Implementing Adaptive Game Difficulty Balancing in Serious Games

Hendrix, M, Bellamy-Wood, T, McKay, S, Bloom, V & Dunwell, I

Original citation & hyperlink:
Hendrix, M, Bellamy-Wood, T, McKay, S, Bloom, V & Dunwell, I 2018, 'Implementing Adaptive Game Difficulty Balancing in Serious Games' *IEEE Transactions on Games,* vol (in press), pp. (in press)
https://dx.doi.org/10.1109/TG.2018.2791019

DOI 10.1109/TG.2018.2791019
ISSN 2475-1502
ESSN 2475-1510

Publisher: IEEE

Implementing Adaptive Game Difficulty Balancing in Serious Games

Maurice Hendrix, Tyrone Bellamy-Wood, Sam McKay, Victoria Bloom, Ian Dunwell

Abstract— **The ability to engage and retain players is perceived as a major factor in the success of games. However, the end-goal of retention differs between entertainment and serious contexts. For an entertainment game, engagement and retention is linked to monetization; for a serious game, this needs to persist for as long as is required for learning or behavioural objectives to be met. User engagement is strongest when a balance is achieved between difficulty and skill, leading to a state of "flow". Hence adapting difficulty could lead to increased and sustained engagement. Implementing this requires the identification of variables linked to mechanics, manipulated based upon a player performance model. In some cases, this is possible by adjusting simple properties of objects, though more comprehensive solutions require extending or adapting content applying procedural techniques. This paper proposes a six step plan, validated against two case studies: an existing serious game, with easily-manipulated parameters, and a platformer game built from scratch, where additional content is required, showing the process for different mechanics. To explore limitations, the results of two small-scale user evaluations with 45 users in total, are reported, contributing to the understanding of how adaptive difficulty might be implemented and received.**

***Index Terms*—**1.2.1 [**Artificial Intelligence**]: Applications and Expert Systems – Games.

I. Introduction

ATTRACTING players attention in a busy marketplace has traditionally been one of the most important challenges for game developers. Player retention is often equally important for serious games, which often rely on a period of retention to convey educational or behavioural outcomes. Retention is also centrally important in many mobile gaming contexts, particularly those relying upon in-game purchases for revenue. A common way to enhance player retention is using virtual reward systems, often with features such as scores, badges or levels. However, a certain level of challenge is a crucial aspect of this reward system, as rewards which are achieved too easily will be less valued [1]. Equally, a goal that appears impossible to achieve, may cause players to disengage and stop playing the game. Csikszentmihalyi [2] posits that engagement and focus on a task commonly arises when there is a balance between challenge and skill. This is described as leading to a state of "flow", or the "flow channel", wherein the challenge increasing as the level of skill increases. Both Oris et al. [3] and Belanich et al. [4] have shown that task difficulty and prior video game experience impact performance and motivation in gamers. Adaptive difficulty may be able to provide players with a constantly appropriate level of challenge, increasing engagement and retention.

There is a growing amount of research into in-game difficulty balancing [5]–[7] and the games industry has in the past implemented approaches with games such as Left4Dead [8], which uses an artificial intelligence controller to determine the amount and type of foes and pick-ups that are created depending on a team's performance. Conversely, however, in games such as the Dark Souls [9] series, flow is sought through design of progressively challenging static encounters, rather than by adjusting the level of challenge dynamically. In general, a ubiquitous approach or solution is difficult to propose, due to the diversity of genres [10], [11], platforms (or engines) [12], [13], and audiences [14]. Indeed, game developers typically strive to calibrate difficulty to be in line with most players within the target audience, and often utilise achievements, timing, or scoring, to allow players to gain varying levels of success, rather than face arbitrary "succeed or fail" outcomes. This allows players to deliberately adopt a more challenging playstyle, to gain an achievement or higher score.

A further common affordance is the ability for the player to select their own difficulty setting either before or during play, allowing them to directly attempt to match the level of challenge to their perceived skill. Player-driven difficulty selection can be both explicit, for example in a menu, or implicit, such as a choice of team in a sports game. These implicit choices are increasingly observable in game designs, however, the traditional choice between "Easy", "Medium" or "Hard" frequently persists.

This paper explores how games might adapt dynamically and individualistically, rather than relying on pre-sets which, by nature, require a subjective assessment by the player of their skill level, and can disrupt flow in the critical early stages of engagement, as the player may have to determine their ideal challenge level through a frustrating cycle of trial-and-error.

M. Hendrix, T. Bellamy-Wood, V. Bloom and Ian Dunwell are with the School of Computing, Electronics and Mathematics, Faculty of Engineering, Environment and Computing, Coventry University, UK (e-mail: {ab0776,ac1393,ab9488,aa6537}@coventry.ac.uk).
S. McKay is with The GenGame, UK (e-mail: sammckay@btinternet.com).

To this end a six-step plan is proposed for including adaptive game balancing in games. Then this plan is validated through two case studies: an existing serious game based on a puzzle and 3D shooter mechanic, as well as a platform game developed from scratch. The rationale behind this is to validate the steps using different mechanics. While platform games do not have a serious purpose, the mechanic has successfully been used in a number of Serious Games [15]–[18].

