Identifying Play Styles in Shoot 'Em Up Games

Madhav Girish Nair
Utrecht University
Utrecht, The Netherlands
m.girishnair@students.uu.nl

Bart van Herk
Utrecht University
Utrecht, The Netherlands
b.r.vanherk@students.uu.nl

Jin Kai Huang
Utrecht University
Utrecht, The Netherlands
j.k.huang@students.uu.nl

Jesse Kettmann
Utrecht University
Utrecht, The Netherlands
j.kettmann@uu.nl

Charl-Pierre Marais
Utrecht University
Utrecht, The Netherlands
c.marais@students.uu.nl

Wouter Ubbink
Utrecht University
Utrecht, The Netherlands
w.a.ubbink@students.uu.nl

Abstract

Player modelling is an approach often used in game development to improve the players' game experience. With the new developments within AI, the scope of modelling a player using AI expands to new avenues. In this paper, we developed a shoot 'em up game, called `Hell-bent`. Data was then collected from various participants playing the game, and the features extracted from the data were used to perform K-means clustering to isolate different strategic approaches different players took. Statistical analysis conducted on the clustered data indicated the presence of passive and aggressive play styles.

CCS Concepts

• **Computing methodologies → Cluster analysis**.

Keywords

Arificial intelligence, Machine learning, AI in games, Play styles, Player types, Player modelling, Unsupervised learning, Clustering

ACM Reference Format:
Madhav Girish Nair, Bart van Herk, Jin Kai Huang, Jesse Kettmann, Charl-Pierre Marais, and Wouter Ubbink. 2024. Identifying Play Styles in Shoot 'Em Up Games. In *Proceedings of AI for game technology (INFOMAIGT '24)*. ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Artificial intelligence (AI) has seen rapid growth in the past few years, with increasingly more interested parties and AI-integrated products and services becoming available to consumers worldwide [22]. One sub-topic of AI is particularly interesting, namely, Game AI. Games have been used by multiple research groups and companies as a benchmark for their new AI technologies, while the game industry has also been implementing various new AI models in their games over the past few decades [12, 22]. Game AI encompasses the latter and plays an important role in game development. One such example of game AI is player modelling, which is an area of research dedicated to using AI to model player behaviour [16, 21]. Player modelling is interesting because it allows us to better understand the players' experience and behaviour while playing a game [7, 8, 22]. This allows modifications to be made to the game itself or features within the game in order to tune it to the likings of the player. This not only makes the game more entertaining for the player, but can also be used to dynamically change features, such as the difficulty, for player retention purposes.

In this paper, we will focus on behavioural patterns as we will try to identify play styles in shoot 'em up games. To achieve this, we implemented a shoot 'em up game, called `Hell-bent`, where the goal is to earn as many points as possible. Various means of obtaining points were implemented in the game and the game logs game events which allows the analysis of behavioural data. Afterwards, participants were recruited, and their game data were collected. After exploring and pre-processing the data, the features were defined. Then, we used these features in K-means clustering to find play style clusters.

The results showed that two play-styles, namely ***passive*** and ***aggressive***, emerged from the collected data with statistically significant differences. The main contributions of this study are: (1) showing that even with a small data set, selecting the right features can lead to segregation of play styles, and (2) showing that there is potential in identifying play styles in shoot 'em up games using only in-game data. However, for comprehensive evaluation of strategies beyond the surface level, careful game and feature engineering is required.

2 Related Work

In order to properly understand the motivation of this research project, it is important to grasp the main concepts in player modelling related research areas and identify where the gaps lie. First, we will briefly explain the shoot 'em up game genre and the game `LUFTRAUSERS`™, which is the main inspiration of `Hell-bent`. Then, we will give a description of player modelling and discuss various different approaches taken in previous literature. Lastly, we will define our research goal.

2.1 Background

In a shoot 'em up game, the player is tasked with defending themselves from a continuous barrage of enemy fire. At its core, the gameplay is quite simple, and the first game of the sort (`Spacewar!`)

INFOMAIGT '24, Sep. –Nov. 2024, Utrecht, The Netherlands

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/XXXXXXX.XXXXXXX

dates back to 1962. However, throughout the decades, the genre has grown and a wide variety of shoot 'em up games with different mechanics have been brought onto the market [19]. One such example is the game LUFTRAUSERS™.

In LUFTRAUSERS™, the player controls a plane flying above the sea. Enemies consist of other planes, but also ships floating on the water, and underwater submarines. The game contains 125 combinations of weapons, bodies, and propulsion systems, allowing the player to customise their plane to their liking. Two unique aspects about the game are its movement and shooting mechanics and how these are intertwined. The plane has buttery movement, and can only move forwards, so turning is required to navigate. This means that mastering the movement mechanics of the game takes time and skill. While the planes move in a traditionally top-down fashion, the game graphically depicts the side view of the game world. Consequently, gravity pulls the plane down whenever the player is not accelerating, and a large body of water prevents the player from moving too far down with its damage over time effect. Furthermore, the plane can only shoot in the direction it is facing, meaning that the shooting mechanics are closely tied to the movement mechanics. This combination makes the core gameplay mechanics of the game hard to master.

