Dynamic Difficulty Adjustment in a Multiplayer Minecraft Server

ZHEYIN ZENG and PENNY SWEETSER, The Australian National University, Australia

Minecraft has given rise to diverse gameplay, including public servers that provide survival gameplay combined with Massively Multiplayer Online Role-Playing Game elements. In these servers, gameplay includes many players fighting mobs (non-player enemies) together. However, it often happens that some players are experienced, while others are not. This leads to a problem where some players enjoy the level of challenge and consequently the game, while others can feel bored or discouraged. This paper proposes an approach to Dynamic Difficulty Adjustment (DDA) that adjusts mobs in multiplayer Minecraft so that the difficulty level accommodates all players involved. We present a new Multiplayer Minecraft DDA Framework, a game design to populate our framework, and a user evaluation study to test our framework. Our work contributes to understanding DDA in multiplayer contexts and to creating better multiplayer experiences for Minecraft and other multiplayer games.

CCS Concepts: • **Software and its engineering → Interactive games**; • **Information systems → Massively multiplayer online games**; • **Applied computing → Computer games**; • **Human-centered computing → User studies**.

Additional Key Words and Phrases: video games, dynamic difficulty adjustment, multiplayer, Minecraft

ACM Reference Format:
Zheyin Zeng and Penny Sweetser. 2022. Dynamic Difficulty Adjustment in a Multiplayer Minecraft Server. In *33rd Australian Conference on Human-Computer Interaction (OzCHI '22), November 30-December 2, 2022, Canberra, ACT, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Minecraft was released in 2011 and began with survival and construction gameplay. Since then, it has expanded to include many ways to play beyond its creator's original intention [5], especially in its versatile multiplayer community [6]. This includes many public Minecraft servers[1] that provide survival and cooperative gameplay combined with Massively Multiplayer Online Role-Playing Game (MMORPG) elements. In these Minecraft servers, players can "grow", becoming stronger during their adventure in the fantasy world. However, the strength of non-player enemies (mobs) near the players might not keep pace with the players, creating a mismatch between the strength of players and mobs.

In the context of multiplayer Minecraft, the strength mismatch is more complicated. A Minecraft survival server usually has a fixed difficulty, so all players in the server share the same difficulty. As a result, new players can meet enemies that are too strong, creating a frustrating experience due to the mismatch. Further, it often happens that a group of players with varied skill gaps fight the same group of non-player enemies together in a local area. If the strength of enemies does not match that of players, more experienced players might find it boringly easy, while less experienced players might find it discouragingly hard. Consequently, the play experience of some players in the group can be negatively affected. However, matching the strength of enemies to every player in a local area, to create a well-balanced experience for all, is challenging.

[1] https://minecraft-mp.com/stats/

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Association for Computing Machinery.
Manuscript submitted to ACM

Dynamic Difficulty Adjustment (DDA) allows game content to be tailored to meet the skill level of players [18]. For example, the strength of mobs can be adjusted to match nearby players. However, DDA is most often used in a single-player context and developing a multiplayer DDA system is more complex [1]. In the context of Minecraft survival mode, the goal of our DDA system is to set the strength of enemies to provide the right level of challenge for players. This requires defining which parameters should be considered in determining the relative strength of players and enemies. In the multiplayer context, we also need to determine how to group players into a local area and how to balance the skill of different players in a group versus nearby enemies. Therefore, our core research question is how do we design an effective and enjoyable DDA system for a multiplayer Minecraft server? In order to answer this question, we designed, implemented, and evaluated a new DDA framework within a multiplayer Minecraft server for a set of target scenarios. Our framework allows developers to systematically implement a DDA system that adjusts the difficulty of mobs (non-player enemies) shared by multiple players. Our work contributes to understanding and developing multiplayer DDA systems that facilitate improved player experience for players of different skill levels in Minecraft and other games.