II. RELATED WORK

A. Adaptive Game Difficulty

Hunicke [19] argues the need for adaptive game difficulty, and various efforts have been made to realize dynamic adaptive game difficulty in practice. This has been accomplished in various ways; for example through dialogue delivery through non-player characters as accomplished by Peirce, at al. [20], adjusting variables within the game itself such as difficulty level, or varying the amount of enemies encountered [21]. In some cases, game difficulty is linked to the content or assets used. For example, when jumping between platforms, the size of platforms and the distance between them affects difficulty if the player's velocity, jump height, and gravity remain constant.

Caro et al. [22] defined a methodology for creating adaptive educational games which includes some dynamic generation of stories and structure. In this case however the aim is to achieve adaptive difficulty balancing in a more generic way through the use of procedural content generation (PCG) [23]–[25], a technique for generating new content and assets that meet the requirements of the game in real-time.

B. Player Experience Models

As serious games have emerged, the need has arisen to tailor these games to the competences and characteristics of different players to replicate pedagogical best practices and better suit players. A particular challenge exists for serious games, as their target demographic is often determined by the problem they seek to address, rather than as a specific segment of the gaming community, and efforts have been made towards implementing adaptive difficulty in a serious gaming context [20], [26]. Player Experience Models are relatively new concepts in the game development world [27]: their main use is to assess how the player is performing within the game, this information can, and frequently is, used to create adaptive difficulty in games [28]. This means that the games difficulty can be adjusted to the player's needs as elicited from their gameplay data, rather than user- or play-testing. Various techniques have been used such as adding and removing pre-defined elements to a scenario such as enemies and items [29], adjusting the behaviour of the artificial intelligence [30], and using case-based reasoning [31]. Some authors have also considered difficulty balancing in multiplayer games [32]. Whilst these different models have been proposed [33]–[35], there is no standardized experience model that can be integrated seamlessly, mainly due to the large challenge of creating a one-size fits all mode given the number of different genres, platforms and technologies in use. Therefore, game developers will need to effectively develop their own model. As discussed in the remainder of this paper, this does not have to be an insurmountable challenge.

C. Procedural Content Generation

Procedural Content Generation (PCG) is a method that has been in use since the 1980's with games such as Rogue (1UP 2008). It is used for a variety of reasons including reduced development time/cost, smaller file sizes, more or infinite content, and greater levels of randomness [23]. Notably the first Elite game [36] was able to store 2048 planet's data in just 22Kb of memory in 1984 [37]. More recently PCG has been used in games like No Man's Sky [38], Minecraft [39] and SkySaga [40]. PCG typically utilises a random but repeatable sequence of numbers, allowing content to be generated programmatically and consistently from an initial seed. Allowing the sequence to be repeatable permits the programmer to "save" a version of the content. There are two main types of PCG: offline and real-time. In offline PCG, a piece of content is generated to aid development and possibly adapted by a developer. An example of this is the use of software to generate a model of a city at design time [41], saving significant development time. Real-time PCG happens when the game is being played. Content being generated as and when it is needed can introduce unpredictability and increase re-playability, as for example in Left 4 Dead [42]. In addition to the methods mentioned, PCG can also be based on more sophisticated AI methods [25] such as reinforcement learning [43].

Procedural story generation techniques [44], [45] have also been used to generate story lines. While this can be very valuable especially for open-world games, the proposed method, needs the procedural generation to be able to take account of desired levels of difficulty. For example, in procedural story generation, if the story is purely based on context it will be difficult to adopt the difficulty, other than by reducing the amount of hints or contextual information to the player.

Hendrikx et al. introduced a game content pyramid [44]. Within 6 levels of game content that can be procedurally generated in theory. These levels are:

- game bits, e.g. textures, sounds etc.
- game space, e.g. maps
- game systems, e.g. eco systems, road networks etc.
- scenarios, e.g. storyboards or puzzles
- game design, e.g. system design or world design
- derived content, such as leader boards

In this work the focus is on the first three levels, as well as manually created scenarios which have been parameterised, with a view of adapting the difficulty. Generation of game designs and new scenarios, while interesting research areas, are beyond the scope of this work.

Procedural creation has its own limitations. Conventionally, players consume content until the "end" of the game, usually in the form of levels or hand-crafted environments. In an infinite, procedural context, content consumption is strongly related to the progressive understanding the player develops of the procedural algorithm. Once the player understands its limitations, and can predict accurately the form and format of future content, the senses of exploration, challenge, and discovery become diminished. A further risk with procedural generation is the challenge testing large volumes of generated content, or that difficulty may be highly variable unless it is carefully controlled.

Moreover, for serious games, procedural content creation many need to incorporate a further dimension related to the educational value of content. This can significantly complicate the challenge, which becomes a task with dual objectives: how can we ensure the user is engaged and retained by procedural content, but without compromising, or whilst enhancing, educational or behavioural outcomes through PCG? In the scope the cases reported here, we adopt the approach of disaggregating the low-level gameplay mechanics from high-level learning outcomes; in simple terms seeking to apply PCG to a 'shooter' aspect of gameplay, whilst retaining learning mechanics linked to combining – and learning properties of – nutrients to upgrade abilities. This has immediate advantages in allowing PCG to be applied via an indirect, rather than direct link to learning content – as the player upgrades abilities, they perform more strongly in the 'shooter' aspect of the game, and therefore the game adapts indirectly to their learning progress. This is not a ubiquitous model; however, it provides a basis for enabling low-level mechanics to be adapted with reference to, rather than a direct impact upon, the more abstract high-level pedagogical model.

