applied sciences

MDPI

Article

The Use of Deep Learning to Improve Player Engagement in a Video Game through a Dynamic Difficulty Adjustment Based on Skills Classification

Edwin A. Romero-Mendez [1], Pedro C. Santana-Mancilla [1], Miguel Garcia-Ruiz [2], Osval A. Montesinos-López [1] and Luis E. Anido-Rifón [3,*]

[1] School of Telematics, Universidad de Colima, Colima 28040, Mexico
[2] School of Computer Science and Technology, Algoma University, Sault Ste. Marie, ON P6A 2G4, Canada
[3] atlanTTic Research Center, School of Telecommunications Engineering, University of Vigo, 36310 Vigo, Spain
* Correspondence: lanido@det.uvigo.es

Abstract: The balance between game difficulty and player skill in the evolving landscape of the video game industry is a significant factor in player engagement. This study introduces a deep learning (DL) approach to enhance gameplay by dynamically adjusting game difficulty based on a player's skill level. Our methodology aims to prevent player disengagement, which can occur if the game difficulty significantly exceeds or falls short of the player's skill level. Our evaluation indicates that such dynamic adjustment leads to improved gameplay and increased player involvement, with 90% of the players reporting high game enjoyment and immersion levels.

Keywords: dynamic difficulty adjustment; deep learning; player engagement

check for updates

Citation: Romero-Mendez, E.A.; Santana-Mancilla, P.C.; Garcia-Ruiz, M.; Montesinos-López, O.A.; Anido-Rifón, L.E. The Use of Deep Learning to Improve Player Engagement in a Video Game through a Dynamic Difficulty Adjustment Based on Skills Classification. *Appl. Sci.* **2023**, *13*, 8249. https://doi.org/10.3390/app13148249

Academic Editor: João M. F. Rodrigues

Received: 15 March 2023
Revised: 11 July 2023
Accepted: 14 July 2023
Published: 16 July 2023

1. Introduction

Since their origins, video games and their associated industries have continued to evolve, limited only by developers' creativity and cutting-edge technologies. These technological advances include artificial intelligence (AI), a growing field that began in the 1950s when Alan Turing questioned whether machines could think [1].

It is known that computers do not think like humans but can use learning techniques to learn to perform tasks autonomously. This action is known as machine learning (ML), a category of AI that provides the ability to learn based on data (datasets) through various learning strategies to classify or predict outcomes by simulating human behavior before events occur [2]. ML focuses on developing systems that collect data, identify patterns, and learn from different scenarios for decision making. For example, in the case of a video game, its behavior can be improved autonomously without explicit programming [3,4].

There are various approaches to ML, and deep learning (DL) is one of the most popular approaches. DL models differ from most ML methods in that human brain functions motivate their operation by mimicking neuronal networks. For this reason, artificial neural networks (ANNs) are described as a set of simple (often adaptive) segments interconnected in massive parallel and hierarchically organized to interact with real-world objects in the same way as the human nervous system. Therefore, DL is defined as a generalization of an ANN in which multiple hidden layers are used, meaning that more neurons are used to implement the model. Deep learning has been successfully applied in many domains [5,6], including gaming [7].

Two essential and fundamental aspects of the video game industry are game development and user experience. The development of a game involves the technologies and platforms available to create the game, and the user experience is the tangible events perceived by the player supported by the overall game–player interaction, including how satisfactory the player experience is [8].

The concept of flow [9,10], a psychological state in which individuals become fully engaged in an activity, experiencing a sense of control and enjoyment, plays a pivotal role in understanding player–game interaction dynamics. This state is closely associated with optimal experiences in gameplay, leading to increased player immersion, satisfaction, and, ultimately, retention [11]. Thus, accounting for flow becomes crucial as it provides key insights into how to enhance the gaming experience, making it an integral part of our study. Cowley et al. [12] described the relationship between the player and the game, characterized by learning and enjoyment during the game, which is fundamental for analyzing the player's experience in front of a video game. They focus on improving the understanding of how gamers interact with games, thus providing valuable information to video game researchers who aim to design immersive games for every user.

As technology advances, video game players demand an increasing number of good user experiences (UX). Therefore, video game difficulty has become a valuable indicator to keep gamers engaged, as games that are easy for an expert gamer can become tedious, while games that are difficult for a novice player can frustrate them, and in both cases, the player will stop playing the game [13]. Similarly, it has been found that once players understand the game pattern and discover their weaknesses, the video game is perceived as boring, uninteresting, and lacking in challenges [14].

Therefore, according to the game flow, there is a natural relationship between the game's difficulty and the player's skill [11]. This study proposes a deep learning-based approach to improve player engagement through two steps: (1) skill-based classification and (2) dynamic difficulty adjustment (DDA). Dynamic difficulty adjustment is a method that automatically modifies video game behavior and features in real-time based on a player's skills [7].

The main contribution of our work is twofold:

Implementing a dynamic difficulty adjustment system in a video game. This system classifies players in real-time based on their skill level, allowing difficulty adjustment that aligns with the player's experience. In this way, the game provides challenges proportional to the players' skills, improving their engagement and immersion.

Finally, our DL model was proven to be effective in improving the gaming experience and obtaining a high degree of accuracy in classifying the player's skill level. This effectiveness leads to an improved personalized game experience, marking an important step forward in player-centered video game design.

2. Related Works

The application of DL techniques to dynamically adjust the complexity of video games has become an increasingly popular research topic in recent years. This approach seeks to enhance player engagement and enjoyment by creating an adaptive gaming experience that challenges players without overwhelming their skills or knowledge. In this section, we present some of the most relevant works in this area, ranging from using reinforcement learning algorithms to training intelligent agents in specific games to implementing neural networks that can learn from player metrics and dynamically adjust the difficulty level of the game.

Porssut et al. [15] proposed an intelligent agent that trains a virtual reality (VR) engine using adaptive and reinforcement learning methods. The purpose of this engine is to adjust the visual effects of the response depending on the executed movements to improve the robot's reactions, increase the participant's performance limits, or correct their errors in the executions.

Comi [16] discussed the use of artificial intelligence in video games using deep reinforced learning techniques. This article explains the techniques used to create an intelligent agent that can learn from its environment for the game Snake and achieve better results with each interaction with the game.

The creators of the video game Unreal Tournament$^{®}$ [17] allow open access to cognitive computing developers to create an intelligent agent that simulates the reactions that a human player could have. This opens many possibilities for the video game industry because they use accurate data to assist developers in the construction of new, more competitive, and, most importantly, different and adaptable games. In [18], an intelligent agent was implemented in Unreal Tournament® that can execute the shots and actions that a player can perform to learn from these events and surpass the player's performance.

As described in [19], intelligent agents contain FALCON, a neural network that exercises reinforcement learning, which measures the results as the game environment changes or makes real-time predictions for the acquisition of new and improved skills. It has been demonstrated that an intelligent agent constantly learns, and its learning scales with each iteration it passes through.

Ismail et al. [20] showcase how reinforcement learning is implemented, evaluating its results and categorizing them into three primary motivations. The goal is to create intelligent agents that can be implemented as non-playable video game characters to identify and solve performance or maintenance tasks. Next, two new metrics are proposed to evaluate and compare agents using these methods to better describe and differentiate character behavior according to their motivation to learn different tasks. These metrics quantify the focus of attention and the duration of the agents' stay. Finally, an empirical evaluation of reinforcement learning agents controlling characters in a simulated game scenario was conducted, comparing the effects of the three motivations on learning achievement and maintenance tasks.

In [21], recent advances in DL were discussed in the context of how they have been applied to play different types of video games, such as first-person shooters, arcade games, and real-time strategy games. The unique requirements that different genres of games pose to a deep learning system are analyzed, and important open challenges are highlighted in the context of applying these machine learning methods to video games. These challenges include dealing with huge decision spaces, sparse rewards, difficulty alterations, and even the creation of new objectives.