2 RELATED LITERATURE

Dynamic Difficulty Adjustment (DDA) commonly refers to techniques that automatically adjust the difficulty of a game in real-time so that players achieve an optimal match between skill and challenge as they progress through a game. A key concept underlying DDA systems is that of Flow [3] and also GameFlow [8–16]. Flow is the psychology of optimal experience with a balance between skill and challenge at its core. If there is a mismatch, the experience can become boringly easy or discouragingly hard. According to GameFlow, games must hold the player's attention through a high workload, with tasks that are sufficiently challenging to be enjoyable. The player must be skilled enough to undertake the challenging tasks, the tasks must have clear goals so that the player can complete the tasks, and the player must receive feedback on progress towards completing the tasks. If the player is sufficiently skilled and the tasks have clear goals and feedback, then the player will feel a sense of control over the task. The resulting feeling for the player is total immersion or absorption in the game, which causes the player to lose awareness of everyday life, concern for themselves, and alters their sense of time. The final element of GameFlow, social interaction, does not map to the elements of flow, but represents the importance of social experiences in games. In general, the goal of DDA is to dynamically adjust the game difficulty so that the challenge matches the player's skill level throughout the game, as well as for players with different levels of skill. DDA has been widely investigated [4] in both game development [17] and research [18] in recent years, with a general aim to improve player experience by customising game difficulty.

A recent literature review identified several approaches used in DDA systems [18]. Although the reviewed studies provide approaches to evaluating the player [4, 19] and how the DDA system should intervene [4], they do not tackle the specific challenges for multiplayer games. There is some previous research that investigates DDA in multiplayer games [1, 7]. Baldwin et. al [1] proposed a framework for multiplayer DDA (mDDA). mDDA is different to single-player DDA, which focuses on the adjustment of non-player opponents, in that mDDA focuses on reducing the difference between players in a competitive multiplayer game. Although the study focuses on competitive multiplayer, it provides inspiration for cooperative multiplayer, such as players fighting mobs in Minecraft. A more recent study [7] presented an innovative method to implement an mDDA system. Instead of relying on the internal game states to evaluate players' enjoyment, the study uses an external sensor, an EEG headset, to detect the activity in the player's brain and takes the readings of the sensor as a measurement of enjoyment. In their test game (multiplayer competitive third person shooting game), the experiment showed that EEG-based DDA greatly improved the enjoyment level of the players.

However, the author concludes that the hardware of the EEG headset needs to be improved for the DDA system to function properly in commercial games.

3 MINECRAFT

Minecraft is a sandbox video game where players are put in a 3D fantasy world made of blocks (e.g., stone and sand) and entities (e.g., friendly animals and hostile mobs). Overall, there are three popular gameplay modes in Minecraft – survival, creative, and minigames. In classic survival gameplay, which is our focus, players can destroy, build, and craft blocks to build their own constructions and, therefore, reshape the world. They can fight hostile mobs and then gather valuable resources dropped by the mobs. It is worth noting that it is not very difficult to survive for most players, despite being called "survival". Apart from the classic survival gameplay, there are countless gameplay possibilities arising from both the existing game mechanics and the extensibility of Minecraft for programmers.

Minecraft comes with single player and multiplayer gameplay. Single player means that there is only one player in the fantasy world and the rest of the things in the world are environmental, such as those that are mobile (e.g., animals, mobs) and those that are not mobile (e.g., plants, blocks). In single player, the player can change the game difficulty. Multiplayer has the same elements as single player, plus there can be more than one player in the fantasy world. Interactions can take place between players, such as building a small town and fighting hostile mobs together. There are no official public multiplayer servers for Minecraft. So, the multiplayer game of Minecraft usually needs a person to be the host, who takes charge of the creation of the server, including game rule formulation and gameplay customisation. Due to this freedom of customisation, there are many public Minecraft servers of diverse gameplay for players to choose and play, including those servers with survival and MMORPG gameplay. In multiplayer, it is generally not allowed to change the difficulty [5], so the servers need to automatically adjust the difficulty for players.

4 DDA SYSTEM AND GAME DESIGN

In this section, we describe our Multiplayer Minecraft DDA framework and our game design. We present our Minecraft DDA system as a framework that allows developers to implement the system in an expandable way. For game design, we describe the case study used in this research to populate our framework for our target scenarios. We focus on servers with survival and massively multiplayer online role-playing game (MMORPG) gameplay. Survival[2] means that players must collect resources, build structures, battle mobs, eat, and explore the world in an effort to survive and then thrive. MMORPG means that each player controls their own character and takes over many actions of the character in a fantasy world. The fantasy world is hosted by a server and continues to exist and evolve even when the player is offline. There are three main assumptions for our gameplay: 1) growth - player characters can become stronger over time, 2) cooperation - gameplay is cooperative and players play as a party, and 3) non-player enemies - all enemies are controlled by algorithms and actively search for and chase players. Our DDA framework is based on the work of Chowdhury and Katchabaw [2]. The key components of our DDA framework are data collection, deciding when to adjust, and deciding how to adjust.