III. Implementation Method

As the previous section indicates, while research on adaptive game difficulty exists, ubiquitous models or frameworks demonstrating how a game developer can practically incorporate adaptive game difficulty are lacking. Based on our experience in game development and evaluation [12], [46]–[54], as well as teaching game development, we propose the following pragmatic six step plan for implementing adaptive difficulty:

1. The first step is to identify variables in the game that are good indicators of player performance. This can be achieved by the game rules and victory conditions. Typical examples include game scores, time taken to complete tasks, or number of tasks completed.
   At this stage, it should also be decided how these variables indicate performance. A linear model is the most straightforward and intuitive in many cases.

2. The second step is to determine variables that influence the difficulty of the game and that can be changed. To achieve this, examine the game play and level design. Typical examples include the number of time given for a task, the complexity of the map or environment, the number and strength of enemies encountered, and the number of tasks (or goals) the player needs to perform simultaneously.

3. If an implementation of the game exists, locate the performance and difficulty indicating variables.

4. Consider whether the game features multiple mechanics and if so, to which mechanic do performance and difficulty variables relate. For example, in many games the player needs to navigate an obstacle course as well as fight enemies. Or navigate an environment whilst completing puzzles. If there are several sufficiently separate mechanics in the game, different difficulties with their own variables for each mechanic should be considered.

5. Decide how the performance variables will be used to calculate difficulty. A pragmatic approach for most developers is to choose a simple calculation, such as a weighted average, or a set number of different levels. More complicated models such as [35], or machine learning techniques such as reinforcement learning [30], [55], could be employed here.

6. Decide upon sensible starting values for the identified variables, impacting the difficulty balancing. The developer could make a best estimate but it would be better to base the default values on a test with a few players.

The plan starts by examining the key player performance indicator. When designing games, it is quite common to consider game mechanics and victory conditions as one of the earliest aspects in the design process. Especially in a serious game context, user assessment (indicators) is sometimes not straightforward. Starting by looking at this aspect, ensures that developers find a good way to measure player performance before proceeding with implementation. If no good way can be found, applying the plan is not suitable. However, one could argue that these are more interactive experiences rather than truly being games.

The rest of this chapter, shows the process of implementing the plan into an actual game. In Section A, an existing serious game is adapted to incorporate a simple game difficulty balancing, based on existing in-game variables. Section B shows how PCG can be used in a side-scrolling platform game in which the user jumps between obstacles.

A. *Adapting in game variables*

The game used to show the process of integrating adaptive game difficulty is PEGASO (Personalised Guidance Services for Optimising Lifestyle in Teenagers) [56] The game is a mobile game and has been developed using the Unity [57] game engine. The game is set in a post-apocalyptic scenario where the player takes on the role of a survivor in day and night cycles. To achieve this there are two main elements, food is gathered by means of completing puzzles during the day cycle, shown in Figure 1, while during the night cycle energy from this food needs to be used to fight off zombies, shown in Figure 2.

Below the six-step plan is followed to incorporate adaptive game difficulty into this game, while in its later stages of development.

1. The first step is to identify what elements of the game affect difficulty by examining the gameplay and victory conditions. The PEGASO game does not have a difficulty setting, but there are multiple variables in the game which can be used to monitor the progression of the player. We decided upon a linear difficulty model based on the following variables:

   - For the zombie shooting mechanic:
     - number of zombies killed
     - highest nightly score
     - number of times died
     - number of nights survived

   - For the food matching puzzle:
     - amount of food gathered
     - experience points
     - game boards completed
     - nights survived

Figure 1 PEGASO game puzzle mechanic


Figure 2 PEGASO game zombie fighting mechanic

2. The second step is to identify what elements influence the game difficulty. For the PEGASO game the main ways that difficulty could be influenced is:

   - The difficulty level and speed of zombie enemies.
   - The "amount of chances to match food" by changing the variety of food available in the puzzles and the sizes of the boards. Smaller puzzles with low variety and more chances to match are considered the easiest whereas larger puzzles with more variety and less chances to match are considered hardest.

3. An early implementation of the game existed in the latter stages of development. The variables identified in the existing code are *zombiesKilledThisNight,* and *level*, in a script which tracks player statistics in terms of the amount of zombies killed and the level of the puzzle mechanic. A new separate script is introduced to track the number of deaths in a variable called *deathCounter*.

4. The game features two quite distinct mechanics: a puzzle mechanic during day time for collecting food and a zombie fighting mechanic during night time. It is quite possible for players to be better at one of these. Therefore, it was decided to introduce two separate difficulty calculations.

