now can incorporate haptics and sensory feedback. Thus, it is no wonder why a large portion of the community has often applied the topics of adaptivity on classical games such as *Tetris* (Chanel et al. 2011) and *Pong* (Liu et al. 2009). Dealing with this complexity is potentially the greatest challenge within the field of game adaptation.

Game adaptation can come in multiple forms, it can focus on specific parameters such as difficulty (Chanel et al. 2011, Liu et al. 2009), attempt to guide the player emotionally (Lopes et al. 2015) or even adapt narrative (Hartsook et al. 2011). Whatever the parameter may be, games are intertwining complex experiences, where the mechanics, visuals, audio and narrative all work in tandem to create whatever experience the designer intended (Fullerton 2019). Thus, this aspect can be thought as a "creative" endeavour, where the machine or algorithm must be able to creatively adapt content to keep the player effectively engaged (Liapis et al. 2014). Thus, this specific topic spreads over multiple fields of research such as: Computational Creativity (Boden 1998), Affective Computing (Picard 2003), Artificial Intelligence (AI) (Liapis et al. 2014), Game Design and even Psychology. This section will provide first a view of why adapting can be problematic, what different methodologies exist that could potentially help solve this problem to get closer to our game adaptation objective.

Dynamic Difficulty Adjustment

Dynamic Difficulty Adjustment (or DDA) as the name suggests consists of adapting a game's difficulty so that it matches a player's skill-set dynamically. It is still one of the most popular game adaptation fields within the literature (Zohaib 2018). Since the early definition by (Hunicke 2005) the concept of DDA has normally taken the work of Csikszentmihalyi's Flow Theory (Csikszentmihalyi and Csikszentmihalyi 1992) as inspiration, whereas ideally during play the goal of the designer is to place the player in a state of flow - a men-

tal state where the person loses their perception of time, becoming fully absorbed in the task at hand. (Csikszentmihalyi and Csikszentmihalyi 1992) defines flow as a delicate balance between boredom and anxiety/frustration, where flow is achieved if the difficulty of said task stays within a boundary between both states. Considering that the player gains more proficiency as they play, DDA argues that by adjusting the difficulty dynamically it is easier to keep the player within this flow state and adjust the mechanics according to each players "learning curve".
Since the early 2000's games have attempted to apply some of these concepts in some shape or form. One of the earliest examples is Max Payne (Remedy Entertainment, 2001), which would monitor player health and add or remove health packs throughout a level considering a player performance metric. Within the literature this concept has often focused on the construction of frameworks and models offering potential solutions for DDA (Zohaib 2018) whilst work that has directly applied DDA into actual games are slimmer, often focusing either on classic simple games such as Tetris (Chanel et al. 2011, Hufschmitt et al. 2021), Pong (Liu et al. 2009, Darzi et al. 2021) or exclusively modifying Non-Player Character (NPC) behaviours (Yang et al. 2009, Tan et al. 2011). The reasoning why this is such a difficult concept to put into practice is that difficulty, similar to other game adaptation problems, is often influenced by a wide variety of complex and intertwining parameters that can influence the perception of difficulty. For example the perception of difficulty in a racing game (Togelius et al. 2007) varies significantly than that of a platforming game such as *Super Mario Bros.* (Pedersen et al. 2010). Thus, simple straightforward games the perception of difficulty tend to be reliant on a lower number of parameters that need to be manipulated. To fully convey the concept of difficulty it is also important to understand what game complexity is. Complexity within this paper is defined as the amount of intertwining mechanics that exist within a game, where each underlying mechanic has a direct relationship with the perception of difficulty in some way. For example:

- **Tetris**: is a low complexity game as difficulty often consists of either withholding certain Tetriminos or increasing/decreasing their falling speed.
- **Pacman**: the complexity increases given that its difficulty lies directly with the ghosts themselves - improving their AI directly influences difficulty of the game, which is less straightforward than Tetris.
- **Super Mario Bros.**: the complexity increases as several factors can increase the difficulty of this game. Such as wider gaps (i.e. more difficult jump), less power-ups, more enemies – the level design itself influences difficulty, which means levels need to adapt autonomously.
- **Call of Duty**: the complexity increases again as several factors in this game can make the game harder – level design, enemy positioning, enemy health values, enemy shooting effectiveness, player health, player guns available.