[2]https://minecraft.fandom.com/wiki/Survival

4.1 Data Collection

DDA systems adjust the game difficulty to be harder or easier based on some indication that players are experiencing the game to be too easy or too hard. Data is the basis of decisions for our DDA system. We define the game data that the DDA system needs to collect as follows:

- Heuristics of player performance: accuracy with which the player uses their bow to shoot arrows, player success rate for blocking incoming arrows, number of times that the player has died, and player skill levels in different abilities.
- Timestamp when mobs were last damaged by the player (prevents the DDA system from adjusting the mobs that were just damaged by the player within a specific duration, to avoid player frustration/confusion if mob strength is adjusted during a fight).

4.2 When to Adjust

DDA systems need to detect when to adjust the game difficulty, usually when players experience the game to be too hard or too easy. Our DDA system tests for the following three cases for when to adjust non-player enemies (i.e., mobs) and notifies the rest of the system that an adjustment is needed:

- When any player dies, the DDA system is notified to adjust all mobs around the player. When the player dies, they are likely to feel that the game is difficult to some degree. Death in survival gameplay is also an indication of failure. Adjusting the game when the player dies can help ensure the game is not discouragingly hard for the player.
- When a mob spawns (i.e., an enemy is created and placed near the player), the DDA system is notified to adjust the mob that just spawned. The spawned mob does not have information about how strong it should be, so the DDA system needs to determine its strength.
- Periodically, at every 100 ticks (5 seconds), the DDA system is notified. There are many dynamic states for the player's performance (e.g., shooting accuracy, levels in the skill plugin). As these states can be updated very frequently, it is more efficient to adjust mobs periodically, rather than on every state change.

4.3 How to Adjust

If any of the 'when to adjust' cases are met, an adjustment plan (i.e., 'how to adjust') is executed. Overall, the adjustment plan is based on the player's performance. When the player's performance is known, the strength of mobs is computed accordingly. This can be seen as a mapping from the player's performance to the strength of mobs. Our adjustment plans involve three steps: 1) quantifying player's performance, 2) quantifying strength of mobs, and 3) mapping player's performance to strength of mobs.

The player's performance is first quantified using the identified heuristics for player performance, including player ability levels (representing their strength), their overall ability to survive, and their proficiency in using ranged weapons and blocking arrows. To quantify the strength of mobs, we use an open-source project LevelledMobs (LM)[3], which provides an interface and sophisticated system for mob levels and levelling. LM implements the level-based strength of mobs by varying the attributes of mobs based on a given level. The final step executes an algorithm that maps the player's performance score to the strength of mobs. The key problem is determining the level of each mob when there are multiple players around the mob. To solve this problem, the algorithm calculates the levels of mobs based on the

[3]https://github.com/lokka30/LevelledMobs

distance between mobs and players. The algorithm loops through the mobs and adjusts the mobs based on players within a certain range. The range is set to 64 blocks, which is the furthest distance a player can see mobs in our test Minecraft server. It computes the distance between each player and the mob and stores the results in a map-like data structure. The contribution weight for each player is computed based on the distances, using a Gaussian function. As players pay more attention to nearby mobs, mob level is weighted more by closer players. Weighted average level is computed based on the contribution weight. The final step first checks whether the mob has recently been damaged. If so, the algorithm terminates without modifying the level. If not, it adds the computed level to the mob, as well as a small random value to create some randomness.

5 PLAYER EVALUATION STUDY

We developed a prototype of a configurable DDA system according to our framework. To investigate the affects of the DDA system on players, we designed a player evaluation study. We ran an online experiment with 23 participants, who were recruited via word of mouth. Our experiment consisted of three parts, with each part requiring participants to complete an in-game quest, followed by a questionnaire. The entire session lasted 30 minutes per participant. In this section, we present the design of our player evaluation study.

5.1 Minecraft Server

We hosted a test Minecraft server and installed the DDA system on the server. The experiment took place online in our test Minecraft server. In order to reduce issues related to high bandwidth (transmitting many moving objects e.g., mobs) and low latency (time-sensitive reactions e.g., blocking incoming actions), we recruited participants from a small Minecraft server community named Mewcraft[4]. A total of 23 people participated in the experiment. All participants had previously played Minecraft, so were able to get started quickly and were representative of Minecraft players.