5. It is important to note these mechanics relate to, rather than directly are, the principal learning and behavioural mechanics. As noted in Section II, the approach was implemented by selecting the aspects of the game most readily compatible with adaptive difficulty, rather than the components of the game most central to the serious objectives. The overall mechanics of the game utilise the food collected in a 'crafting' system, to allow the player to combine food items to gain abilities, gaining insight into nutrition and nutritional properties in the process. Furthermore, an 'energy' system rewards the player for undertaking real-world actions (e.g. using a pedometer), by increasing their rate of progression. Due to space and scoping constraints, it is impossible to detail the game's mechanics in their entirety here; see [58] for this detail

6. In terms of the calculation three variables were introduced to track average player performance over different playthroughs. These variables are the *average*, *lower* and *higher* quartile. These are tracked for both the puzzle mechanic and for the zombie fighting mechanic. The player's performance is compared to the lower and higher quartiles. If a player performs better than the average the values are scaled up and if he performs worse the values are scaled down. The variable start at 10, 5 and 15 respectively and are scaled, together with zombie difficulty, as follows

$$s = \begin{cases} (f > h, 50\% \\ h \leq f > l, 25\% \\ f < l, -50\% \end{cases}$$

Where
s = the scaling factor
f = the amount of food collected in a day
h=the higher quartile of performances up till now
l= the lower quartile of performances up till now

The player's performance is compared to the lower and higher quartiles. If a player performs better than the average the values are scaled up and if he performs worse the values are scaled down. The scaling factor depends on how much better or worse the player performs. In particular, if the player falls outside the lower or higher quartile the values are scaled by 50%, otherwise they are scaled by 25%. These factors were set during development and were determined empirically by a small amount of experimentation by the developers.

7. Default starting variables were decided by the developers based on picking numbers that seemed reasonable. To verify this, two other members of the team played the game with these default variables set fixed for the duration of the test. While this is clearly not a representative sample, this allowed for a considered starting value, which in turn is adaptable based on further data from players.

B. PCG based adaptation approach

The previous section documents the six-step plan applied to an existing game whereas this section applies the plan to a new game. Applying the plan to a new game gives developers the chance to make the variables easily accessible. The new game is a simple but engaging side-scrolling platform game in which the user needs to jump between obstacles. This allows us to test the six step plan with a different mechanic, also used for Serious Games [15]–[18], as well as with a game built from scratch. As the balancing is intended to work directly on the platformer mechanic we limited development to this mechanic and did not include any serious aims.

In this game, it was easy to identify and influence variables that controlled game difficulty and performance.

The game design is based the platformer genre, similar to Mario Bros. The player sees a 2D side view of the game world including an avatar representing the player. The player needs to guide the avatar to the finish by jumping between suspended platforms. In order to implement this game design, a prototype implementation was built as a mobile game, using the Unity 3D [57] game engine. Below is the result of applying the six step plan.

1. The first step is to identify what elements of the game affect difficulty by examining the gameplay and victory conditions. In this game a good indication of performance is how often a player dies by falling into the gap between platforms. Another way of looking at this is how far they manage to get in a level in a certain amount of time, restarting every time the player dies. In this case we decided to use a linear model of difficulty.

2. The second step is to identify which elements influence the game difficulty. The main way that game difficulty could be influenced is the distance between and placement of platforms. However, this requires a way to indicate distance and placement of platforms within the variables and requires a way to present platforms of different sizes and placements on demand.

   To achieve this PCG will be utilized. A rectangular base-platform is created as an asset and the generation algorithm varies the width and the space between platforms. This is then based on a difficulty variable. This variable can then be changed based on performance.

3. No implementation exists.

4. The game design is quite simple and only features one single mechanic.

5. The player experience model is shown in Figure 3. The time expired and distance travelled are tracked.

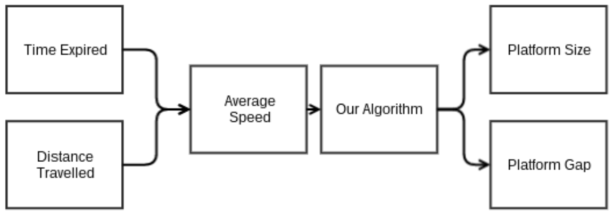


Figure 3 Player experience model for PCG difficulty balancing

   Then the average speed is used as the difficulty level. This is then used to calculate platform size and platform gap, by simply multiplying the starting variable by a normalized average speed. The average speed is normalized as follows:

$$N_{speed} = \begin{cases} \dfrac{avg_{speed}}{def_{speed}} - 1, & \dfrac{avg_{speed}}{def_{speed}} > 1 \\ 1 + 1 - \left(\dfrac{avg_{speed}}{def_{speed}}\right), & \dfrac{avg_{speed}}{def_{speed}} \leq 1 \end{cases}$$

   Where
   N_{speed} = the normalized average speed.
   avg_{speed} =the current recorded average speed.
   def_{speed} = the default starting speed.

6. The starting variables were decided by the developer empirically. The default values can be adjusted if required.

IV. VALIDATION

In order to validate the proposed six step plan, user studies were performed with the two example games. While a large-scale evaluation with multiple games and more participants would be ideal, these studies provide a valuable insight into whether our development and by extension our six step plan, has been applied correctly. While in both cases a convenience, sample was used, participants did not know which version they were testing and the testing orders of the versions was varied to prevent the learning effect and participants' familiarity with the developers from influencing the results.

A. PEGASO game

The adapted PEGASO game was evaluated using a convenience sample of eight test subjects with no background knowledge in PEGASO. The convenience sample consisted of friends and family of the developers and was split between keen mostly male gamers between 18 and 22 with some development experience and people who rarely play games, between 40 and 50 equally split between male and female, and have no development experience. Four people tried the original game before trying the version with the difficulty balancing, while the other four tried the difficulty balancing version first. The participants played both game versions for three day and night cycles. Their performance was tracked in terms of zombies killed, food collected and whether they died, as well as which version they thought was more fun and engaging to play.