3. Materials and Methods

This research consists of designing and developing a video game to evaluate whether dynamic complexity adjustment of a video game can be achieved by identifying a player's skills using a deep learning model. This paper also reports on the design, training, and evaluation of the intelligent model and the assessment of the player's experience.

A project-based methodology [22] was used to develop this study. The methodology was as follows.

- Design and create a video game to train and test the DL model;
- Generate a dataset to train the model from the player event log;
- Train and validate a DL model appropriate to the game design created;
- Integrate the DL model in the video game;
- Validate the user experience.

Figure 1 shows a flowchart of the general process of this methodology, in which we appreciate the integration of the training and learning phases of the DL model.

3.1. Astro the Video game

A game called Astro was developed (Figure 2). It was inspired by the Space Invaders© [23] video game, and its gameplay consists of the player controlling a spaceship and, during its journey, facing scenarios of alien ships blocking their path, which must be eliminated to continue with the next level of the game. Once the entire fleet of alien ships has been eliminated, the game displays the same number of alien ships on the screen, but with faster scrolling according to the level being played. Astro development was performed in Python programming language using the Pygame library [24].

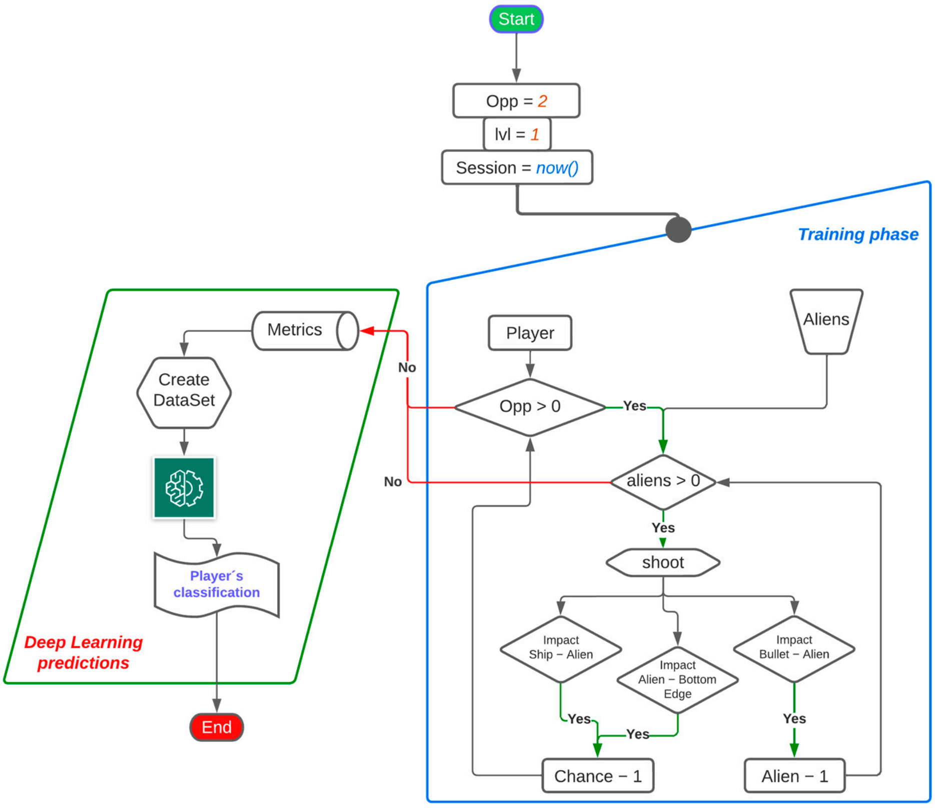


Figure 1. Methodology flowchart.

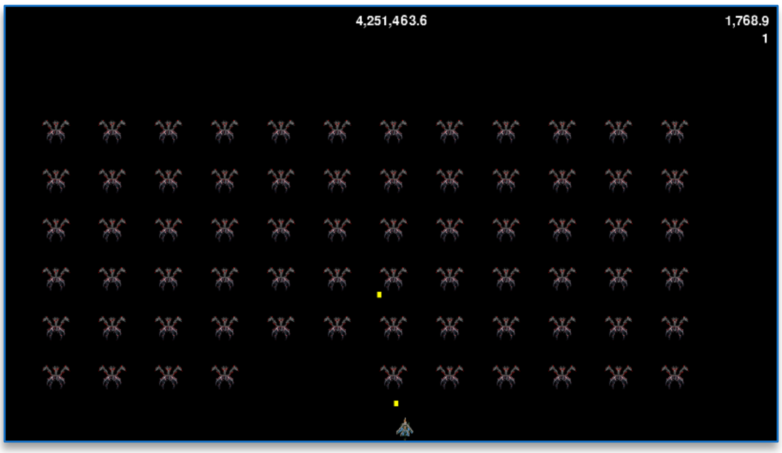


Figure 2. The video game Astro.

The functional requirements that Astro must meet are:

- The player can control a spaceship and move on its four axes;
- The game must create a fleet of enemy ships;
- The player-controlled ship will have the ability to shoot at enemy ships;
- Enemy ships can zigzag horizontally and vertically from the top of the screen to the bottom or collide with a player's ship;
- The player will have two opportunities to beat the game;
    a. An opportunity is lost every time the player's ship collides with an alien or the alien float reaches the bottom of the screen.
- The game must end when the player's opportunities have been finished;
- The game must collect metrics and group them in a dataset per session.

3.2. Dataset Creation

Classifying a player's abilities while playing the game requires data-driven DL model training.

It is necessary to first collect the necessary data for the training phase to utilize DL techniques to analyze player metrics. This process involves gathering information on various aspects of player behavior, including player movement, interactions with game objects, and the time spent in different areas of the game.

After analyzing the various variables and metrics that can be considered to classify a player in a video game like Astro, the following dataset (see Table 1) was designed (variable names are mainly based on their Spanish name).

Table 1. Variables selected to build the dataset.

Variable	Description
px	Horizontal position of the player's ship on the screen.
py	Vertical position of the player's ship on the screen.
nb	Bullet number in contact with an alien.
mizq	Number of movements to the left.
marb	Number of moves up.
mabj	Number of moves down.
mder	Number of moves to the right.
score	Score.
ax	Horizontal position of the alien on the screen.
ay	Vertical position of the alien on the screen.
di	Distance between the alien and the ship at the bullet's impact.
v	Remaining opportunities.
lvl	Game level.
acc	Shot accuracy.
hit	An alien was impacted.

With the variable selection finished, we generated a dataset from 148 gaming sessions carried out by 56 players using Astro (see Figure 3), with a total of 32,280 data records in the dataset.

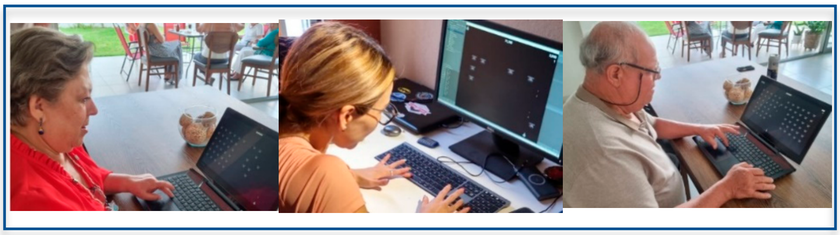


Figure 3. Game session sample for dataset creation.

Table 2 provides an overview of the players sampled to generate the dataset for the deep learning model to identify player levels in the video game. In addition, the table presents the players' age ranges and number of players in each category.

Table 2. Players' demographic data for the dataset creation.