Another unique aspect of the game is that the player can also deal damage by ramming into enemies. This combined with its health regeneration system, where a player only regenerates health when it is not shooting bullets or taking damage, allows for interesting play styles such as melee-focussed play styles. Shooting also negatively affects the turning speed of the plane, discouraging players from constantly holding down the fire button [18]. The fact that the game encourages the player to try different play styles is what makes us think the game is a good fit for this research project, as we are trying to identify different player types.

2.2 Player Modelling

As defined by Yannakakis and Togelius, player modelling is *"the detection, prediction, and expression of human player characteristics that are manifested through cognitive, affective and behavioural patterns while playing games [22]."* At a high level, player modelling approaches are typically classified as either model-based (top-down) or model-free (bottom-up) [20, 21].

In a model-based approach, a player model is built on a theoretical framework. These approaches are often inspired by theory from psychology and other related fields, e.g., the cognitive appraisal theory [10, 15]. Although there exists an abundance of literature in these fields, it is important to take into account that most of it has not been tested on interactive media such as games [2, 22]. It is easy to create a model, but it is difficult to create a model that yields valuable results.

The second option is a model-free approach. This approach consists of a data-driven construction of an unknown mapping between a player input and a player state [22]. Recently, this approach has become a popular choice for behavioural data mining in games, which aligns with our goal of identifying behaviours, i.e., player types [7]. Various other works have tried to identify different behavioural patterns in games before [8].

Drachen and Schubert provided a brief review of the state of the art of spatial game analytics in their work. They mention that spatial game analytics, and analytics in general, have the potential to transform game development in the same way it has done in other sectors such as marketing. Furthermore, they presented four key areas of spatial and spatio-temporal analytics: spatial outlier detection, spatial clustering, spatial predictive models, and spatial pattern and rule mining [5]. Drachen et al. applied spatial clustering in their work to find player types in `Tomb Raider: Underworld`. In their work, they used six features and were able to identify four player type clusters using unsupervised learning. They mention that their approach should work for any game genre, especially those featuring a central player character, as long as reliable playing behaviour data is available [4]. Therefore, an unsupervised learning approach might be a good fit for our research project as well. Drachen et al. further enforce this belief. In their work, they presented two case studies focussing on clustering analysis applied to high-dimensionality player behaviour telemetry from two major game titles, namely `Tera` and `Battlefield 2: Bad Company 2`. They applied K-means and Simplex Volume Maximisation clustering to the two data sets and were able to identify actionable behavioural profiles [6].

Various researchers have applied player modelling to shoot 'em up games in the past. de Araujo and Feijó evaluated dynamic difficulty adaptivity in their work. They implemented an adaptive shoot 'em up game where the difficulty adapts between every enemy wave. Their adaptive algorithm took the player's performance as input and controlled the values of enemy variables, such as their speed and firing rate. They found that the adaptiveness suited hardcore players more, while casual players got more easily frustrated [3]. Halbhuber et al. took a different approach. In their study, they used the player's affective state in a shoot 'em up game to avoid frustration and boredom. Instead of dynamically adjusting the game's difficulty solely based on the player's performance, their algorithm also took into account their emotions, which were tracked through facial expressions, self-evaluation, and keystrokes. They note that facial expressions and keystrokes may not be accurate enough to fully capture a player's emotions, but that performance should be a good indicator for a player's skills, and can therefore be reliably used to dynamically adjust the difficulty [11]. More in line with our research project, Bicalho et al. used K-means clustering and decision tree algorithms to identify player type profiles in a shoot 'em up game. In their work, they used in-game behavioural attributes and self-report data to identify player types [1].

2.3 Research Goal

Based on the literature discussed in the previous section, a research gap can be described and a variety of approaches can be taken. Our research bears resemblance to the work by Bicalho et al., but we will not use self-report data in our approach. Instead, we will solely rely on data that can be gathered during gameplay. Furthermore, their shoot 'em up game implementation had simpler mechanics. In our study, we would like to put more emphasis on different play styles instead, which also affected the design and implementation of our game. Various other researchers have applied player modelling to shoot 'em up games to measure player experience, frustration, and boredom, but the emphasis was never put on play styles [1, 3, 11].

Inspired by previous literature, we will apply unsupervised learning and clustering algorithms to identify play styles in our implementation of a shoot 'em up game. The research goal we aim to achieve can be defined as follows:

Identify play styles in a shoot 'em up game using in-game data.

3 Methodology

In order to achieve our research goal, several steps had to be taken. First, a player modelling approach was chosen and motivated in relation to the research goal. Then, we developed a shoot 'em up game heavily inspired by LUFTRAUSERS™. Afterwards, we recruited participants to collect data. Finally, we explored and analysed the collected data and defined the input features which would be used in the player modelling approach, on which further statistical analysis was conducted.