Out of eight participants five preferred the balanced version and the remaining three had no preference. None of the respondents preferred the non-balanced version. Figure 4 and Figure 5 show player performance for each player for each play through.

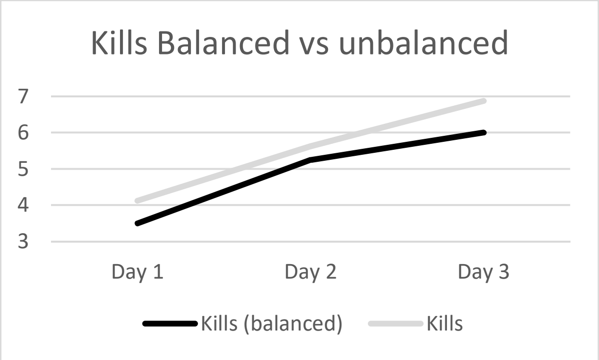


Figure 4 Zombies killed

The four players to the left played the difficulty balancing version first while the others played the non-balanced version first. In terms of performance, as can be expected, players get better with multiple play-throughs. They also perform worse on the difficulty balancing version. Zombies killed and food collection especially seem to follow a similar improvement pattern for most players. Open-ended feedback was requested from two of the eight players, who mentioned that the balanced game seemed to adapt to their skill level.

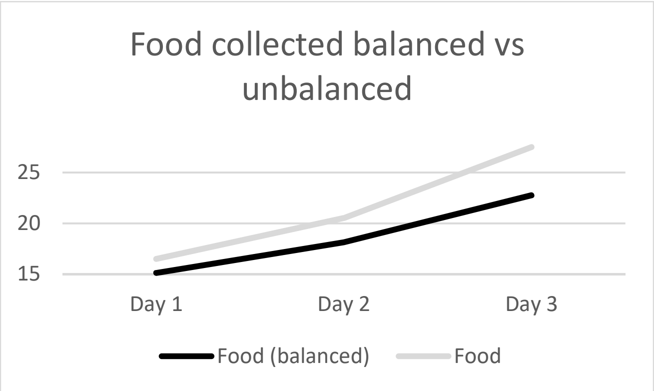


Figure 5 Food collected

B. *Platformer game*

The platformer game was tested with the help of an online survey. It was deployed to the web using the WebGL functionalities of Unity, so that the game could be publicly hosted. 37 participants were recruited using several game related Facebook groups. No further information was captured about the participants. Participants tried a version of the game with difficulty balancing and with a fixed manually set difficulty and were not aware of the difference between the versions. As Figure 6 shows, participants had a varied gaming background, and 89% had regular gaming experience. This is to be expected given that the survey was distributed through game related Facebook groups.

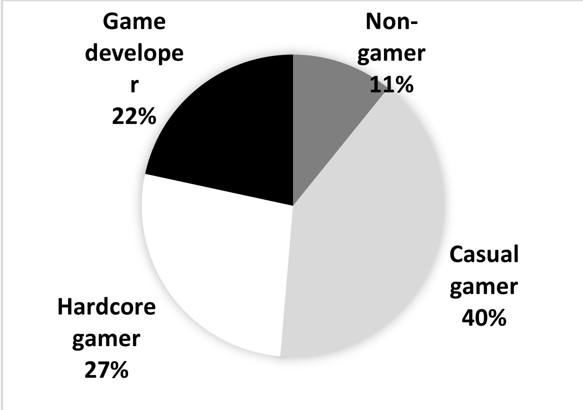


Figure 6 Gaming experience platformer game survey

Whilst, as shown in Figure 8, a greater number of participants for the adaptive difficulty setting strongly agreed the version of the game was "fun", analysis of the results did not show significance under a U-Test (U=11.5, Z=0.10, p=0.46). Self-reporting itself is a limited means by which to examine the broader constructs of "engagement" or "flow", and considering how to operationalise the experience of play is an ongoing research challenge.

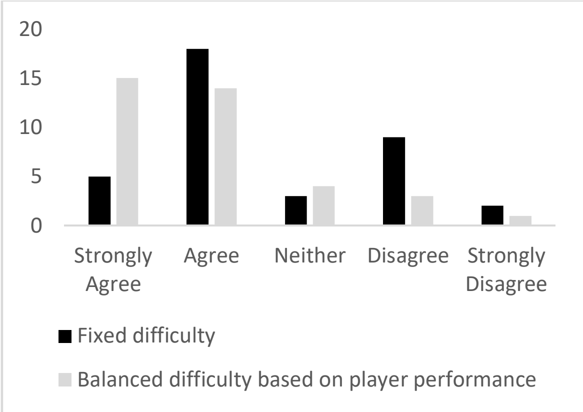


Figure 7 Platformer game - this version of the game is fun

V. Conclusions & Further Work

The ability to engage players is crucial to the success of a game. It can be equally important in serious games, particularly those that rely on player engagement and retention to deliver the serious message. However, every player is different. Engagement is considered to work best when the player is offered a challenge that fits their level of capability, creating a state of flow [2]. In this paper a six-step plan for implementing difficulty balancing in games was presented. The plan was evaluated by applying it to two different types, a mobile game with a zombie fighting and puzzle mechanic as well as a classic platformer game. Small scale user evaluations were conducted of both these games which compared a version with difficulty balancing against a version without to show that they worked as intended. It remains important to manually determine the optimal balancing for the intended target audience, however, as this heavily influences the experience when first playing a game.