Category	Age Range	Number of Players
Adolescents	13–18	16
Young adults	19–25	9
Adults	26–59	28
Seniors	>59	3

The players were categorized based on their age ranges: adolescents (13–18 years old), young adults (19–25 years old), adults (26–59 years old), and seniors (above 59 years). The sample included 16 players in the "Adolescents" category, nine players in the "Young Adults" category, 28 players in the "Adults" category, and three players in the "Seniors" category.

Table 2 presents the age range and number of players in each category sampled to generate the dataset for the deep learning model. The table provides a quick and clear overview of the sample composition and insights into the age distribution of the players in the dataset.

3.3. Model Training

Once the player metrics have been recollected during the training phase, a model can be trained using DL techniques to apply it to the obtained metrics. Finally, the game should evaluate the events performed by the players and classify their level of familiarity with the game into three categories: beginner, intermediate, and expert. By categorizing players according to their expertise levels, gameplay mechanics can be adjusted to provide a better experience for all players.

Figure 4 illustrates the process of utilizing DL techniques to analyze player metrics. This process is divided into three main paths.

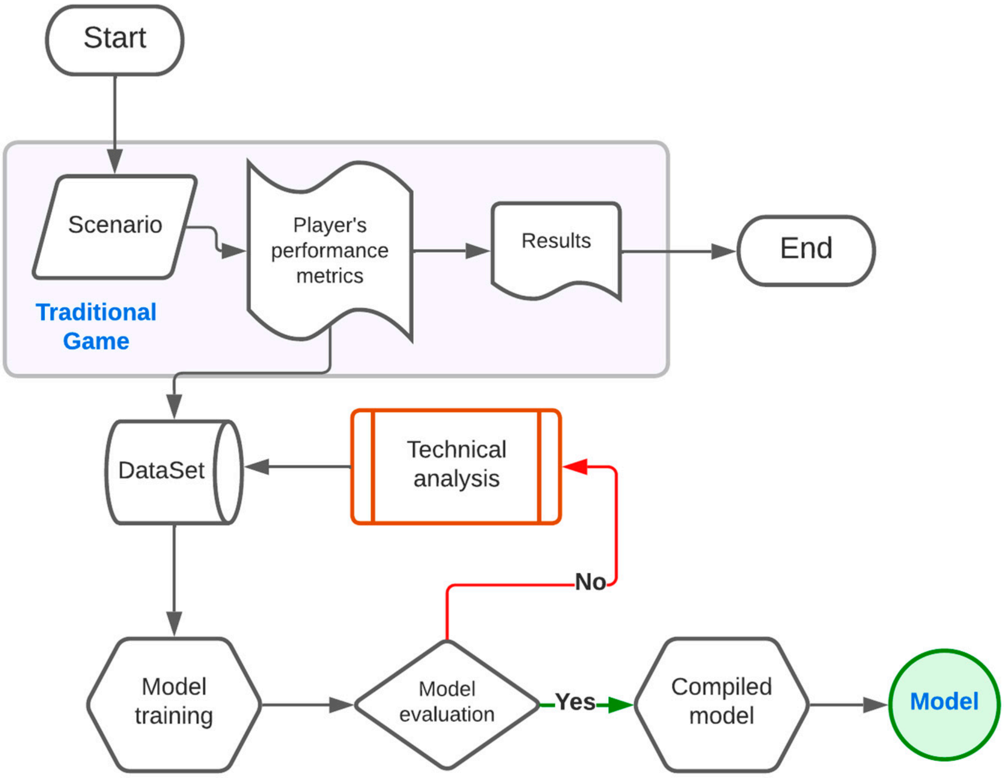


Figure 4. Data collection and training flow in Astro.

Path 1 represents the traditional method for evaluating a player's performance in a game scenario. First, the players' performance metrics were measured, and the results were obtained based on these metrics.

Path 2 represents the data collection and training processes. First, the player plays the scenario, and the performance metrics are collected to create a dataset. The dataset was then used to train a deep learning model. Finally, model evaluation was conducted, and if the model performance was satisfactory, it was compiled and used for further analysis.

Path 3 represents an alternative approach when model evaluation results are unsatisfactory. In this case, technical analysis was conducted to identify the shortcomings of the model. The dataset was then further refined based on this analysis, and the model training and evaluation process was repeated until satisfactory results were obtained. The compiled model was then used for further analysis.

The process involves data collection, model training, and evaluation, and may require multiple iterations to achieve satisfactory results.

Deep learning has shown great potential for creating models that can adapt to the changing skill levels of players in a game. By analyzing player metrics, such as the bullet number in contact with an alien, accuracy, and game progress, neural nodes can be trained to dynamically adjust the complexity of the game to enhance the player's experience and provide a personalized gameplay experience that caters to the individual's skill level.

Once the model is validated, it must be implemented in a game. Figure 5 illustrates how integrating the trained model into Astro expedites the game level adjustment process. The diagram begins with the initial level of the game.

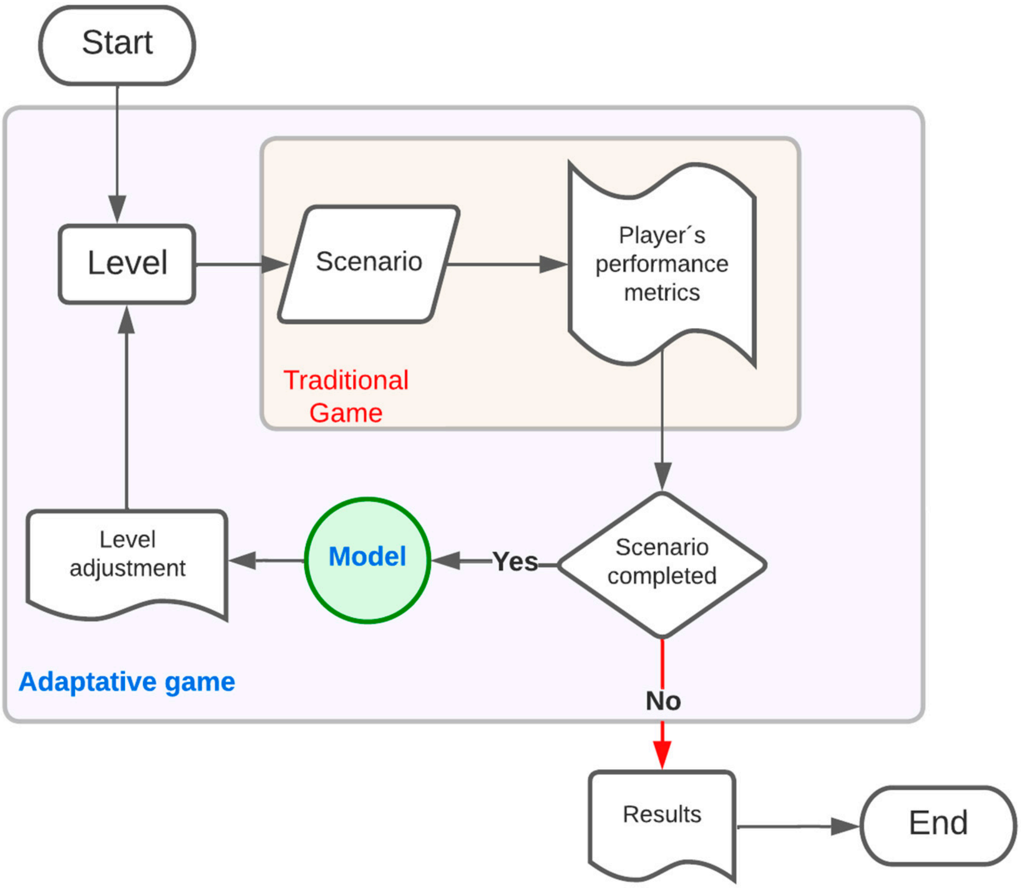


Figure 5. Astro game flow with dynamic complexity adjustment.