3.1 Player Modelling Approach

No prior instructions would be given to the participant other than the controls and general idea of the game during the data collection. The participants could choose any strategy that they deemed viable to maximise their scores (and survival time). For instance, a player could choose to focus on killing enemies to increase their chances of survival or do the same by avoiding any damage from enemy bullets. The choice of actions being at the hands of each player meant that we did not have any prior expectations of what play styles or strategies would arise from the collected data. This meant that the appropriate player modelling approach is a model-free (bottom-up) one. The idea behind this approach is to collect data of events that occur during the gameplay, identify features from the collected data that can be used to estimate or distinguish between strategic choices of the players, and finally analyse and group the data to then identify separate groupings of strategies which can be explained using the feature values corresponding to each strategic group. The grouping of the data was performed using the unsupervised learning method of K-means clustering. Now that the modelling approach has become concrete, we started on the development of the game.

3.2 Game Implementation

Our game, `Hell-bent`, was developed using the open-source `Godot` engine. Development commenced with the construction of the core gameplay loop, specifically by implementing player movement and shooting mechanics. To enhance the sense of motion, a background layer was incorporated, adding a visual element to velocity. Basic enemy agents with straightforward AI were introduced, adopting the same physics as the player character. Enemy decision-making relies on a simple targeting mechanism: they continuously turn toward the player and accelerate when the player falls within a specified forward-facing cone. Additionally, enemies attempt to fire when the player is within a similar but narrower targeting cone, though their firing rate is limited by a significantly longer cool-down period compared to the player.

Following these foundational interactions, a body of water was introduced at the lower boundary of the level. The body of water not only inflicts continuous damage to the player on contact, but also exerts an upward buoyant force. The buoyancy parameter limits the player's diving depth, preventing them from descending excessively or potentially exiting the bottom of the water area. At this stage, the camera functionality was refined to follow the player smoothly, with slight biases toward the player's facing and velocity directions, thereby improving the player's forward view. Additionally, a simple screen shake effect was implemented to emphasise impactful events, such as collisions with enemies.

A circular bar surrounding the player character was added to indicate the player's health. Notably, this bar displays health inversely by showing damage taken rather than the remaining health. This approach ensures the bar is usually invisible (as players regenerate health quickly) and minimises obstruction to the player's view. A score display was also added, derived from the survival time and the number of defeated enemies. This scoring system is not included in the player modelling approach, but simply serves as a means to signal actions and behaviours associated with successful gameplay to the player. Furthermore, basic sound effects were integrated to enhance the overall player experience. Lastly, when the player dies, they will see a screen with a "try again" button. Clicking this button will return them to the main menu screen, allowing them to start another run. A screenshot of `Hell-bent` is depicted in Appendix A.

3.3 Data Collection

To collect data during gameplay, a logger was implemented. The logger records in-game events, such as player movements and interactions, by generating a `CSV` file for each run. It begins logging when a player starts a run by creating a new file, and stops logging the moment the player dies. To avoid data loss, the logger immediately logs data to the file when an event occurs in-game.

While the player is playing the game, the following data is logged to the `CSV` file:

- Left turn, right turn, and acceleration key inputs
- Bullets fired
- Bullets missed
- Bullets hit + bullet travel time until hit
- Enemy kills
- Enemy collisions
- Damage taken source (enemy bullets, collision with enemies, or water damage) + damage taken value
- Total time of the run

Now that the logger had been fully implemented, we could start recruiting participants to collect data, but first, the study had to pass the Ethics and Privacy scan of the institution. The approved application can be viewed in Appendix I. Participants were recruited via various means. A sign-up survey with consent form was sent out online, and fellow students were approached during the seminars.

For the study, the participants had to play five to eight runs of the game. The first run was considered a practice run, which helped the participants to get used to the mechanics of the game. The other four to seven runs were used for the study. As mentioned in Section 3.2, the game logs collected data in `CSV` files. After the player had played the game at least five times, the `CSV` files generated by the game were sent to us via email. A total of 26 participants signed up for the study. 6 participants identified as female, 1 participant identified as non-binary, and 19 participants identified as male (*M* = 23.46 years old, *SD* = 3.26, range = 18–35 years old). From these participants, a total of 117 CSV files with game data were obtained. Before we could use this data, we had to perform data exploration first.

3.4 Data Exploration and Feature Generation

The data exploration and player modelling code was written in Python [17]. The data from the 117 CSV files collected during the data collection phase were visualised to check for abnormalities and outliers. This visualisation can be seen in Appendix B. As can be observed, most histograms do not contain big outliers or abnormalities, but one run contained 1547 enemies killed, which is a large outlier and should not be possible in a normal run. Therefore, this file was removed from the data set. Furthermore, a small percentage of bullets were able to travel a large distance before hitting an enemy, indicating that the player got a lucky kill off-screen. Therefore, we decided to clamp the bullet distance value to 750 to ensure the usefulness of the values. Other than that, no other pre-processing steps were taken.