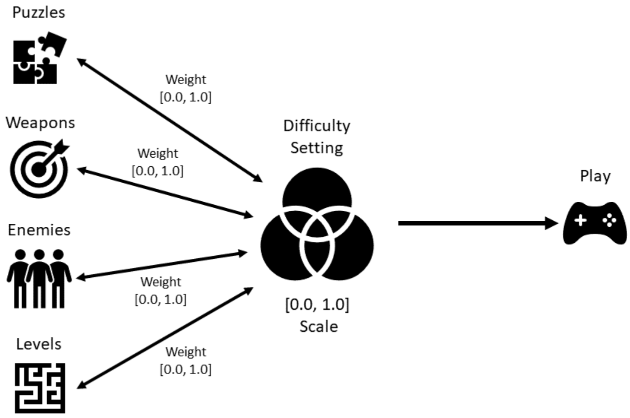

Figure 1: Condensing difficulty factors into one difficulty variable. Each factor can have a specific weight, influencing the final "difficulty score", which in turn influences play.

The combination of different "difficulty parameters" can rise exponentially given the mechanical complexity of a game. Whereas it can even become difficult to playtest as there are so many diverging parameters that can influence and alter the perception of difficulty towards players. Thus, both from a game design and an AI perspective, condensing difficulty parameters (see Figure 1) can be quite beneficial as a tool for designers and facilitate DDA for AI models. However, this is not as straight-forward as it may seem as each parameter is substantially different when categorizing difficulty and may have more/less influence on the overall perception of difficulty. Not to mention that certain aspects such as levels or puzzles, are either pre-defined by designers (each of them are hand crafted and subsequently tagged a specific difficulty metric) or an autonomous process creates a level/puzzle with an expected difficulty of x, for example. As such DDA can be explored through both adaptation techniques described further below, i.e. Facet Parameter Adaptation and Procedural Content Generation, or even a hybrid method of the two. Depending on the complexity of the game, difficulty may derive from the level design itself which may require a Content Generation solution such as in *Super Mario Bros.* (Pedersen et al. 2010) or *Angry Birds* (Rovio Entertainment, 2009) (Ferreira and Toledo 2014), or may be something as simple as adapting the falling velocity parameter of tetrimino pieces in *Tetris* (Chanel et al. 2011).

Going Beyond Difficulty

Despite difficulty being one the most popular types of game adaptation, other game-playing experience factors can also have considerable benefits for players. For example (Pedersen et al. 2010) already suggested focusing on other types of experiences such as "fun" or "frustration" at the time, which (Yannakakis and Togelius 2011) further explored. Furthermore, games can also influence players emotionally (Yannakakis and Paiva 2014) given the audio-visual nature of the medium. Thus, another goal for optimizing adaptation algorithms can be through labels of emotional psychological theory such as: (Ekman 1999) basic emotion theory or the more granular emotional theories like the (Russell et al. 1989) circumplex model of affect. In essence, the latter is the application of affective computing methods (Picard 2003) within the domain of game content adaptation, e.g. optimizing a horror game based on the varying levels of player anxiety (Lopes et al. 2015, Graja et al. 2020).

Methods of Adaptation

Game adaptation – or simply adaptation – in the context of this paper refers to an autonomous process whereby game mechanics and/or virtual content are optimized (or minimized) towards an intended metric. For example an adaptation algorithm can consist of a system that changes the environmental lighting and audio to induce different degrees of anxiety (Lopes et al. 2015), or alter the architecture of a level inducing more or less challenging game-play (Shaker et al. 2010).
The following sections will provide an overview of how adaptation methods have been explored previously within literature and their current limitations.