As the player proceeded through the game scenario from the initial game level, their performance metrics were recorded. If the player completes the scenario successfully, the process advances to the "Scenario completed (Yes)" point.

At this juncture, the trained model leveraged the player's performance metrics to calibrate the game level. Subsequently, the game presents an adjusted game level scenario for the player, and the performance metrics are measured again. If a player succeeds in the scenario, the model further tailors the game level. This cycle continues until the player can no longer complete the scenario, and the process proceeds to the "Scenario completed (No)" point.

The game level adjustment process concludes by presenting the outcomes to players. Finally, the player can restart or finalize the game.

Scatter plots were generated for each attribute in the original dataset (Figure 6). Notably, patterns emerged concerning the precision of players' ship movement events. The relationship between the main attributes of the dataset can be observed in the case of collisions (*hit*), where a linear relationship with the rest of the attributes can be identified in the central column. However, for the level (*lvl*) in the right column of the same Figure, a discernible pattern that would allow us to prioritize the attribute cannot be identified.

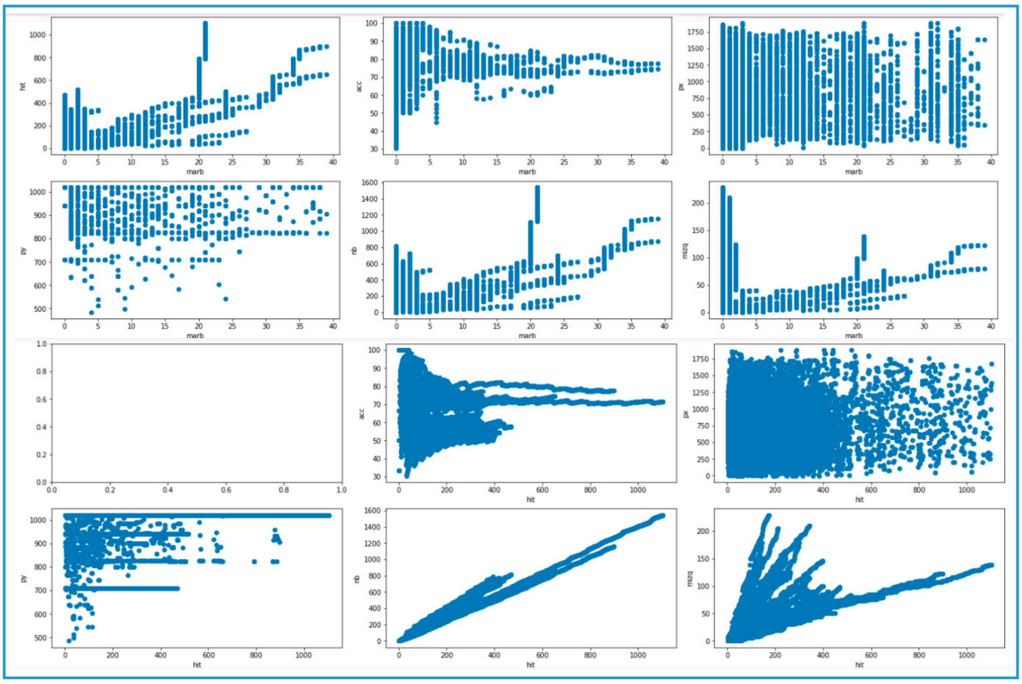


Figure 6. Scatter plots for the dataset variables.

3.4. Model Validation

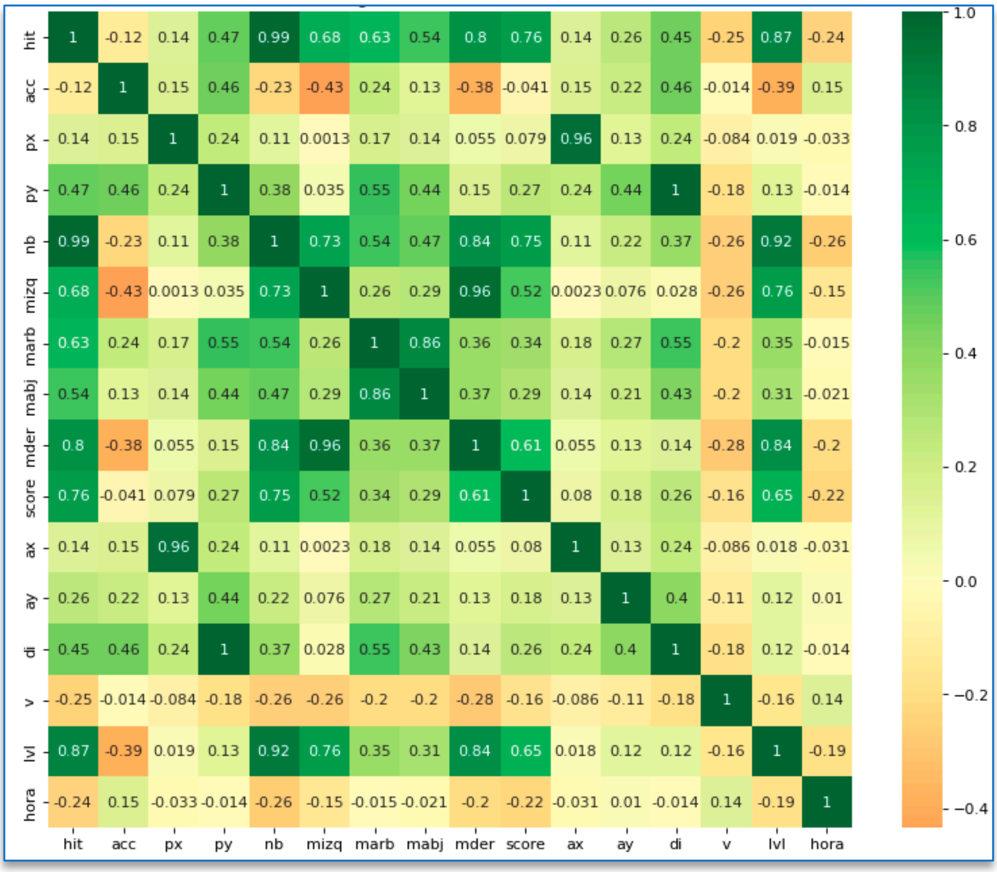
The model was developed in Python running TensorFlow, Scikit-learn, and other libraries, such as NumPy, Matplotlib, and Pandas.

A confusion matrix [25] was generated to analyze how our FFN algorithm works. Figure 7 shows the confusion matrix.

A confusion matrix is a performance measurement tool used in machine learning to evaluate the accuracy of a classification model. This helps us visualize the true positive, true negative, false positive, and false negative predictions of the model [26]. In this case, the confusion matrix helped to identify the input and output attributes for generating the deep learning model. For example, it revealed a strong correlation between the number of shots and the level of execution, as well as the movements of the player's ship with the collision attribute. Furthermore, by analyzing the confusion matrix, we identified the attributes that were most influential in predicting the outcome, allowing us to choose the most relevant attributes to be used in developing the deep learning model.

We used a Feedforward Neural Network (FNN) architecture [27] because the connections between the nodes do not create a cycle, allowing the data flow from their input to move through the hidden layers until they reach the output node. This prevents a cycle from occurring between nodes; that is, information flows in one direction from the input layer through several hidden layers to the output layer.