With reference to the comparison between entertainment and serious games, results tentatively demonstrate that the process as applied to entertainment games can be successfully carried over to serious contexts. As noted in Section III, we deliberately avoided modifying core learning mechanics, and rather scaffolded adaptive difficulty such that learning aspects influenced indirectly, rather than directly, the adaptive difficulty approach. This is self-limiting in that it is applicable only to games whose designs support such a technique. Clearly, a significant space exists for future work here that examines how PCG and adaptive difficulty can be specifically implemented to directly modify, extend, and adapt educational content dynamically.

The six-step plan presents a relatively simple and usable set of steps that a game developer can follow and is not dependent on game genre, platform or technology chosen.

In terms of future work it would be interesting to conduct larger scale evaluations. Another avenue of research is creating a framework that can provide difficulty balancing, with common performance models and balancing algorithms built in. To this end the plan would need to be expressed more formally. However, there are significant challenges to overcome to achieve this, such as finding a practical way to cater for the plethora of platforms, programming languages and games engines used and supporting effective use of authoring difficulty balancing in such a framework. As a practical way forward developing plug-ins for popular game engines such as Unity 3D [57], will allow developers to continue using their favourite engine.

VI. Acknowledgements

This work has been part-supported by the European Commission under the collaborative project PEGASO ("Personalised Guidance Services for Optimising Lifestyle in Teenagers") funded by the European Commission under the Seventh Framework Programme, FP7-ICT-2013-10.

References

[1] E. Aronson and J. Mills, “The effect of severity of initiation on liking for a group.,” *J. Abnorm. Soc. Psychol.*, vol. 59, no. 2, p. 177, 1959.

[2] M. Csikszentmihalyi, *Flow: The psychology of optimal experience*. Harper Perennial, 1991.

[3] K. A. Orvis, D. B. Horn, and J. Belanich, “The roles of task difficulty and prior videogame experience on performance and motivation in instructional videogames,” *Comput. Hum. Behav.*, vol. 24, no. 5, pp. 2415–2433, 2008.

[4] J. Belanich, D. E. Sibley, and K. L. Orvis, “Instructional characteristics and motivational features of a PC-based game,” DTIC Document, 2004.

[5] R. Lopes and R. Bidarra, “Adaptivity challenges in games and simulations: a survey,” *IEEE Trans. Comput. Intell. AI Games*, vol. 3, no. 2, pp. 85–99, 2011.

[6] M. Mehta and A. Ram, “Runtime behavior adaptation for real-time interactive games,” *IEEE Trans. Comput. Intell. AI Games*, vol. 1, no. 3, pp. 187–199, 2009.

[7] G. N. Yannakakis and J. Hallam, “Real-time game adaptation for optimizing player satisfaction,” *IEEE Trans. Comput. Intell. AI Games*, vol. 1, no. 2, pp. 121–133, 2009.

[8] “Left4dead,” *left 4 dead*. [Online]. Available: http://www.valvesoftware.com/games/l4d.html. [Accessed: 29-Mar-2017].

[9] “Dark Souls II.” [Online]. Available: http://www.darksoulsii.com/us/. [Accessed: 29-Mar-2017].

[10] T. H. Apperley, “Genre and game studies: Toward a critical approach to video game genres,” *Simul. Gaming*, vol. 37, no. 1, pp. 6–23, 2006.

[11] M. J. Wolf, “Genre and the video game,” *Medium Video Game*, pp. 113–134, 2001.

[12] P. Petridis, I. Dunwell, S. Arnab, A. Protopsaltis, M. Hendrix, and S. de Freitas, “Game Engines Selection Framework for High-Fidelity Serious Applications,” *Int. J. Interact. Worlds*, vol. 2012, 2012.

[13] B. Cowan and B. Kapralos, “A survey of frameworks and game engines for serious game development,” in *Advanced Learning Technologies (ICALT), 2014 IEEE 14th International Conference on*, 2014, pp. 662–664.

[14] “Take a look at the average American gamer in new survey findings,” *Polygon*, 29-Apr-2016. [Online]. Available: http://www.polygon.com/2016/4/29/11539102/gaming-stats-2016-esa-essential-facts. [Accessed: 05-Apr-2017].

[15] S. Tang and M. Hanneghan, “Game content model: an ontology for documenting serious game design,” in *Developments in E-systems Engineering (DeSE), 2011*, 2011, pp. 431–436.

[16] J. Fanfarelli, “The effects of narrative and achievements on learning in a 2D platformer video game,” 2014.

[17] J. R. Fanfarelli, S. Vie, and others, “Medulla: A 2D sidescrolling platformer game that teaches basic brain structure and function,” *Well Play.*, vol. 4, no. 2, pp. 7–29, 2015.

[18] V. McArthur and R. J. Teather, “Serious mods: A case for modding in serious games pedagogy,” in *Games Entertainment Media Conference (GEM), 2015 IEEE*, 2015, pp. 1–4.

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

[20] N. Peirce, O. Conlan, and V. Wade, “Adaptive educational games: Providing non-invasive personalised learning experiences,” in *Digital Games and Intelligent Toys Based Education, 2008 Second IEEE International Conference on*, 2008, pp. 28–35.