Game Content Orchestration

(Liapis et al. 2018) defines orchestration as: *The harmonization of the game generation processes.* More precisely, (Liapis et al. 2018) state that games are interactive multimedia experiences that can be manipulated in multiple ways. Orchestration is the process that "combines" these diverging systems (e.g. level, sound and visual generator) into something cohesive and playable (see Figure 2). Metaphorically, much like a musical conductor who signals the different instruments in an orchestra, orchestration consists of the development of autonomous systems that signal or modify the different audio-visual and mechanical parameter generators so as to achieve an intended player experience or aesthetic (i.e. the metric). For example, in the previous work by (Graja et al. 2020), the authors explored how changing light intensity and colour, in addition to different sound cues can potentially be used for the manipulation of tension and player anxiety within a horror game. Another

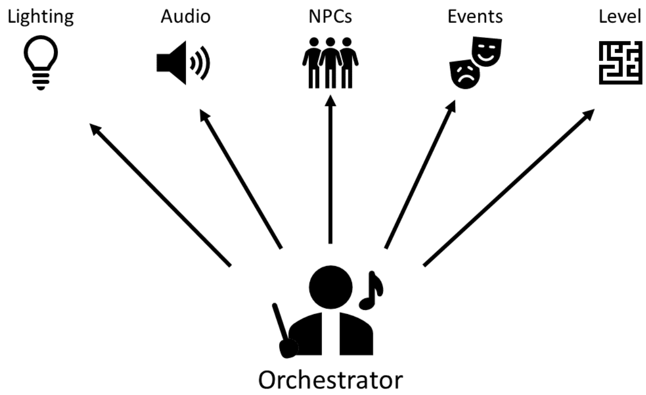

Figure 2: Example of an Orchestration Model. Orchestration controls and manipulates the different facet parameters of a game to dynamically provide different game-playing experiences and aesthetics.

popular example is the AI Director from the *Left 4 Dead* series (Valve Corporation, 2008-2009). Although, the most popular feature of the algorithm are adversarial aspects of the game, allowing the system to control the flow of enemies sent towards the player (i.e. the zombies), it also had features that would control the amount of weapons and health that would appear around the map and modify certain paths forcing players to detour from the main path. All decisions are taken according to a *curve of suspense*, where the system constantly monitors the player performance and the current state of the game. In fact, the director system is one of the first industry attempts of adapting games based on pace and *suspense* metric. If the players are going through the level too fast, the system will try and complicate matters by restricting ammunition and health and increasing the amount of enemies, on the other hand if the players are struggling the system will offer more locations for "downtime" allowing players to restock and wind down until facing another "outbreak".
(Liapis et al. 2018) further decomposes the concepts of orchestration into an additional approach referred to as *bottom-up* (while the latter is referred to as *top-down*). The *Bottom-Down* approach forgoes the overarching system, where all facet generators work towards a common or diverging goals, ideally resulting in an emergent coherent whole.

Facet Parameter Adaptation

Facet Parameter Adaptation consists of systems that slightly alter specific in-game parameters of handcrafted content. The previously mentioned game *Left 4 Dead*, is a good example of Facet Adaptation, where the system for example directly alters specific paths of the level with other handcrafted ones, or adjust the flow of appearing enemies accordingly. Facet adaptation had been used to invoke certain audio-visual aesthetics (Graja et al. 2020), adjust the speed of falling tetriminos in *Tetris* (Chanel et al. 2011) or the distribution of health packs and monsters in the game *Left 4 Dead*.

Procedural Content Generation

Unlike Adaptation, Procedural Content Generation (PCG) consists of creating algorithms that can construct content "from scratch" (Togelius et al. 2010, Shaker et al. 2016). For example, by using a rule-based system with a degree of randomization it is possible to construct entire virtual universes with different fauna and flora – *No Man's Sky* (Hello Games, 2016). This concept has existed since the 1980's with games such as *Rogue* (Wichman and Arnold, 1980) and *Elite* (Braben, 1984), where the latter popularized the "Rogue-like" genre by integrating the generation into the gameplay (i.e. new levels every time the player "dies"); and the former popularized the idea of large open generated universes filled with galaxies and planets. However, both games examples showcase some of the different concepts within the PCG field. PCG in particular has also been used quite extensively within literature from the generation of racing tracks (Togelius et al. 2007), levels of *Angry Birds* (Ferreira and Toledo 2014) and even *Super Mario Bros.* (Pedersen et al. 2010).
It is also important to consider that most PCG algorithms exploring multi-facet generation tend to include some form of Facet Parameter Adaptation within the algorithmic process, thus it is quite common to see some form of hybridization between PCG and the latter within literature. Particularly the work of (Nogueira et al. 2016, Lopes et al. 2015) explores the generation of dungeons by combining facet adaptation of audio and lighting.