An FNN is useful in supervised learning [28], where there is a set of input data and its corresponding expected output. The network is trained to learn the relationship between the inputs and outputs by adjusting the weights and biases of the neurons in the hidden layers of the network. Once the network has been trained, it can predict the output of the new inputs.



	hit	acc	px	py	nb	mizq	marb	mabj	mder	score	ax	ay	di	v	lvl	hora
hit	1	-0.12	0.14	0.47	0.99	0.68	0.63	0.54	0.8	0.76	0.14	0.26	0.45	-0.25	0.87	-0.24
acc	-0.12	1	0.15	0.46	-0.23	-0.43	0.24	0.13	-0.38	-0.041	0.15	0.22	0.46	-0.014	-0.39	0.15
px	0.14	0.15	1	0.24	0.11	0.0013	0.17	0.14	0.055	0.079	0.96	0.13	0.24	-0.084	0.019	-0.033
py	0.47	0.46	0.24	1	0.38	0.035	0.55	0.44	0.15	0.27	0.24	0.44	1	-0.18	0.13	-0.014
nb	0.99	-0.23	0.11	0.38	1	0.73	0.54	0.47	0.84	0.75	0.11	0.22	0.37	-0.26	0.92	-0.26
mizq	0.68	-0.43	0.0013	0.035	0.73	1	0.26	0.29	0.96	0.52	0.0023	0.076	0.028	-0.26	0.76	-0.15
marb	0.63	0.24	0.17	0.55	0.54	0.26	1	0.86	0.36	0.34	0.18	0.27	0.55	-0.2	0.35	-0.015
mabj	0.54	0.13	0.14	0.44	0.47	0.29	0.86	1	0.37	0.29	0.14	0.21	0.43	-0.2	0.31	-0.021
mder	0.8	-0.38	0.055	0.15	0.84	0.96	0.36	0.37	1	0.61	0.055	0.13	0.14	-0.28	0.84	-0.2
score	0.76	-0.041	0.079	0.27	0.75	0.52	0.34	0.29	0.61	1	0.08	0.18	0.26	-0.16	0.65	-0.22
ax	0.14	0.15	0.96	0.24	0.11	0.0023	0.18	0.14	0.055	0.08	1	0.13	0.24	-0.086	0.018	-0.031
ay	0.26	0.22	0.13	0.44	0.22	0.076	0.27	0.21	0.13	0.18	0.13	1	0.4	-0.11	0.12	0.01
di	0.45	0.46	0.24	1	0.37	0.028	0.55	0.43	0.14	0.26	0.24	0.4	1	-0.18	0.12	-0.014
v	-0.25	-0.014	-0.084	-0.18	-0.26	-0.26	-0.2	-0.2	-0.28	-0.16	-0.086	-0.11	-0.18	1	-0.16	0.14
lvl	0.87	-0.39	0.019	0.13	0.92	0.76	0.35	0.31	0.84	0.65	0.018	0.12	0.12	-0.16	1	-0.19
hora	-0.24	0.15	-0.033	-0.014	-0.26	-0.15	-0.015	-0.021	-0.2	-0.22	-0.031	0.01	-0.014	0.14	-0.19	1

Figure 7. Confusion matrix of the proposed model.

One of the advantages of the FNN architecture is that it can be trained to solve a wide variety of problems, from classification to regression. It can also handle data with multiple features, meaning that it can work with complex and high-dimensional datasets [29].

The complexity of the Feedforward Neural Network (FNN) used in our study contributes to the robustness and adaptability of the system, enabling it to handle a variety of players and skill levels. The multilayered structure of the FNN, combined with its ability to learn from and adapt to new data, allows the system to analyze a wide range of player skills in-depth [30]. By discerning complex patterns within the gameplay data, our system can accurately adjust game difficulty to enhance player engagement and satisfaction. Our analysis demonstrates the equilibrium we achieved between system complexity and its ability to accommodate various player skill levels.

However, the FNN architecture has certain disadvantages. One of these is that it can be prone to overfitting, which means that the network can learn to fit too closely to the training data and cannot generalize well to new data. Additionally, training the network requires intensive data and computation [31].

In summary, the FNN architecture is a powerful machine learning tool that is used in a wide variety of applications. Although it has some disadvantages, its ability to handle complex data and solve classification and regression problems makes it a valuable tool for data scientists and machine learning engineers.

These advantages fit our sequential model process since the recording of the game sessions is performed every time the player makes a shot; therefore, the obtained metrics depend on the previous metrics to perform their classification. It has 13 main input attributes and 64 unique enemy ship identifiers. The main attributes are mentioned in the previous metric collection section of this document with a ReLU activation function. This layer is considered the input layer in a neural network [32].

The next hidden layer of the network consisted of eight states with softmax activation. Finally, three result attributes were identified in the last layer using a sigmoid activation function. Finally, the model was compiled with the instruction to perform a binary cross-entropy algorithm with the Adam optimizer.

Model validation, which enables the classification of the video gamer level, reveals a high degree of accuracy in both the training and validation phases. The results demonstrated a precision of 99.6% in the training phase and 99.83% in the validation phase. These findings suggest that the Feedforward Neural Network architecture used in this study is highly effective in accurately classifying the level of a video gamer.

The high accuracy achieved by the model in this study can be attributed to the ability of Feedforward Neural Networks to learn complex patterns and relationships within the data. During the training phase, the network adjusts the weights of its connections to minimize the difference between the predicted and actual outputs. This learning process continues until the network can accurately classify new data.

The deep learning model's predictions are represented by ranges from 0 to 2 and divided into three distinct categories based on the player's level of expertise. A prediction of 0 indicates that the player has a high level of experience in the game, whereas a prediction of 1 suggests an intermediate level of experience. Finally, a prediction of 2 indicated that the player had a low level of experience in the game.

Dynamic difficulty adjustment (DDA) was developed with this deep learning model to adjust game difficulty based on player performance. When the model classifies the player as having a high level of experience (0), the game difficulty increases by 3. When a player is classified as having an intermediate experience level (1), the game difficulty increases by 1. Conversely, when the player is classified as having a low level of experience (2), the game difficulty is decreased by one, allowing for the dynamic adjustment of game complexity based on the player's detected level. This approach to DDA has significant implications for game design as it creates personalized and challenging gaming experiences that can adapt to the player's skill level in real-time. Furthermore, this approach can enhance player engagement and satisfaction by providing a more immersive and enjoyable gaming experience.

To validate the player engagement through DDA and adjust the game difficulty in real-time based on the player's performance, we conducted a user experience evaluation. The following section presents a study with users to evaluate the effectiveness of DDA in enhancing player engagement in the video game, Astro. The evaluation aimed to investigate how the DDA algorithm adapts to a player's skill level and affects the game experience. The findings of this study provide insights into the potential of DDA to improve player engagement and inform the development of future games.

Finally, Figure 8 presents a flowchart to illustrate our research framework visually. This diagram offers readers a quick overview of the research process.

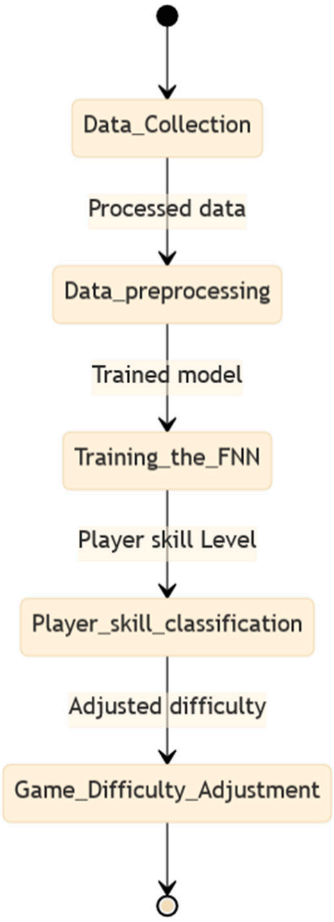


Figure 8. Research framework diagram.