Based on the collected data, we made a selection of features that could indicate play styles, meaning that we tried to omit features that are more indicative of skill than play styles. These features and their descriptions are as follows,

Average bullet time: The time each bullet travelled before hitting an enemy, averaged over each run of every player. This feature is proportional to the distance travelled by each bullet, therefore indicating whether the shot was a close range hit or a snipe.
Average healing percentage: The player starts regenerating health when not shooting. The proportion of these health regeneration windows with relation to the entire duration of the game was computed as a percentage and averaged over each run for every player.
Total bullets fired: The number of times a bullet was fired in a run.
Percentage of bullet damage: Out of the three possible damage sources (enemy bullets, collision with enemies, or water damage), the percentage of damage the player took from enemy bullets.
Percentage of collision damage: Out of the three possible damage sources (enemy bullets, collision with enemies, or water damage), the percentage of damage the player took from enemy collisions.
Total enemies killed: The total number of enemies killed in a run.
Percentage of bullet hits: The percentage of bullets that hit an enemy out of all the bullets that were fired in a run.
Total time alive: The total duration of a run.

All the features were averaged per run and not per player, as it became evident to us during data collection that various players would try to change strategies between runs. With regard to the clustering, this essentially meant that each run was considered as a different player. Since each run is assumed to be independent of each other (no carry over is possible from the previous runs other than the knowledge the player obtained while playing), it is more beneficial to not average over the runs per player, as that would lead to data loss in case the player did adjust their strategy between runs, and furthermore, this approach led to more data for the clustering algorithm to take into consideration.

3.5 K-Means Clustering

Now that all the features had been defined, the model could be created and run using the collected data, in order to determine underlying strategies that players could have potentially used. Since we decided on a bottom-up approach, and as the purpose of modelling is to isolate clusters that represent different strategies, K-means clustering seemed like an appropriate model.

Naturally, other clustering mechanisms exist. A common alternative is Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [9]. The choice of K-means over DBSCAN is due to the latter being density-based. With our game data, it is not a valid assumption that each strategy was used equally. Since the clusters represent strategies, and the player had the freedom to choose their own play style, it was certainly possible that one strategy would be more dominant than others, resulting in large density differences, which DBSCAN was not designed to handle well. Since the features we had were limited to the small scope of a simple shoot 'em up game with a limited number of actions to choose from, it was better to assume that the clusters would be somewhat globular than to assume homogenous density between the clusters.

3.5.1 The Algorithm. K-means clustering algorithm is a simple and intuitive one. The idea is to identify k clusters within the given data such that all data points can be assigned to each cluster based on the distance from the cluster centre, also more commonly known as the centroid. The Python library `scikit-learn` was used to perform K-means clustering for this research project [13, 17]. The default algorithm for this module is Lloyd's algorithm. This algorithm follows the following steps [13]:

(1) **Initialisation step:** Initialise k random centroids, say μ_k (scikit-learn uses an algorithm called `kmeans++` that optimises this initial selection based on a probabilistic distribution, but the explanation is out of the scope of this paper).
(2) **Assignment step:** Assign each data point to its nearest cluster based on the Euclidean distance to μ_k for each k.
(3) **Update step:** Update each centroid μ_k as the mean of the data points in each cluster.

Steps 2 and 3 are repeated until convergence or until a pre-defined limit of steps is reached (300 by default).

3.5.2 Dimensionality Reduction. Since we have six features, there are six dimensions across which the K-means algorithm had to cluster. Even though six dimensions can be handled by K-means without many issues (especially given the low correlation between the features as can be observed in Appendix C), to make computation more efficient and for visualisation purposes, a dimensionality reduction method such as Principal Component Analysis (PCA) can be used. One thing to keep in mind is that dimensionality reduction could lead to potential loss of interaction between features that existed within the original data. To ensure that this did not have a significant impact, the cumulative explained variance ratio was calculated. This ratio indicates how much variance of the original data is captured as you reduce the number of dimensions using PCA. The results of this computation are shown in Appendix D. The results show that with just two components, 99.5% of the variance of the original data is captured. This means that the six features could confidently be reduced to two PCA components.

3.5.3 Choice of k. Since we did not have a predetermined number of strategies we expected to find, the value of k must be identified via the means of optimisation. There are many optimisation methods such as Elbow methods, Silhouette score, Calinski-Harabasz index, Davies-Bouldin index, etc. that uses different metrics to determine the optimal value of k. We chose the Silhouette score as it is very robust for low-dimensional data. Calinski-Harabasz index works best with well-separated data, which is an assumption we could not make with the data that we had collected. Davies-Bouldin index is robust assuming the clusters are of similar sizes and densities, again an assumption that cannot be made. Elbows methods are notoriously subjective and also assume good separation of clusters. Therefore, the Silhouette score was chosen. The optimal value of k was 2 according to the Silhouette score, which was in fact the only value that obtained a score above 0.5, indicating well-separated clusters [14]. The results of the Silhouette score are shown in Appendix E. Now that the model parameters had been chosen, we could start analysing the data.

4 Results

4.1 Clustering

Figure 1 shows the cluster separation obtained after performing K-means clustering on the PCA dimensions. As can be seen in the figure, the formed clusters are well-separated and distinct. However, the PCA dimension provides little to no information on how the features are distributed across the clusters and, therefore, what strategies correspond to each cluster.