Deterministic PCG
Deterministic PCG is as the name suggests, PCG algorithms that will generate the same world every time. This method is often used to save disk space, allowing developers to re-use assets and create worlds that would be impossible otherwise (e.g. The Large Universes of No Man Sky, for example). The game Elite is one of the first examples of doing this – a game with a gigantic world that could be saved on a single floppy disk.
These types of systems often have no randomization and are entirely guided by deterministic functions that will generate the same world every time. These types of algorithms are used mostly for optimizing disk space.

Exploratory PCG
Unlike deterministic PCG, exploratory uses randomization for the construction of diverging content each time the algorithm is run. These types of systems are often accompanied by a numerical seed, which can be defined by the developer (i.e. same seed for all), or players (i.e. share the seed so others can play in a similar world), which in turn influence the randomization functions.

The strength of this method is its diverging capabilities, meaning that every time a player starts a new game it will generate content that has (ideally) never been previously experienced by the Player. Most approaches, however, do tend to use a Hybrid-type approach, where the generation can often have certain deterministic parameters in addition to a degree of randomness.

PCG Granularity

When discussing granularity in PCG it often refers to the extent of control that is given to the generator. For example, let's take the levels of the popular game *Super Mario Bros.* (Nintendo, 1984). One way of generating levels for this game is to think of a level as pre-made subsections (i.e. small vertical sliced chunks of levels), which are placed next to each other logically. In this example, each chunk is designed by a human, which was then picked and placed by an autonomous system. This form of PCG is used by the popular game *Spelunky* (Yu, 2008), and is often referred to as a "Low Granular" generator.

"Higher Granularity" is relying on less human-authored content and offering more control on the generator. Using the Super Mario Bros. example again, a more granular generator would place each individual tile-piece. Such as placing the ground, tubes, enemy, power-ups, gaps and where the level finishes. As such the only human-authored content in this situation are the tile designs themselves, but how they are formed and structured in a level is entirely defined by the generator. As expected, these types of generators are far more complex, but can offer significantly more variability in their generated levels.

Offline and Online Generation – What does it mean?

These concepts often refer to when exactly the generation takes place, which is before play (offline) or during play (online). Most PCG systems currently employ an offline tactic, where most of the generation takes place before actual play. The previous examples such as: *No Man's Sky* or *Spelunky* all employ an offline method where all the generation takes place a-priori to play.

Online methods are generators that will generate content during actual gameplay and is often more suited for adaptation methods. The problem with this type of generation is that it can often be intensive and cause framerate issues, and it is more difficult to implement as it must work side-by-side with the actual game. On the plus side, these types of games respond to player emotion/experience almost instantaneously i.e. while the game is actually being played (Nogueira et al. 2016). This can be advantageous for the model as it is able to address divergences that occur during actual play rather than "in-between" different levels, for example.

Learning to Orchestrate

The current limitation of orchestration is that all facets need to have some form of "controllability" which allows the orchestrator to modify them. For example, if the designer wants the lighting to be controllable, the variables of this facet must be accessible to the orchestrator (e.g. intensity, angle, color). More complex systems such in-game events or designing Non-Playable Characters (NPC) still currently requires the hand of a designer (see figure 3). For example, an event can be as simple as opening or closing a door (e.g. effective way of creating tension in a horror game) or unleashing a specific type of monster towards the player (e.g. the choice is dynamic, but the monster/event was designed by a human). For generative algorithms which already have to deal with their own algorithmic parameters, will potentially require an additional "meta-parameter(s)" extending beyond the generator, relating its outputted content within the context of the overall scenario and how it "harmonizes" with other facet generators and parameters informing the orchestrator of the current scenario (i.e. once all elements are combined, what is the playing experience or aesthetic quality?).
Thus, here lies the main challenge when developing such a system as the complexity of intertwining content can lead to a wide range of player experience output, which ideally should be data-driven through extensive game-playing data collection. For example by annotating the player experience and constructing relations between the scenario meta-parameters and said annotation, it would be possible to extend said relations to each generator/adaptation's local parameters. However, such an optimization could be costly particularly in pin-pointing exactly the weight and influence of each generator on the overall experience. Thus, before even creating an orchestrator one must be able to understand how to collect such a wide-range of multi-faceted data for the construction of an orchestration system in the first place.