4. Evaluation

To validate whether the dynamic adjustment of game complexity improves engagement in video game players, a gameplay experience evaluation was conducted. Ten undergraduate students, who were convenience sampled based on availability, participated in the evaluation (see Figure 9).

Figure 9. Evaluation session.

The average age of the participants was 20 years (min. 20, max. 22). All participants mentioned that they were video game players, with 60% identifying themselves as casual players and 40% as experienced players. The most commonly used platforms were PC (90%), laptops (60%), and consoles (30%).

4.1. Procedure

A two-hour evaluation was conducted, consisting of four stages [33,34]. During Stage 1 of the evaluation, the participants were given a brief explanation of the objectives and goals of the evaluation as well as an overview of the features and mechanics of Astro. This was done to provide context and establish expectations for the rest of the evaluation.

In Stage 2, the participants were given a live demonstration of Astro, where they were shown how to navigate the game's menus, use the controls, and interact with various game elements. Again, this ensured that the participants understood how to play the game.

In Stage 3, the participants played through the entire Astro scenario, which involved completing various tasks and challenges while navigating at different levels. This stage aimed to assess how well the participants could engage with the game and how well they could perform under the different levels of complexity assigned by the deep learning model.

Finally, in Stage 4, participants were asked to complete a survey regarding their experience with the video game. The survey included questions about their level of engagement, enjoyment, and satisfaction with the game, and their opinions on the level of challenge and complexity presented by the game. The instrument was based on two validated instruments [35,36] and can be found in Appendix A. These data were collected and analyzed to determine the overall effectiveness of the dynamic complexity adjustment feature in improving player engagement and satisfaction.

4.2. Results

Most participants enjoyed playing the game based on the survey responses. Most participants (8 out of 10) rated their enjoyment of the game as 4 out of 5, -1 (less enjoyment) and 5 (most enjoyment); however, the remaining two participants rated their enjoyment as 3 out of 5. This result suggests that most players found the game enjoyable, but not everyone found it to be equally engaging.

The participants' responses varied more regarding the difficulty of the game, with scores ranging from 4 to 9 on a 10-point scale, 1 (too easy) and 10 (too hard). Most participants rated the game as moderately challenging, with 50% of the players giving scores ranging from four to six. However, one participant scored nine, suggesting that they found the game challenging. This suggests that the game is relatively easy for most players.

Regarding the self-perception of performance in the game, the results of the question were primarily negative, with 60% of players rating their performance below the average as < 5 out of 10 on the scale where one is very bad and ten is awesome. This result could indicate that the players found the game challenging and struggled to perform well. On the other hand, the remaining 40% rated their performance in a range from 6 to 10, indicating that they found the game more manageable than other players.

The survey also validates the player's flow experience (see Figure 10).

The results showed that playing Astro provided a flow experience to most participants, with 90% of the players reporting high levels of enjoyment, interest, and immersion in the game. Furthermore, all the participants felt in control of their actions and knew what to do during the gaming activity, indicating a high level of competence.

Interestingly, the participants' responses were divided according to their feeling of time passing fast during the activity, with 70% agreeing and 30% disagreeing.

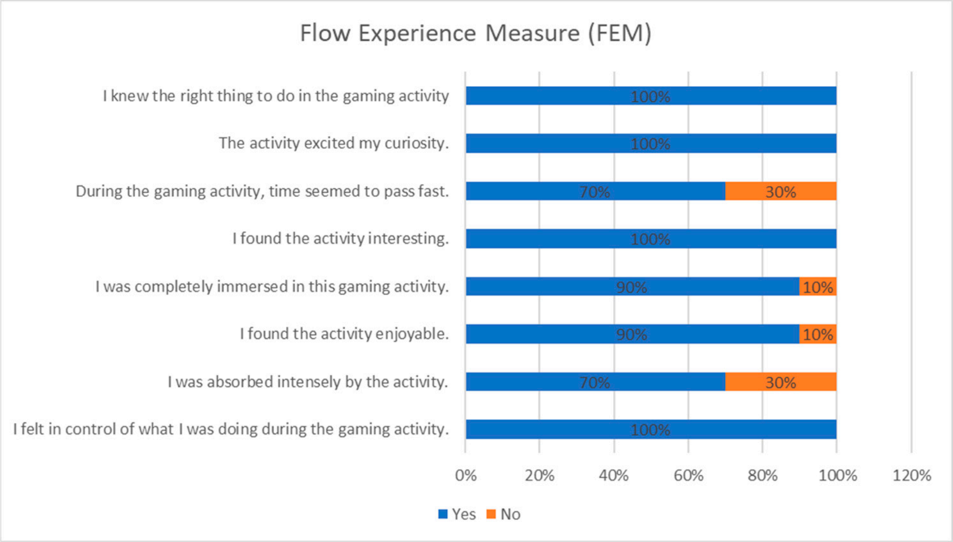


Figure 10. Results of the flow experience measure.

Overall, this result suggests that the dynamic adjustment of game complexity in Astro provided a flow experience to most participants, which is a positive outcome in this research project because we validated the flow experience of the player.

Finally, participants were asked about their self-perception of the dynamic adjustment of complexity. Seventy percent of the participants mentioned that they noticed some sort of complexity adjustment, while the remaining 30% did not. This result suggests that a significant majority of the participants noticed the dynamic adjustment of complexity in Astro, implying that the game provided a challenging and stimulating experience. Those who noticed the complexity adjustment were asked if they believed the adjustment affected their performance. Responses were on a scale of 1 to 10, where 1 was "Very bad" and 10 was "Wonderful." A total of 29% believed that their performance was moderately affected, while 71% responded in the range of six to nine. This indicates that they perceived the dynamic adjustment of complexity in Astro to be favorable for their game experience, indicating that the game's adaptive difficulty was effective in enhancing their experience. This confirms that incorporating dynamic complexity adjustment in game design can improve player satisfaction and enjoyment, showing that the intelligent model successfully classifies players into skill levels and leads to improved player engagement and satisfaction.

Four game sessions were randomly selected for analysis and discussion to demonstrate how the model adjusted difficulty and levels. Table 3 shows that the FNN model can effectively adjust the difficulty of a game session. In most cases, players advanced to higher levels, demonstrating that FNN achieved a balance between challenge and accessibility.

Table 3. Summary of game sessions and difficulty settings.

Session	Start Level	End Level	Skill Level	Accuracy (%)
1	1	12	Beginner	62.39
2	1	16	Intermediate	65.60
3	1	10	Intermediate	92.17
4	1	11	Expert	69.70

In session 1, for example, the player started at Level 1 and advanced to Level 12. The model adjusted the difficulty so that the player experienced a progression of skill, ending up in the beginner category with an accuracy of 62.39%. In Session 2, the player also started at Level 1 but progressed to Level 16, finishing in the intermediate category with an accuracy of 65.60%.

This result suggests that the FNN model effectively matches the players' abilities, allowing them to experience progression and maintain a high level of engagement. However, it is important to note that the model not only adjusts the difficulty upwards but can also decrease the difficulty if the player's performance drops, ensuring that the game remains accessible and challenging.

Finally, the possibility of a placebo effect in the players' knowledge that the system can adjust difficulty is evident, but further analysis is required to confirm this hypothesis.

5. Discussion

While the Feedforward Neural Network (FNN) used in this study demonstrated high precision in the training and validation phases, the inherent randomness in neural network algorithms may still introduce some uncertainty. Cross-validation can help manage this uncertainty and provide a more robust estimation of model performance. Cross-validation involves:

- Partitioning the dataset into several subsets or folds;
- Training the model on most of these folds;
- Testing the model's performance on the remaining fold.

This process is repeated with different folds serving as the test set each time, ensuring that the performance of the model is assessed on various subsets of data.