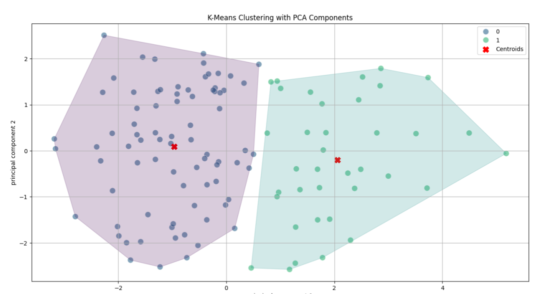

Figure 1: Clusters formed using K-means clustering.

In order to identify the relevance of the features, the PCA components must be reverse engineered first, such that each data point contains information about the original features. This was easily achieved as each PCA component was tagged with the file that the original features came from; therefore, a simple filename matching function allowed the initial features to be assigned to the appropriate cluster. Using this data, first the significance in the difference between the distribution and median of the features must be identified for each feature. Once this is established, further analysis can be conducted by observing the distribution of features to identify the underlying strategy of each cluster.

4.2 Pre-analysis Tests

In order to determine whether the significance should be determined using a parametric or non-parametric test, the normality and homogeneity were tested for using the Shapiro-Wilk and Levene's test, respectively. The results of Shapiro-Wilk test varied depending on each feature, however, Levene's test revealed that the variances are significantly different from each other, breaking the assumption of homogeneity for applying parametric tests, $F = 217, p < 0.001$. Therefore, non-parametric analysis was conducted on the results. The results of Shapiro-Wilk test can be found in Appendix F.

4.3 Statistical Analysis

For the non-parametric analysis, a Kruskal-Wallis test was chosen. Since we have two groups, this is equivalent to performing a Mann-Whitney U test. The results indicated that every feature median and distribution were significantly different between the two clusters. The results of the test can be found in Appendix G.

4.4 Strategy Analysis

The results of the Kruskal-Wallis test were plotted and displayed in Appendix H[1], for each feature. These plots allow us to determine the relation between the median of each cluster and, given that each feature is significant, allows us to determine a strategy.

Given the data from comparing the median and distribution of each feature across the clusters, the following summary can be generated:

In general, the players in Cluster 1 compared to Cluster 0,

- take more damage from enemy bullets;
- take less damage from enemy collisions;
- kill more enemies;
- fire more bullets;
- hit more bullets;
- stay alive longer.

Based on these observations, a confident assumption can be made that players assigned to Cluster 1 are generally more ***aggressive*** compared to players assigned to Cluster 0, who are more ***passive***.

5 Discussion

Given the results from the statistical analysis, combined with the observations of the feature behaviour between the clusters, we discuss whether the research goal

Identify play styles in a shoot 'em up game using in-game data.

was achieved or not.

As briefly mentioned in Section 4.4, there is a clear distinction between the clusters, as Cluster 1 players generally display a more ***aggressive*** play style whereas Cluster 0 players display a more ***passive*** play style. One thing to keep in mind is that this is a comparative analysis and therefore, this distinction only exists between the two clusters.

[1]x-axis labels are intuitively switched- 1 is on the left and 0 is on the right.

Further analysing the results shown in Appendix H, a more concrete explanation can be given for the strategies, which will be discussed separately.

5.0.1 Cluster 1. The players show an aggressive play style indicated by more bullets fired, more enemies killed, and more damage taken from enemy bullets. This indicates active engagement in battle. Taking less damage from enemy collisions can be explained by the players eliminating enemies before they are able to get close enough to collide with them. These players seem to favour killing enemies at the cost of taking damage from enemy bullets. Through these methods, the players are able to survive longer.

5.0.2 Cluster 0. The players display a much more defensive or passive play style. Firing less bullets and killing less enemies leads to them taking more damage from enemies colliding into them. They seemed focussed on minimising damage taken while compromising on killing enemies, preferring to obtain points by staying alive or letting enemies die to the body of water.

One might expect that Cluster 0 players' survival times would be longer, since they play a more defensive strategy. However, data suggests otherwise. This could be a strong indicator that given the scope of our game, a defensive strategy is much less effective than being offensive. This can be explained given that the enemies play an offensive play style, combined with their increased spawn rate over time (time between enemy spawning follows the trend defined by $5 * 0.98^t$ where t is time from start of run). Under these conditions, if enemies are not dealt with fast enough, their number keeps increasing, reducing the chances of long-term survival.

If the number of enemies was finite, a defensive strategy would have been more viable as the player can "win" by killing the finite number of enemies while avoiding taking damage (especially if the enemies play a predictable, simple strategy like in our game). However, if the goal is to survive as long as possible, with an ever-increasing number of enemies spawning, a defensive play style will not be viable for long.

5.1 Limitations

The obvious limitation of our game is the lack of features and mechanics that allow for various, drastically different play styles. Adding features such as customisation of and modifications to the player's plane, in the style of LUFTRAUSERS™, might have incentivised players to try more different play styles. Another option would be the addition of more advanced enemy AI, which would require the player to adapt their play style to survive.

Another limitation lies in the variety and amount of data. In the case of the former, most participants were friends or fellow students. Given a wider range of participants over a larger demographic would likely lead to more variety in play styles. In the case of the latter, we only had 116 CSV files (after pre-processing) worth of game data to work with, which is not a great deal compared to the work by Drachen et al. for example, as they had access to over 250,000 data points.