Data Collection Process and Annotation

Let us further assume that we have an orchestrator which can access different content modifying parameters of the game, such as controlling level parameters (e.g. funnel players towards specific locations) or light and audio sources (e.g. set and trigger different sounds). Given the available parameters and the information obtained from the model, how does the game modify these values to maximize metric x. According to (Graja et al. 2020), one way to accomplish this is by creating a range of static scenarios, in which the aesthetic quality is measured through play-testing or "tagged" by designers. Given the number of combinations, depending on how many parameters exist, it can be extremely difficult to pin-point exactly the type of aesthetic conveyed. Not to mention personalizing this content further complicates matters given that in this situation the aesthetic goal varies along with the said parameters, thus a sort of "calibration" system would be necessary to accomplish this task. One solution could be through autonomous testing using player archetype models such as in (Holmgård et al. 2018). Recent advances in bayesian parameter estimation (Jaakkola and Jordan 2000) and optimization (Shahriari et al. 2015) could be a solution, as a way of reducing the search space between scenarios and allowing our data collection process to focus on the specific relevant parameters instead of brute forcing through the entire spectrum of values. Furthermore, it is also important to consider that certain facets can also play off each other, such as how dark lights can complement somber sounds (Ekman and Lankoski 2009), which can further increase the complexity of the problem given how this interplay-ability can also influence player aesthetics in a wider variety of ways, either from the audio-visual perspective (Palmer et al. 2013) or through the way interacting mechanics and dynamics of play (Hunicke et al. 2004).

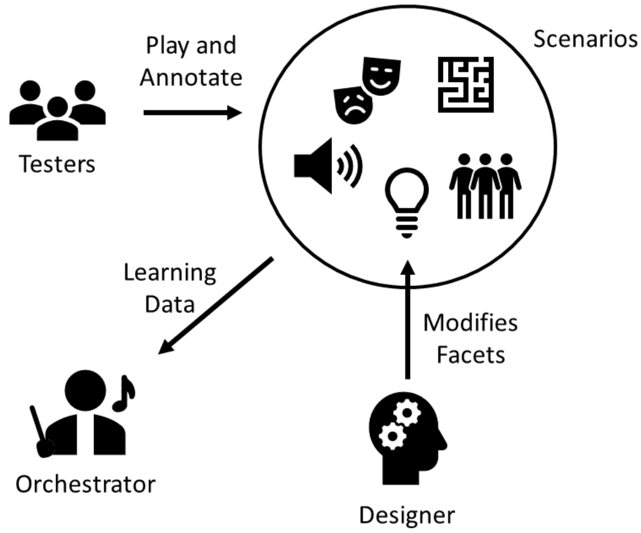

Figure 3: Data Collection Methods for "Orchestration Learning".

Game Aesthetic Assumptions

For the purposes of context aesthetics can be defined as the perceived emotion that a certain combination of digital game facets is trying to evoke. If we take horror as an example, the audio, the player movement, and the visuals are often constructed and designed in such a way that it maximizes the feeling of player helplessness such as Amnesia: The Dark Descent (Frictional Games, 2010). On the other hand, games that intend to make the players feel powerful rely on other types of mechanics and visuals whereas the player can easily overcome certain challenges, for example.

As an alternative to the bottom-up approach, it may also be worthwhile to explore top-down methods using game-playing and audio-visual aesthetics as domain knowledge. Instead of focusing on solely collecting data to construct our algorithms, it can also be beneficial to "aid" autonomous systems with domain knowledge, which can further increase the quality of the content. For example, it may be beneficial to provide scenarios, which are theoretically proven to induce certain types of emotion to further reduce the search space of the orchestrator. Previous work explored the idea of game playing aesthetics and even film aesthetics (Lopes et al. 2015, Yannakakis and Paiva 2014, Graja et al. 2020) and how it can potentially benefit the construction of generators. It is also viable to have the designers themselves tag and annotate the different facet parameters according to emotion/challenge, which can include the various combinations between diverging facets ideally aiding the decision making process of an orchestrator. Despite the laborious process for designers such systems have been used within the industry to create more dynamic systems that lead to more emergent situations such as *Left 4 Dead*.