[21] J. Hall-Tipping, *Video game difficulty level adjuster dependent upon player’s aerobic activity level during exercise*. Google Patents, 1991.

[22] R. M. Carro, A. M. Breda, G. Castillo, and A. L. Bajuelos, “A methodology for developing adaptive educational-game environments,” in *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, 2002, pp. 90–99.

[23] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, “Search-based procedural content generation,” in *European Conference on the Applications of Evolutionary Computation*, 2010, pp. 141–150.
[24] N. Shaker, J. Togelius, and M. J. Nelson, *Procedural Content Generation in Games: A Textbook and an Overview of Current Research. Springer.*, 1st ed. 2016 edition. New York, NY: Springer, 2016.
[25] G. N. Yannakakis and J. Togelius, “Experience-driven procedural content generation,” *Affect. Comput. IEEE Trans. On*, vol. 2, no. 3, pp. 147–161, 2011.
[26] F. Bellotti, R. Berta, A. De Gloria, and L. Primavera, “Adaptive experience engine for serious games,” *Comput. Intell. AI Games IEEE Trans. On*, vol. 1, no. 4, pp. 264–280, 2009.
[27] G. N. Yannakakis, P. Spronck, D. Loiacono, and E. André, “Player modeling,” *Dagstuhl Follow-Ups*, vol. 6, 2013.
[28] A. Zook and M. O. Riedl, “A Temporal Data-Driven Player Model for Dynamic Difficulty Adjustment.,” in *AIIDE*, 2012.
[29] R. Hunicke and V. Chapman, “AI for dynamic difficulty adjustment in games.,” presented at the Challenges in game artificial intelligence (AAAI Workshop), Pittsburgh, 2004.
[30] G. Andrade, G. Ramalho, H. Santana, and V. Corruble, “Extending reinforcement learning to provide dynamic game balancing,” in *Proceedings of the Workshop on Reasoning, Representation, and Learning in Computer Games, 19th International Joint Conference on Artificial Intelligence (IJCAI)*, 2005, pp. 7–12.
[31] S. Bakkes, P. Spronck, and J. Van den Herik, “Rapid and reliable adaptation of video game AI,” *IEEE Trans. Comput. Intell. AI Games*, vol. 1, no. 2, pp. 93–104, 2009.
[32] H. Prendinger, K. Puntumapon, and M. Madruga, “Extending Real-Time Challenge Balancing to Multiplayer Games: A Study on Eco-Driving,” *IEEE Trans. Comput. Intell. AI Games*, vol. 8, no. 1, pp. 27–32, 2016.
[33] F. De Grove, J. Van Looy, and C. Courtois, “Towards a serious game experience model: validation, extension and adaptation of the GEQ for use in an educational context,” *Play. Play. Exp.*, vol. 10, pp. 47–61, 2010.
[34] A. Kultima and J. Stenros, “Designing games for everyone: the expanded game experience model,” in *Proceedings of the International Academic Conference on the Future of Game Design and Technology*, 2010, pp. 66–73.
[35] S. Chaisriya, L. Gilbert, and G. B. Wills, “The game challenge model,” 2013.
[36] “Frontier Developments.” [Online]. Available: https://www.frontier.co.uk/games/elite/. [Accessed: 29-Mar-2017].
[37] F. Spufford, *Backroom boys*. Faber & Faber, 2010.
[38] “No Man’s Sky,” *No Man’s Sky*, 13-Mar-2014. [Online]. Available: https://www.nomanssky.com/. [Accessed: 29-Mar-2017].
[39] “Minecraft.net,” *Minecraft.net*. [Online]. Available: https://minecraft.net/en-us/. [Accessed: 29-Mar-2017].
[40] “SkySaga - Welcome to SkySaga.” [Online]. Available: https://www.skysaga.com. [Accessed: 29-Mar-2017].
[41] G. Kelly and H. McCabe, “Citygen: An interactive system for procedural city generation,” in *Fifth International Conference on Game Design and Technology*, 2007, pp. 8–16.
[42] M. Booth, “The ai systems of left 4 dead,” in *Keynote, Fifth Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE09)*, 2009.
[43] V. Rieser and O. Lemon, *Reinforcement learning for adaptive dialogue systems: a data-driven methodology for dialogue management and natural language generation*. Springer Science & Business Media, 2011.
[44] M. Hendrikx, S. Meijer, J. Van Der Velden, and A. Iosup, “Procedural content generation for games: A survey,” *ACM Trans. Multimed. Comput. Commun. Appl. TOMM*, vol. 9, no. 1, p. 1, 2013.
[45] D. D. Merritt and B. Z. Liles, “Narrative AnalysisClinical Applications of Story Generation and Story Retelling,” *J. Speech Hear. Disord.*, vol. 54, no. 3, pp. 438–447, 1989.
[46] M. Hendrix *et al.*, “Authoring Adaptive Serious Games,” in *7th European Conference on Technology Enhanced Learning*, 2012.
[47] M. Hendrix *et al.*, “Serious Games and E-Learning Standards: Towards an Integrated Experience,” *J. Adv. Distrib. Learn. Technol. JADLeT*, vol. 1, 2013.
[48] I. Dunwell *et al.*, “A Game-Based Learning Approach to Road Safety: The Code of Everand,” in *CHI’14: Proceedings of the 2014 CHI Conference on Human Factors in Computing Systems*, 2014.
[49] I. Dunwell *et al.*, “MeTycoon: A Game-Based Approach to Career Guidance,” in *International Conference on Games and Virtual Worlds for Serious Applications (VS-Games)*, 2013.
[50] P. Backlund and M. Hendrix, “Educational Games - Are They Worth the Effort? A Literature Survey of the Effectiveness of Serious Games,” in *2013 5th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES)*, 2013, pp. 1–8.
[51] I. Dunwell, P. Petridis, M. Hendrix, S. Arnab, M. Al-Smadi, and C. Guetl, “Guiding Intuitive Learning in Serious Games,” in *ALICE 2012 workshop, CISIS 2012*, 2012.
[52] I. Dunwell, S. Arnab, M. Hendrix, and P. Petridis, “Developing a Digital Game to Support Cultural Learning amongst Immigrants,” in *1st International Workshop on Intelligent Digital Games for Empowerment and Inclusion (IDGEI) at Foundations of Digital Games 2013*, 2013.
[53] I. Dunwell, P. Petridis, S. Arnab, A. Protopsaltis, M. Hendrix, and S. de Freitas, “Blended Game-Based Learning Environments: Extending a Serious Game into a Learning Content Management System,” in *Intelligent Networking and Collaborative Systems (INCoS), 2011 Third International Conference on*, 2011, pp. 830–835.
[54] S. Arnab *et al.*, “The development approach of a pedagogically-driven serious game to support Relationship and Sex Education (RSE) within a classroom setting,” *Comput. Educ.*, vol. 69, no. 0, pp. 15–30, Nov. 2013.