Lastly, not all logged events were utilised in the feature generation, as certain game mechanics we came up with during the game design and development phase were not utilised by the players. If the participants were given more time to learn the game, they might have discovered these mechanics, which would have led to more identifiable play styles.

6 Conclusion

This study aimed to identify play styles in shoot 'em up games using only in-game data. A model-free player modelling approach was defined and implemented. The approach took in-game data collected from participants and generated features that could hint at certain play styles. Then, PCA and K-means clustering were applied to identify play style clusters. The results show that there were two clusters with statistically significant differences in the feature values' medians and distributions. Upon analysing the results, the two clusters we found can be identified as ***passive*** and ***aggressive*** play styles. The study shows that there is potential in identifying play styles in shoot 'em up games using only in-game data, which could help game developers in the never-ending task of balancing and updating a game without the need to collect self-report data through means such as surveys.

References

[1] Luis Fernando Bicalho, Augusto Baffa, and Bruno Feijó. 2019. A Game Analytics Model to Identify Player Profiles in Singleplayer Games. In *2019 18th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*. 11–20. https://doi.org/10.1109/SBGames.2019.00013
[2] Gordon Calleja. 2011. *In-game.* MIT Press, London, England.
[3] BBPL de Araujo and Bruno Feijó. 2013. Evaluating dynamic difficulty adaptivity in shoot'em up games. In *Proceedings of the XII Brazilian Symposium on Games and Digital Entertainment-SBGames 2013*. 229–238.
[4] Anders Drachen, Alessandro Canossa, and Georgios N. Yannakakis. 2009. Player modeling using self-organization in Tomb Raider: Underworld. In *2009 IEEE Symposium on Computational Intelligence and Games*. IEEE, 1–8. https://doi.org/10.1109/cig.2009.5286500
[5] Anders Drachen and Matthias Schubert. 2013. *Spatial Game Analytics.* Springer London, 365–402. https://doi.org/10.1007/978-1-4471-4769-5_17
[6] Anders Drachen, Rafet Sifa, Christian Bauckhage, and Christian Thurau. 2012. Guns, swords and data: Clustering of player behavior in computer games in the wild. In *2012 IEEE Conference on Computational Intelligence and Games (CIG)*. 163–170. https://doi.org/10.1109/CIG.2012.6374152
[7] Anders Drachen, Christian Thurau, Julian Togelius, Georgios Yannakakis, and Christian Bauckhage. 2013. *Game Data Mining.* https://doi.org/10.1007/978-1-4471-4769-5_12
[8] Magy El-Nasr, Anders Drachen, and Alessandro Canossa. 2013. *Game Analytics: Maximizing the Value of Player Data.*
[9] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (Portland, Oregon) *(KDD'96)*. AAAI Press, 226–231.
[10] N.H. Frijda. 1986. *The Emotions.* Cambridge University Press. https://books.google.nl/books?id=QkNuuVf-pBMC
[11] David Halbhuber, Jakob Fehle, Alexander Kalus, Konstantin Seitz, Martin Kocur, Thomas Schmidt, and Christian Wolff. 2019. The Mood Game - How to Use the Player's Affective State in a Shoot'em up Avoiding Frustration and Boredom. In *Proceedings of Mensch und Computer 2019 (MuC'19)*. ACM. https://doi.org/10.1145/3340764.3345369
[12] OpenAI, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique Pondé de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. 2019. Dota 2 with Large Scale Deep Reinforcement Learning. (2019). arXiv:1912.06680 https://arxiv.org/abs/1912.06680
[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.
[14] Peter Rousseeuw. 1987. Rousseeuw, P.J.: Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. Comput. Appl. Math. 20, 53-65. *J. Comput. Appl. Math.* 20 (11 1987), 53–65. https://doi.org/10.1016/0377-0427(87)90125-7

[15] K.R. Scherer, A. Schorr, and T. Johnstone. 2001. *Appraisal Processes in Emotion: Theory, Methods, Research.* Oxford University Press. https://books.google.nl/books?id=10tnDAAAQBAJ

[16] Adam Smith, Chris Lewis, Kenneth Hullett, Gillian Smith, and Anne Sullivan. 2011. An Inclusive Taxonomy of Player Modeling. (04 2011).

[17] Guido Van Rossum and Fred L. Drake. 2009. *Python 3 Reference Manual.* CreateSpace, Scotts Valley, CA.

[18] Vlambeer. 2013. *LUFTRAUSERS.* https://luftrausers.com/

[19] Bryan Wirtz. 2023. *All-Time Favorites: Shoot-em-up Games.* https://www.gamedesigning.org/gaming/shoot-em-up/

[20] Georgios Yannakakis and Julian Togelius. 2011. Experience-Driven Procedural Content Generation. *Affective Computing, IEEE Transactions on* 2 (07 2011), 147–161. https://doi.org/10.1109/T-AFFC.2011.6

[21] Georgios N. Yannakakis, Pieter Spronck, Daniele Loiacono, and Elisabeth André. 2013. Player Modeling. In *Artificial and Computational Intelligence in Games*, Simon M. Lucas, Michael Mateas, Mike Preuss, Pieter Spronck, and Julian Togelius (Eds.). Dagstuhl Follow-Ups, Vol. 6. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 45–59. https://doi.org/10.4230/DFU.Vol6.12191.45

[22] Georgios N Yannakakis and Julian Togelius. 2018. *Artificial Intelligence and Games* (1 ed.). Springer International Publishing, Cham, Switzerland.