Conclusions

Dynamic content adaptation for digital games is not a straightforward task as it can depend on a series of factors that can influence the complexity of said adaptation. Despite the amount of work that already exists within the field of statistical player modeling, the literature often focuses on just that - modeling player emotion, experience or likelihood to continue playing (Yannakakis and Togelius 2011, Yannakakis et al. 2013, Yannakakis and Paiva 2014, Debeauvais et al. 2011). This paper argues that statistical models should be applied for this purpose, however to achieve this it is necessary to look beyond statistical modelling and explore methodologies capable of altering the state of the game through either established Game AI techniques (e.g. Orchestration or PCG) and the further exploration of Game Aesthetics as inspiration for different adaptation methodologies.

Acknowledgements

The work is co-funded by Fundação para a Ciência e Tecnologia (FCT), under HEI-Lab center (UIDB/05380/2020).

REFERENCES

Aponte M.V.; Levieux G.; and Natkin S., 2011. *Measuring the level of difficulty in single player video games. Entertainment Computing*, 2, no. 4, 205–213.

Boden M.A., 1998. *Creativity and artificial intelligence. Artificial intelligence*, 103, no. 1-2, 347–356.

Chanel G.; Rebetez C.; Bétrancourt M.; and Pun T., 2011. *Emotion assessment from physiological signals for adaptation of game difficulty. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 41, no. 6, 1052–1063.

Csikszentmihalyi M. and Csikszentmihalyi I.S., 1992. *Optimal experience: Psychological studies of flow in consciousness.* Cambridge university press.

Darzi A.; McCrea S.M.; Novak D.; et al., 2021. *User Experience With Dynamic Difficulty Adjustment Methods for an Affective Exergame: Comparative Laboratory-Based Study. JMIR Serious Games*, 9, no. 2, e25771.

Debeauvais T.; Nardi B.; Schiano D.J.; Ducheneaut N.; and Yee N., 2011. *If you build it they might stay: Retention mechanisms in World of Warcraft.* In *Proceedings of the 6th International Conference on Foundations of Digital Games*. 180–187.

Drachen A.; Sifa R.; Bauckhage C.; and Thurau C., 2012. *Guns, swords and data: Clustering of player behavior in computer games in the wild.* In *2012 IEEE conference on Computational Intelligence and Games (CIG)*. IEEE, 163–170.

Ekman I. and Lankoski P., 2009. *Hair-raising entertainment: Emotions, sound, and structure in silent hill 2 and fatal frame. Horror video games: Essays on the fusion of fear and play*, 181–199.

Ekman P., 1999. *Basic emotions. Handbook of cognition and emotion*, 98, no. 45-60, 16.

El-Nasr M.S.; Drachen A.; and Canossa A., 2016. *Game analytics.* Springer.

Ferreira L. and Toledo C., 2014. *A search-based approach for generating angry birds levels.* In *2014 IEEE Conference on Computational Intelligence and Games*. IEEE, 1–8.

Fullerton T., 2019. *Game design workshop: a playcentric approach to creating innovative games.* AK Peters/CRC Press.

Graja S.; Lopes P.; and Chanel G., 2020. *Impact of Visual and Sound Orchestration on physiological arousal and tension in a horror game. IEEE Transactions on Games.*

Hartsook K.; Zook A.; Das S.; and Riedl M.O., 2011. *Toward supporting stories with procedurally generated game worlds.* In *2011 IEEE Conference on Computational Intelligence and Games (CIG'11)*. 297–304. doi:10.1109/CIG.2011.6032020.

Hokkanen V.; Holmes T.; Koivuranta H.; Sandberg A.; Sorva H.; Toikka J.; Hämäläinen P.; and Kaos M., 2018. *Plusminus: Augmenting physics to promote emergent gameplay.* In *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts.* 321–327.

Holmgård C.; Green M.C.; Liapis A.; and Togelius J., 2018. *Automated playtesting with procedural personas through MCTS with evolved heuristics. IEEE Transactions on Games*, 11, no. 4, 352–362.

Hufschmitt A.; Cardon S.; and Jacopin É., 2021. *Dynamic Manipulation of Player Performance with Music Tempo in Tetris.* In *26th International Conference on Intelligent User Interfaces.* 290–296.