For instance, one cross-validation method involves splitting the dataset into ten parts. The model is trained on nine parts; the remaining part is used for validation. This process is repeated ten times, each part being used for training and validation. The results are then averaged to provide a more accurate estimate of the model's performance. Cross-validation can also help determine if the model is overfitting the training data, which is important when using complex models like FNNs. This method provides a better understanding of the model's ability to classify player skill levels and reliability.

In terms of development challenges, we can highlight the following:

- **Data quality and availability**: In the context of our project, the performance of the FNN relies heavily on the amount and diversity of the gameplay data it is trained on. Accumulating relevant, high-quality data representing diverse gameplay styles, skill levels, and game situations is a substantial challenge.
- **Hyperparameter tuning**: The performance of our model depends significantly on the precise tuning of various hyperparameters, including the number of hidden layers, neurons in these layers, and the learning rate. Determining the optimal values of these parameters requires multiple trial-and-error runs.

6. Conclusions

The results of this study suggest that the use of the DL approach can be an effective strategy for generating a skill-based classification of players. This classification can personalize the gaming experience and improve player engagement. Furthermore, game developers can provide different challenges to players by understanding their expertise based on their skills, leading to a more tailored experience.

The results of this study indicate that incorporating dynamic complexity adjustment into game design can enhance player satisfaction and enjoyment. The intelligent model trained with DL effectively adapted the game's difficulty level to the player's skill level, leading to a more challenging and engaging game experience. In addition, players who found the game challenging and stimulating were likelier to rate their performance better, leading to a more satisfying experience. Moreover, incorporating dynamic complexity adjustment could lead to better retention rates, as players are more likely to continue playing a game that adapts to their skill level rather than one that is too easy or difficult.

However, this study also revealed that not all players might find the game equally engaging, and some may struggle with the difficulty level. Therefore, further research could investigate ways to adjust the game difficulty for individual players based on their performance to optimize their gaming experience.

Overall, this study's results suggest that DL's use of classifying players based on their skills and incorporating dynamic complexity adjustment into game design could lead to a more engaging and satisfying game experience for players. These findings have significant implications for game developers and could lead to more personalized and challenging games that appeal to a broader range of players.

7. Limitations and Future Works

Although our research findings are promising, we acknowledge certain limitations in our study that require further exploration in future research. The data to train and validate the deep learning model was limited to one game. Given the diversity of gameplay mechanics, styles, and player interactions across various games, it is not possible to acknowledge the model's generalizability.

Furthermore, a player's skill level may not match their preferred game difficulty. For instance, highly skilled players might opt for an easier gaming experience, whereas less-skilled players may seek challenges beyond their current abilities. Future research could explore mechanisms that account for player preferences when adjusting game difficulty to achieve a more personalized gaming experience.

In addition, our current system adjusts the difficulty level based on skill classification only once a game session is concluded. This method must be revised to adjust for real-time player skills and engagement fluctuations during a gaming session.

We must consider the computational complexity of our deep learning model, which may impose constraints on its practical implementation, particularly in environments with limited resources. Future research could leverage more computationally efficient models or optimization techniques to address complexity. The goal is to reduce the computational load while ensuring that the model's accuracy in classifying player skill and adjusting game difficulty remains uncompromised.

The study did not present an evaluation with a control group in which the DL model is not utilized during gameplay. By comparing the standard gameplay of Astro with the gameplay featuring complexity adjustment, it would be possible to determine whether there were statistically significant differences in the gaming experience between the two conditions.

Moreover, there is a need to delve deeper into players' perceptions of the complexity adjustment mechanism. Understanding how players perceive and react to dynamic difficulty adjustments can provide valuable insight into their subjective experiences and emotional responses. This exploration can shed light on the factors contributing to player motivation, self-evaluation, and overall enjoyment of the gaming experience.

Additionally, an analysis to explore potential gender differences in reactions and responses would allow for developing more inclusive and tailored approaches to game design and difficulty adjustment.

Our goal is to enhance the effectiveness and practicality of dynamic game difficulty adjustment systems by recognizing our limitations and developing strategies to overcome them. By doing so, we aim to improve the overall gaming experience and increase player engagement and satisfaction in the industry.

Author Contributions: Conceptualization, P.C.S.-M. and O.A.M.-L.; methodology, P.C.S.-M.; software, E.A.R.-M.; validation, E.A.R.-M. and P.C.S.-M.; formal analysis: M.G.-R. and L.E.A.-R.; data curation, E.A.R.-M.; writing—original draft preparation, P.C.S.-M. and E.A.R.-M.; writing—review and editing: M.G.-R., O.A.M.-L., and L.E.A.-R. All authors have read and agreed to the published version of the manuscript.

Funding: This study received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all the subjects involved in the study.

Data Availability Statement: The data presented in this study are available upon reasonable request from the corresponding author.

Acknowledgments: The authors extend their gratitude to the volunteers who contributed their time and effort to participate in the gaming experience assessment and dataset creation. Their participation significantly affected the quality and depth of the research.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

The player experience assessment instrument is presented next.

General information

- Name
- Age
- Where do you regularly play Video Games? (You can mark more than one option)
  - ○ Home
  - ○ School Cybercafe
  - ○ Other
- What platform do you use to play Video Games on? (You can mark more than one option)
  - ○ Console
  - ○ PC
  - ○ Portable console
  - ○ Other

Enjoyment

- Are you a casual or experienced gamer?
  - ○ Casual
  - ○ Experienced
- How much did you enjoy the game?
  - ○ 1 to 5, where 1 is Less and 5 is Most
- How difficult did you find the game?
  - ○ 1 to 10, where 1 is Too easy and 10 is Too hard
- How well do you think you did?
  - ○ 1 to 10, where 1 is Very bad and 10 is Awesome

Flow Experience Measure (FEM)

- I felt in control of what I was doing during my gaming activity. (Yes/No)
- I was absorbed intensely by the activity (Yes/No)
- I found the activity enjoyable (Yes/No)
- I was completely immersed in this gaming activity (Yes/No)
- I found the activity interesting (Yes/No)
- During the gaming activity, time seemed to pass fast (Yes/No)
- The activity excited my curiosity (Yes/No)
- I knew the right thing to do in the gaming activity. (yes/no)

Closing questions

- Do you notice some difficult adjustments? (Yes/No)
- If yes, Do you believe difficult adjustment affected your performance?
  - ○ 1 to 10, where 1 is Very bad and 10 is Awesome