A Screenshot of `Hell-bent`

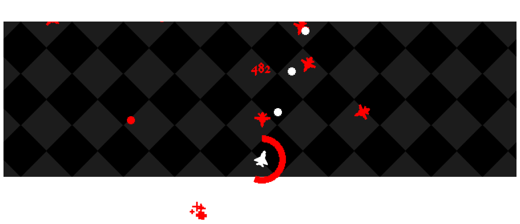

Figure 2: Screenshot of `Hell-bent` during a run. The player (white plane) has a circular inverse health bar around itself. The player's bullets are white and the enemies' (red planes) bullets are red. The bottom half of the screen is covered by a white body of water. At the top of the screen, the current score is displayed.

B Occurrences of Values in the Collected Data

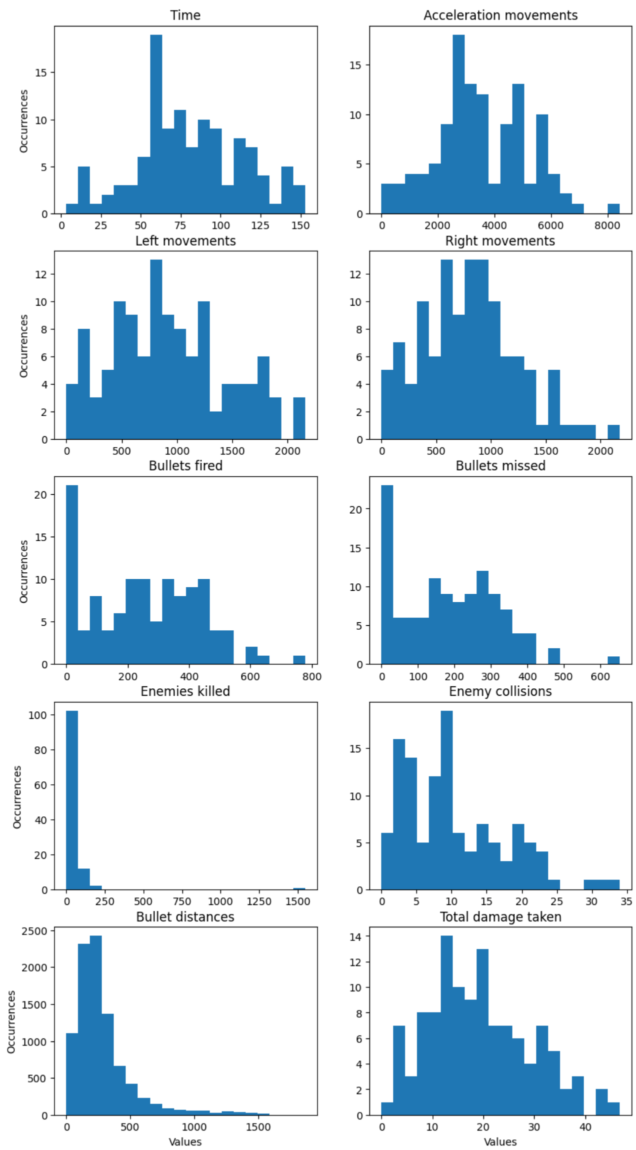

Figure 3: Histograms showing the occurrence counts of all collected data values.

C Features Correlation Matrix

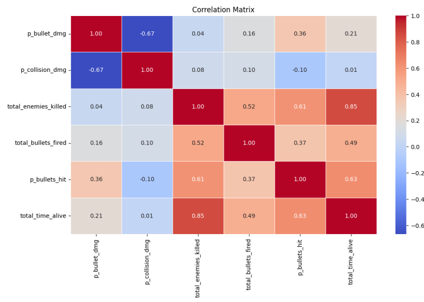

Figure 4: Correlation matrix of the six features.

D Cumulative Explained Variance

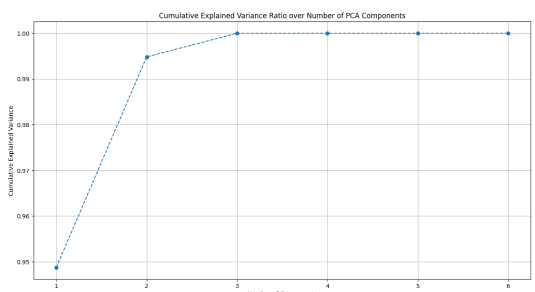

Figure 5: Cumulative explained variance for different number of PCA components.