Hunicke R., 2005. *The case for dynamic difficulty adjustment in games.* In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology.* 429–433.

Hunicke R.; LeBlanc M.; and Zubek R., 2004. *MDA: A formal approach to game design and game research.* In *Proceedings of the AAAI Workshop on Challenges in Game AI.* San Jose, CA, vol. 4, 1722.

Jaakkola T.S. and Jordan M.I., 2000. *Bayesian parameter estimation via variational methods. Statistics and Computing*, 10, no. 1, 25–37.

Liapis A.; Yannakakis G.N.; Nelson M.J.; Preuss M.; and Bidarra R., 2018. *Orchestrating game generation. IEEE Transactions on Games*, 11, no. 1, 48–68.

Liapis A.; Yannakakis G.N.; and Togelius J., 2014. *Computational game creativity.* The International Conference on Computational Creativity.

Liu C.; Agrawal P.; Sarkar N.; and Chen S., 2009. *Dynamic difficulty adjustment in computer games through real-time anxiety-based affective feedback. International Journal of Human-Computer Interaction*, 25, no. 6, 506–529.

Lopes P.; Liapis A.; and Yannakakis G., 2015. *Targeting horror via level and soundscape generation.* In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment.* vol. 11.

Nogueira P.A.; Torres V.; Rodrigues R.; Oliveira E.; and Nacke L.E., 2016. *Vanishing scares: biofeedback modulation of affective player experiences in a procedural horror game. Journal on Multimodal User Interfaces*, 10, no. 1, 31–62.

Palmer S.E.; Schloss K.B.; and Sammartino J., 2013. *Visual aesthetics and human preference. Annual review of psychology*, 64, 77–107.

Pedersen C.; Togelius J.; and Yannakakis G.N., 2010. *Modeling player experience for content creation. IEEE Transactions on Computational Intelligence and AI in Games*, 2, no. 1, 54–67.

Picard R.W., 2003. *Affective computing: challenges. International Journal of Human-Computer Studies*, 59, no. 1-2, 55–64.

Russell J.A.; Lewicka M.; and Niit T., 1989. *A cross-cultural study of a circumplex model of affect. Journal of personality and social psychology*, 57, no. 5, 848.

Shahriari B.; Swersky K.; Wang Z.; Adams R.P.; and De Freitas N., 2015. *Taking the human out of the loop: A review of Bayesian optimization. Proceedings of the IEEE*, 104, no. 1, 148–175.

Shaker N.; Togelius J.; and Nelson M.J., 2016. *Procedural content generation in games.* Springer.

Shaker N.; Yannakakis G.; and Togelius J., 2010. *Towards automatic personalized content generation for platform games.* In *Sixth artificial intelligence and interactive digital entertainment conference.*

Tan C.H.; Tan K.C.; and Tay A., 2011. *Dynamic game difficulty scaling using adaptive behavior-based AI. IEEE Transactions on Computational Intelligence and AI in Games*, 3, no. 4, 289–301.

Togelius J.; De Nardi R.; and Lucas S.M., 2007. *Towards automatic personalised content creation for racing games.* In *2007 IEEE Symposium on Computational Intelligence and Games.* IEEE, 252–259.

Togelius J.; Yannakakis G.N.; Stanley K.O.; and Browne C., 2010. *Search-based procedural content generation.* In *European Conference on the Applications of Evolutionary Computation.* Springer, 141–150.

Yang J.; Gao Y.; He S.; Liu X.; Fu Y.; Chen Y.; and Ji D., 2009. *To create intelligent adaptive game opponent by using Monte-Carlo for tree search.* In *2009 Fifth International Conference on Natural Computation.* IEEE, vol. 5, 603–607.

Yannakakis G.N. and Paiva A., 2014. *Emotion in games. Handbook on affective computing*, 2014, 459–471.

Yannakakis G.N.; Spronck P.; Loiacono D.; and André E., 2013. *Player modeling. Dagstuhl Follow-Ups*, 6.

Yannakakis G.N. and Togelius J., 2011. *Experience-driven procedural content generation. IEEE Transactions on Affective Computing*, 2, no. 3, 147–161.

Zohaib M., 2018. *Dynamic difficulty adjustment (DDA) in computer games: A review. Advances in Human-Computer Interaction*, 2018.