5.2 Scenarios

We developed three test scenarios to evaluate the player experience for three different configurations of the DDA system. During the experiment session, the player completed the same quest three times, each with a different DDA system configuration (i.e., scenario). The order of play was as follows:

- Scenario 1: the DDA system was disabled, so that the difficulty of mobs was static during the scenario.
- Scenario 2: the DDA system was enabled, but only adjusted mobs once immediately after spawning, so mob difficulty was determined by nearby players when the mob spawned and then remain unchanged.
- Scenario 3: same as Scenario 2, plus the DDA system adjusted mobs at certain intervals, so mob difficulty changed if nearby players became weaker or stronger.

5.3 In-Game Quest

Full survival gameplay in Minecraft usually lasts for hours or days. In order to reduce game duration for our experiments, we created a quest within our DDA system to simulate the core game experience in our target Minecraft servers. As our DDA system was dedicated to adjusting mobs, the quest was focused on killing mobs. Two players were required to simulate a multiplayer environment. The two players played together to kill a certain number of mobs to complete the quest. Each quest included three phases: 1) starting, 2) fighting, and 3) closing.

[4]https://www.mimaru.me/

- Starting Phase: the players joined the test Minecraft server. The players encountered a named Guide (NPC: non-player character), who told gave them the quest (i.e., told them what to do). A named Trader (NPC) acted as an equipment shop, giving the players a way to upgrade their equipment during the quest.
- Fighting Phase: the DDA system took effect. The three configurations (scenarios) were were used in this phase. In this phase, the players needed to kill two different waves of mobs, while the DDA system adjusted the mobs in real-time. The players could become stronger through killing mobs. The first wave of mobs was Zombies, which can only deal damage (using their fists) to a nearby opponent. The second wave of mobs was Skeletons, which can deal damage (by shooting arrows) to an opponent at a distance. There was a halftime between the two waves of mobs, in which the players could upgrade their equipment at the NPC Trader. The second wave of mobs were harder to deal with by design.
- Closing Phase: all mobs were cleared by the players. The participants were then asked to complete a questionnaire about play experience, while the game states were reset, ready for the next scenario, until all three scenarios were completed.

5.4 Questionnaire

The online questionnaire included 11 questions related to game experience, which the player completed following each scenario (i.e., three times). The questions were designed to provide insight into the player experience for further refinement of the DDA system. The questions related to the overall game enjoyment, difficulty, and enemy levels, as well as the difficulty and levels of enemies during different phases of the fighting.

6 RESULTS AND DISCUSSION

In this section, we present our questionnaire results and provide a discussion of the results. The participants rated their enjoyment similarly between the three scenarios, with Scenario 3 (4.3/5) achieving a slightly higher average rating than Scenarios 1 and 2 (4.2/5). No participants rated the experience as "not enjoyable" for any scenario, which could be due to the pace of the quests (killing mobs) being much faster than usual survival gameplay. As such, the game required substantial player concentration. There was also guidance telling the players what to do next during the quest, whereas in Minecraft survival gameplay, players do not have a set goal or guidance of what to do. Both concentration and clear goals are core elements of GameFlow [16].

The participant ratings for difficulty were more distinct across the three scenarios, compared to enjoyment. The average difficulty rating was 2.0 ("somewhat easy") for Scenario 1, 2.8/5 (almost "neutral") for Scenario 2, and 3.1/5 (above "neutral") for Scenario 3. No players rated any scenarios as "extremely hard", while some players rated Scenario 2 (22%) and Scenario 3 (32%) as "somewhat hard". During the quest, players could experience difficulty for different reasons (e.g., dying while fighting strong mobs). In the scenarios where the DDA system was active, the rated difficulty was higher.

Participants paid increasing attention to the level of mobs from Scenario 1 (2.0/3 or "often") to Scenario 2 (2.5/3 or between "often" and "always") to Scenario 3 (2.8/3 or almost "always"). Level of mobs were displayed above their heads (see Figure 1). We believe that this reflects the participants' increasing awareness of both the DDA system and of increasingly higher-level mobs throughout the scenarios.