[55] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, vol. 1. MIT press Cambridge, 1998.
[56] “Translating Open Data to Educational Minigames Creating Minigames for Nutritional Learning using Open Data,” presented at the In Semantic and Social Media Adaptation and Personalization (SMAP), 2016.
[57] “Unity - Game Engine,” *Unity*. [Online]. Available: https://unity3d.com. [Accessed: 29-Mar-2017].
[58] I. Dunwell, R. Dixon, K. C. M. Bul, M. Hendrix, P. M. Kato, and A. Ascolese, “Translating open data to educational minigames,” in *2016 11th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP)*, 2016, pp. 145–150.

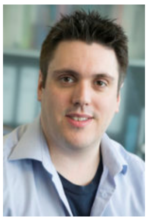
Biographies

Dr Maurice Hendrix is a Lecturer in games software development at the Faculty of Engineering, Environment and Computing, at Coventry University. His research interests centre around personalisation and adaptation, especially in educational games. He has experience of working in several European research projects including the GRAPPLE and MEducator FP7 projects and the EduGameLab Life Long Learning Project (now Erasmus+). His current teaching portfolio includes Games and Artificial Intelligence, physics for computer graphics and game development in Unity3D. He completed a BSc (2004) and MSc (2005) in Computer Science from Eindhoven University of Technology and a PhD in Computer Science from The University of Warwick (2010), on authoring adaptive personalised course material. He has extensive expertise of building both web-based systems and educational games as well as expertise in system evaluation and has published over 65 papers in the area.

Mr Tyrone Bellamy-Wood graduated from Coventry University with a BSc in Games Technology in 2017 with a First. He is currently a Software Developer at Cooperative Web. He conducted a summer internship, which implemented difficulty balancing algorithms in an existing Serious Game. In addition, he has published work in gamification with interest in gamifying complex tasks into semi-autonomous computer intelligent designs.

Mr Sam McKay graduated from Coventry University with a BSc in Games Technology (with honours) in 2016. He is currently the Lead Game Developer at GenGame Ltd where he is developing several games for mobile platforms with a key focus on the energy sector. His dissertation project looked at using procedural generation to create adaptive difficulty balancing in a platformer game.

Dr Victoria Bloom is a Lecturer in Games Software Development at Coventry University’s Faculty of Engineering, Environment and Computing. She received her PhD from Kingston University and also holds degrees from Imperial College and Kings College London. Her research interests are computer vision and machine learning techniques for human action recognition in video games.



Dr Ian Dunwell is a Senior Lecturer in Serious Games within Coventry University’s Faculty of Engineering, Environment and Computing, Coventry University and a Research Fellow at the University’s Serious Games Institute, leading the area of educational games. Having obtained his PhD in Computer Science in 2007, he also holds a degree in Physics from Imperial College London, and is an Associate of the Royal College of Science. His research interests lie primarily in the application of an understanding of cognitive psychology as a means for providing optimised, evaluated, and effective learning experiences or healthcare interventions, and developing underlying enabling technologies. He led the final stage delivery of the evaluation of Code of Everand, commissioned in 2009 by the Department for Transport and the largest publicly-funded serious game project in the UK to date. Under the UK EPSRC, he has acted as co-investigator on the £1.2m Servitize project, using games to promote servitisation in industry, and as PI of a successful industrial CASE grant, and as a reviewer for the funding council. He also led the SGI contribution to ALICE, a €2.2m EU-funded FP7 project developing next-generation adaptive learning environments, which was awarded the highest possible rating by the European Commission on conclusion in 2012. Ian is currently leading contributions to the €14.2m FP7 project PEGASO, €1.5m H2020 project OrbEEt, and Erasmus+ project GOAL.