References

1. Turing, A.M. Computing Machinery and Intelligence. In *Parsing the Turing Test*; Epstein, R., Roberts, G., Beber, G., Eds.; Springer: Dordrecht, The Netherlands, 2009; pp. 23–65, ISBN 978-1-4020-9624-2.
2. El Naqa, I.; Murphy, M.J. What Is Machine Learning? In *Machine Learning in Radiation Oncology*; El Naqa, I., Li, R., Murphy, M.J., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 3–11, ISBN 978-3-319-18304-6.
3. Ishii, R.; Ito, S.; Thawonmas, R.; Harada, T. A Fighting Game AI Using Highlight Cues for Generation of Entertaining Gameplay. In Proceedings of the 2019 IEEE Conference on Games (CoG), London, UK, 20–23 August 2019; IEEE: London, UK, 2019; pp. 1–6.
4. Zhang, M.; Zhang, J.; Zhang, D. ATVR: An Attention Training System Using Multitasking and Neurofeedback on Virtual Reality Platform. In Proceedings of the 2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR), San Diego, CA, USA, 9–11 December 2019; IEEE: San Diego, CA, USA, 2019; pp. 159–1593.
5. Buenrostro-Mariscal, R.; Santana-Mancilla, P.C.; Montesinos-López, O.A.; Nieto Hipólito, J.I.; Anido-Rifón, L.E. A Review of Deep Learning Applications for the Next Generation of Cognitive Networks. *Appl. Sci.* **2022**, *12*, 6262. [CrossRef]
6. Montesinos-López, O.A.; Montesinos-López, A.; Pérez-Rodríguez, P.; Barrón-López, J.A.; Martini, J.W.R.; Fajardo-Flores, S.B.; Gaytan-Lugo, L.S.; Santana-Mancilla, P.C.; Crossa, J. A Review of Deep Learning Applications for Genomic Selection. *BMC Genom.* **2021**, *22*, 19. [CrossRef] [PubMed]
7. Zohaib, M. Dynamic Difficulty Adjustment (DDA) in Computer Games: A Review. *Adv. Hum. -Comput. Interact.* **2018**, *2018*, 5681652. [CrossRef]
8. Sánchez, J.L.G.; Vela, F.L.G.; Simarro, F.M.; Padilla-Zea, N. Playability: Analysing User Experience in Video Games. *Behav. Inf. Technol.* **2012**, *31*, 1033–1054. [CrossRef]
9. Soderman, B. *Against Flow: Video Games and the Flowing Subject*; The MIT Press: Cambridge, MA, USA, 2021; ISBN 978-0-262-04550-6.
10. Csikszentmihalyi, M.; Abuhamdeh, S.; Nakamura, J. Flow. In *Flow and the Foundations of Positive Psychology*; Springer: Dordrecht, The Netherlands, 2014; pp. 227–238, ISBN 978-94-017-9087-1.
11. Chen, J. Flow in Games (and Everything Else). *Commun. ACM* **2007**, *50*, 31–34. [CrossRef]
12. Cowley, B.; Charles, D.; Black, M.; Hickey, R. Toward an Understanding of Flow in Video Games. *Comput. Entertain.* **2008**, *6*, 1–27. [CrossRef]
13. Alvarado Villa, D.A.; Montesinos López, O.A.; Santana-Mancilla, P.C. Training of an Intelligent Agent to Improve the Gaming Experience for Video Gamers. *Av. IHC* **2021**, *123*, 123–125. [CrossRef]
14. Pedersen, C.; Togelius, J.; Yannakakis, G.N. Modeling Player Experience for Content Creation. *IEEE Trans. Comput. Intell. AI Games* **2010**, *2*, 54–67. [CrossRef]
15. Porssut, T.; Hou, Y.; Blanke, O.; Herbelin, B.; Boulic, R. Adapting Virtual Embodiment Through Reinforcement Learning. *IEEE Trans. Visual. Comput. Graphics* **2022**, *28*, 3193–3205. [CrossRef]
16. Comi, M. How to Teach AI to Play Games: Deep Reinforcement Learning. *Towards Data Sci.* **2018**. Available online: https://towardsdatascience.com/how-to-teach-an-ai-to-play-games-deep-reinforcement-learning-28f9b920440a (accessed on 14 March 2023).
17. Epic Games. Unreal Tournament. 1999. Available online: https://www.epicgames.com/unrealtournament (accessed on 14 March 2023).
18. Berg, A. If It's Fun, It's Fun: Deep Reinforcement Learning In Unreal Tournament 2004. Bachelor's Thesis, University of Borås, Borås, Sweden, 2018.
19. Wang, D.; Tan, A.-H. Creating Autonomous Adaptive Agents in a Real-Time First-Person Shooter Computer Game. *IEEE Trans. Comput. Intell. AI Games* **2015**, *7*, 123–138. [CrossRef]
20. Ismail, H.; Merrick, K.; Barlow, M. Self-Motivated Learning of Achievement and Maintenance Tasks for Non-Player Characters in Computer Games. In Proceedings of the 2014 IEEE Symposium on Computational Intelligence for Human-like Intelligence (CIHLI), Orlando, FL, USA, 9–12 December 2014; IEEE: Orlando, FL, USA, 2014; pp. 1–8.
21. Justesen, N.; Bontrager, P.; Togelius, J.; Risi, S. Deep Learning for Video Game Playing. *IEEE Transact. Games* **2017**, *12*, 1–20. [CrossRef]
22. Aleem, S.; Capretz, L.F.; Ahmed, F. Game Development Software Engineering Process Life Cycle: A Systematic Review. *J. Softw. Eng. Res. Dev.* **2016**, *4*, 6. [CrossRef]
23. Wikipedia. Space Invaders. Wikipedia, the Free Encyclopedia. 2023. Available online: https://en.wikipedia.org/wiki/Space_Invaders (accessed on 14 March 2023).
24. McGugan, W. *Beginning Game Development with Python and Pygame: From Novice to Professional*; Expert's Voice in Open Source; Apress; Distributed to the Book Trade Worldwide by Springer-Verlag New York; Springer: Berkeley, CA, USA; New York, NY, USA, 2007; ISBN 978-1-59059-872-6.
25. Visa, S.; Ramsay, B.; Ralescu, A.L.; Van Der Knaap, E. Confusion matrix-based feature selection. *Maics* **2011**, *710*, 120–127.
26. Susmaga, R. Confusion Matrix Visualization. In *Intelligent Information Processing and Web Mining*; Kłopotek, M.A., Wierzchoń, S.T., Trojanowski, K., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; pp. 107–116, ISBN 978-3-540-21331-4.
27. Ojha, V.K.; Abraham, A.; Snášel, V. Metaheuristic Design of Feedforward Neural Networks: A Review of Two Decades of Research. *Eng. Appl. Artif. Intell.* **2017**, *60*, 97–116. [CrossRef]
28. Murat, H.S. A brief review of feed-forward neural networks. *Commun. Fac. Sci. Univ. Ank.* **2006**, *50*, 11–17. [CrossRef]

29. Ozanich, E.; Gerstoft, P.; Niu, H. A Feedforward Neural Network for Direction-of-Arrival Estimation. *J. Acoust. Soc. Am.* **2020**, *147*, 2035–2048. [CrossRef]
30. Zhang, Z.; Qiao, J. A Node Pruning Algorithm for Feedforward Neural Network Based on Neural Complexity. In Proceedings of the 2010 International Conference on Intelligent Control and Information Processing, Dalian, China, 13–15 August 2010; IEEE: Dalian, China, 2010; pp. 406–410.
31. Chaudhuri, T.; Soh, Y.C.; Li, H.; Xie, L. A Feedforward Neural Network Based Indoor-Climate Control Framework for Thermal Comfort and Energy Saving in Buildings. *Appl. Energy* **2019**, *248*, 44–53. [CrossRef]
32. Eckle, K.; Schmidt-Hieber, J. A Comparison of Deep Networks with ReLU Activation Function and Linear Spline-Type Methods. *Neural Netw.* **2019**, *110*, 232–242. [CrossRef]
33. Santana-Mancilla, P.C.; Rodríguez-Ortiz, M.A.; Gaytan-Lugo, L.; Garcia-Ruiz, M. User Experience Evaluation to El Encanto: A Serious Game for Learning Community Intervention. *Res. Comput. Sci.* **2015**, *108*, 81–88. [CrossRef]
34. Santana-Mancilla, P.C.; Gaytán-Lugo, L.S.; Rodríguez-Ortiz, M.A. Usability Testing of Serious Games: The Experience of the IHCLab. In *Games User Research: A Case Study Approach*; CRC Press: Boca Raton, FL, USA, 2016; pp. 271–283.
35. Alexander, J.T.; Sear, J.; Oikonomou, A. An Investigation of the Effects of Game Difficulty on Player Enjoyment. *Entertain. Comput.* **2013**, *4*, 53–62. [CrossRef]
36. Sung, H.-Y.; Hwang, G.-J.; Yen, Y.-F. Development of a Contextual Decision-Making Game for Improving Students' Learning Performance in a Health Education Course. *Comput. Educ.* **2015**, *82*, 179–190. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.