Participant responses to what they were thinking about when reading the levels of mobs changed throughout the scenarios. In Scenario 1, most common responses were judging mob strength (10) and determining their match for the mob (9), as well as "no idea" (10). In Scenario 2, most common responses were judging mob strength (18), determining

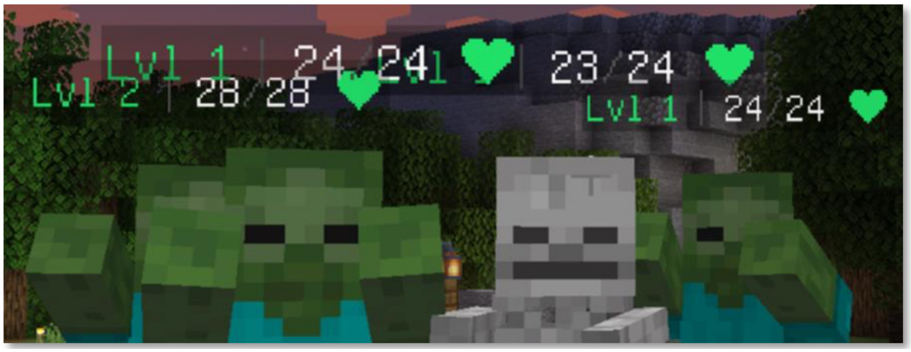

Fig. 1. Screenshot of mobs in the quest during fighting. Levels of mobs and health points (current/max) are shown above their heads.

their match for the mob (16), and estimating rewards (10). In Scenario 3, most common responses were judging mob strength (14), hoping mob strength didn't change (13), and determining their match for the mob (10). In Scenario 3, the DDA system adjusted mobs at certain intervals, so it was possible that players would see the levels of mobs change suddenly. The Scenario 3 responses indicate that the participants did not want the mobs to suddenly become stronger or weaker while fighting them. If a player was in a situation where they were about to kill a mob and it suddenly leveled up, the player could feel that their work was lost.

When asked about the first wave of mobs (zombies), participants indicated they increased in difficulty from Scenarios 1 to 3 for both the first half and second half of the wave. Average responses ranged from around "easy to defeat" (2.0/5 and 2.1/5 for first and second waves) for Scenario 1 to around "just right" (2.9/5 and 3.3/5) for Scenario 3. For the second half of the wave for Scenario 3, 35% of participants indicated they were "hard to defeat". Participants were also asked about the zombie levels, which increased from Scenario 1 to Scenarios 2 and 3, with 2.0/3 indicating "levels were just right". In Scenario 1, all mobs were the same level, as the DDA system was disabled. The DDA-enabled scenarios seemed to provide a better skill match for our participants by this measure.

During the break between the two waves of mobs, the majority of players took the opportunity to upgrade their equipment from the NPC Trader, increasing from Scenario 1 (52%) to Scenarios 2 (95%) and 3 (85%). Upgrading equipment helped the players to defeat the second wave of mobs (skeletons) with less effort. Checking whether players upgraded their equipment between the waves provided more insight into the player experience for the second wave. It is likely that many players were not aware of this option during the first playthrough (Scenario 1).

When asked about the second wave of mobs (skeletons), participants indicated they increased in difficulty from Scenarios 1 to 3 for both the first half and second half of the wave, similar to the first wave of zombies. Average responses for the first half of the wave ranged from around "easy to defeat" (2.0/5) for Scenario 1 to around "just right" (3.1/5) for Scenario 3. For the second half of the wave, perceived difficulty ranged from between "easy" to "just right" (2.4/5) for Scenario 1 up to "hard to defeat" (4.0/5) for Scenario 3, with 85% of participants indicating they were "hard" or "too hard" to defeat in Scenario 3. Participants were also asked about the skeleton levels, which increased slightly from Scenario 1 to Scenarios 3, but stayed around the "just right" level. However, 25% of participants did indicate that levels were "too high" in Scenario 3. Although the second waves of mobs (skeletons) were harder to fight by design, these results indicate that the DDA system might need some further refinement to decrease the difficulty of skeletons.

Despite the perceived increase in difficulty over the three scenarios, the rated game enjoyment remained high. This could indicate that the players enjoyed the increasing difficulty, or that the changes had no impact on their enjoyment. Participants experienced Scenario 3 to be of "neutral" difficulty overall (not too easy or too hard), despite higher ratings of the difficulty of each successive wave of mobs during this scenario. Our evaluation enabled us to put our DDA framework to the test and provided some initial insights into how effective and enjoyable our DDA system was for our player sample in multiplayer Minecraft. Limitations and future work are outlined in the following section.

6.1 Limitations and Future Work

This study constitutes an initial test of our proposed multiplayer Minecraft DDA framework. There are many limitations to our design and evaluation so far, as well as many future opportunities for employing and testing our framework. First, our approach only focuses on the difficulty of mobs. In contrast, the difficulty in a real Minecraft server with survival and MMORPG gameplay not only includes mobs, but could also include other aspects of the server gameplay, such as economy (e.g., trading items with other players) and construction (e.g., collecting resources and building towns). Second, we employed a simple questionnaire in our experiment to collect player opinions and experiences. However, future research could include both an expanded questionnaire and more extensive analysis, as well as more objective player observations, such as in-game metrics and physiological measures. Third, our experimental design could be improved in future to include a larger set of players, longer play sessions, more players per session (i.e., multiplayer beyond two players), and mixed order of scenarios. We also note that the questions we asked in earlier scenarios could have influenced how players perceived and acted in later scenarios (e.g., creating an awareness of mobs with changing levels).

Based on the limitations of our preliminary DDA system design and experimentation, future work could explore various new directions. First, our approach focuses on the difficulty of mobs, which benefits players who enjoy the fighting aspect of Minecraft. There are many Minecraft players who prefer to be a reputable merchant, to build complex constructions, or to simply socialise in Minecraft servers. In the future, we plan to design a more sophisticated DDA system for players who enjoy construction, which constitutes a large player base. The concept of difficulty for these players potentially includes the process of collecting resources and trimming the terrain. Second, we plan to incorporate the logging of player data in future experiments. By combining player data with questionnaire responses, we hope to gain deeper insight into player decisions, actions, and experience.

7 CONCLUSIONS

In this paper, we presented a new approach to developing DDA systems for public Minecraft servers, in the form of a proposed Multiplayer Minecraft DDA Framework. We developed a test DDA system following our proposed framework and developed test scenarios to evaluate different configurations of our DDA system via an experiment with Minecraft players. Our DDA system focused on adjusting mobs around players, to attempt to improve the experience of all players by better matching mob levels to the collective skill levels of players within a local area. Although the results of our experiment are preliminary and further experimentation is needed, our results indicate that the resulting game experience was enjoyable and that players were aware of increasing difficulty level and harder mobs. Our framework provides a suitable and usable approach to developing a DDA system for multiplayer Minecraft. Future research can make use of our framework to further develop multiplayer Minecraft DDA systems for research and general play. Our DDA system can be easily tuned, extended, and customised, which lends itself to further experimentation and customisation of different configurations and scenarios.

REFERENCES

[1] Alexander Baldwin, Daniel M. Johnson, Peta Wyeth, and Penelope Sweetser. 2013. A framework of Dynamic Difficulty Adjustment in competitive multiplayer video games. *2013 IEEE International Games Innovation Conference (IGIC)* (2013), 16–19.

[2] M. Chowdhury and M. Katchabaw. 2013. Bringing auto dynamic difficulty to commercial games: A reusable design pattern based approach. In *2013 18th International Conference on Computer Games: AI, Animation, Mobile, Interactive Multimedia, Educational Serious Games (CGAMES)*. IEEE Computer Society, Los Alamitos, CA, USA, 103–110. https://doi.org/10.1109/CGames.2013.6632615

[3] Mihaly Csikszentmihalyi. 1991. *Flow: The Psychology of Optimal Experience*. Harper Perennial, New York.

[4] Robin Hunicke. 2005. The Case for Dynamic Difficulty Adjustment in Games. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology* (Valencia, Spain) *(ACE '05)*. Association for Computing Machinery, New York, NY, USA, 429–433. https://doi.org/10.1145/1178477.1178573

[5] H. Chad Lane and Sherry Yi. 2017. Chapter 7 - Playing With Virtual Blocks: Minecraft as a Learning Environment for Practice and Research. In *Cognitive Development in Digital Contexts*, Fran C. Blumberg and Patricia J. Brooks (Eds.). Academic Press, San Diego, 145–166. https://doi.org/10.1016/B978-0-12-809481-5.00007-9

[6] Daniel Lee, Gopi Krishnan Rajbahadur, Dayi Lin, Mohammed Sayagh, Cor-Paul Bezemer, and Ahmed E. Hassan. 2020. An Empirical Study of the Characteristics of Popular Minecraft Mods. *Empirical Softw. Engg.* 25, 5 (sep 2020), 3396–3429. https://doi.org/10.1007/s10664-020-09840-9

[7] Adi Stein, Yair Yotam, Rami Puzis, Guy Shani, and Meirav Taieb-Maimon. 2018. EEG-triggered dynamic difficulty adjustment for multiplayer games. *Entertainment Computing* 25 (2018), 14–25. https://doi.org/10.1016/j.entcom.2017.11.003

[8] Penny Sweetser. 2020. GameFlow 2020: 15 Years of a Model of Player Enjoyment. In *Proceedings of the 32nd Australian Conference on Human-Computer-Interaction* (Sydney, NSW, Australia) *(OZCHI'20)*. ACM, New York, NY, USA.

[9] Penny Sweetser and Daniel Johnson. 2019. Evaluating the GameFlow Model with Different Stakeholders. In *Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts* (Barcelona, Spain) *(CHI PLAY '19 Extended Abstracts)*. ACM, New York, NY, USA, 697–703. https://doi.org/10.1145/3341215.3356286

[10] Penny Sweetser and Daniel Johnson. 2019. GameFlow and Player Experience Measures: An Initial Comparison of Conceptual Constructs. In *Proceedings of the 31st Australian Conference on Human-Computer-Interaction* (Fremantle, WA, Australia) *(OZCHI'19)*. ACM, New York, NY, USA, 317–321. https://doi.org/10.1145/3369457.3369486

[11] Penny Sweetser, Daniel Johnson, and Jay Kyburz. 2020. Evaluating GameFlow in a Multiplayer Online Strategy Game Under Development. In *Proceedings of the Australasian Computer Science Week Multiconference* (Melbourne, VIC, Australia) *(ACSW '20)*. ACM, New York, NY, USA, Article 47, 10 pages. https://doi.org/10.1145/3373017.3373068

[12] Penelope Sweetser, Daniel Johnson, and Peta Wyeth. 2013. Revisiting the GameFlow Model with Detailed Heuristics. *The Journal of Creative Technologies* 3 (May 2013). https://ojs.aut.ac.nz/journal-of-creative-technologies/index.php/JCT/article/view/16

[13] Penelope Sweetser, Daniel Johnson, Peta Wyeth, Aiman Anwar, Yan Meng, and Anne Ozdowska. 2017. GameFlow in Different Game Genres and Platforms. *Comput. Entertain.* 15, 3, Article 1 (April 2017), 24 pages. https://doi.org/10.1145/3034780

[14] Penelope Sweetser, Daniel Johnson, Peta Wyeth, and Anne Ozdowska. 2012. GameFlow Heuristics for Designing and Evaluating Real-Time Strategy Games. In *Proceedings of The 8th Australasian Conference on Interactive Entertainment: Playing the System* (Auckland, New Zealand) *(IE '12)*. ACM, New York, NY, USA, Article 1, 10 pages. https://doi.org/10.1145/2336727.2336728

[15] Penny Sweetser, Zane Rogalewicz, and Qingyang Li. 2019. Understanding Enjoyment in VR Games with GameFlow. In *25th ACM Symposium on Virtual Reality Software and Technology* (Parramatta, NSW, Australia) *(VRST '19)*. ACM, New York, NY, USA, Article 96, 2 pages. https://doi.org/10.1145/3359996.3364800

[16] Penelope Sweetser and Peta Wyeth. 2005. GameFlow: A Model for Evaluating Player Enjoyment in Games. *Comput. Entertain.* 3, 3 (July 2005), 3. https://doi.org/10.1145/1077246.1077253

[17] Su Xue, Meng Wu, John Kolen, Navid Aghdaie, and Kazi A. Zaman. 2017. Dynamic Difficulty Adjustment for Maximized Engagement in Digital Games. In *Proceedings of the 26th International Conference on World Wide Web Companion* (Perth, Australia) *(WWW '17 Companion)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 465–471. https://doi.org/10.1145/3041021.3054170

[18] Mohammad Zohaib and Hideyuki Nakanishi. 2018. Dynamic Difficulty Adjustment (DDA) in Computer Games: A Review. *Adv. in Hum.-Comp. Int.* 2018 (jan 2018), 12 pages. https://doi.org/10.1155/2018/5681652

[19] Alexander Zook and Mark Riedl. 2021. A Temporal Data-Driven Player Model for Dynamic Difficulty Adjustment. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* 8, 1 (Jun. 2021), 93–98. https://ojs.aaai.org/index.php/AIIDE/article/view/12504