E Silhouette Score

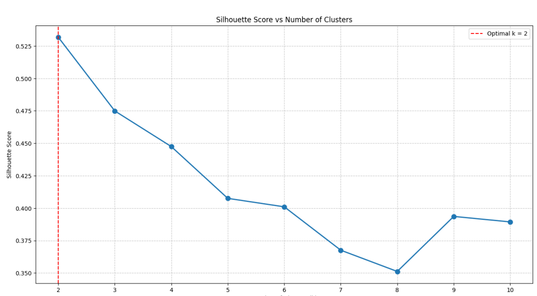

Figure 6: Silhouette score for different values of *k*.

F Shapiro-Wilk Test Results

Feature	W-statistic	p-value
Bullet damage	0.97	0.013
Collision damage	0.98	0.203
Total enemies killed	0.85	<0.001
Total bullets fired	0.96	<0.001
Bullets hit	0.86	<0.001
Total time alive	0.98	0.157

G Kruskal-Wallis Test Results

Feature	H-statistic	p-value
Bullet damage	13.7	<0.001
Collision damage	6.06	0.01
Total enemies killed	63.1	<0.001
Total bullets fired	26.3	<0.001
Bullets hit	56.7	<0.001
Total time alive	65.8	<0.001

H Kruskal-Wallis Test Plots

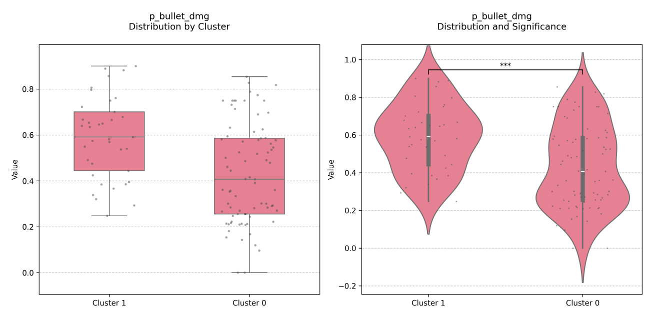

Figure 7: Kruskal-Wallis test result and distribution for bullet damage.

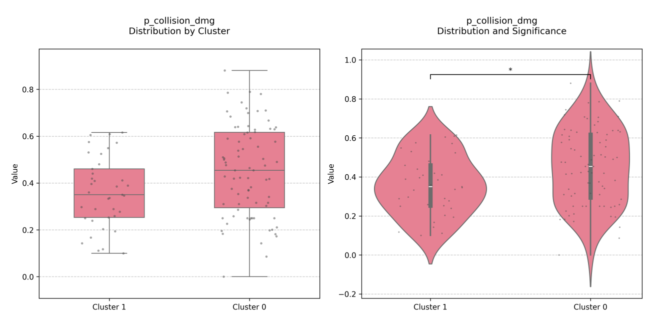

Figure 8: Kruskal-Wallis test result and distribution for collision damage.

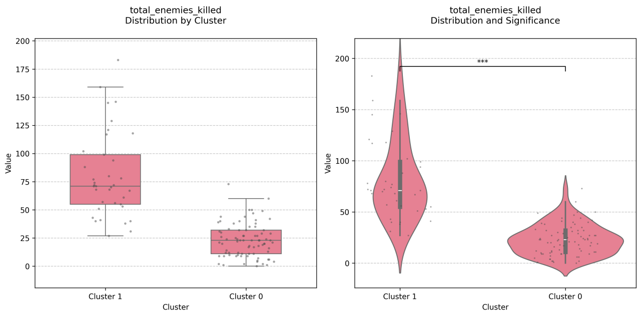

Figure 9: Kruskal-Wallis test result and distribution for total enemies killed.

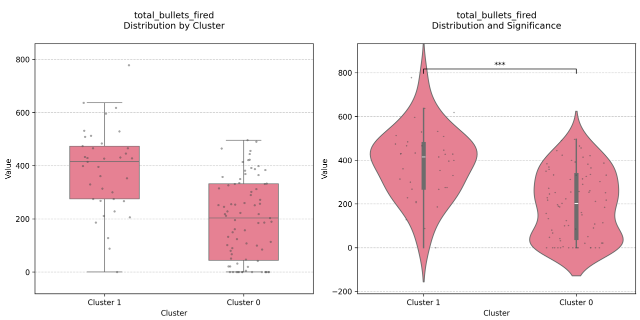

Figure 10: Kruskal-Wallis test result and distribution for total bullets fired.

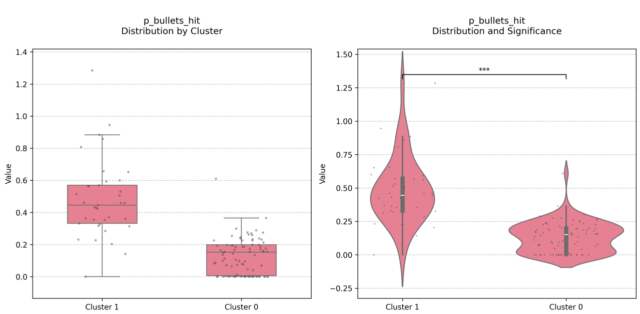

Figure 11: Kruskal-Wallis test result and distribution for bullets hit.

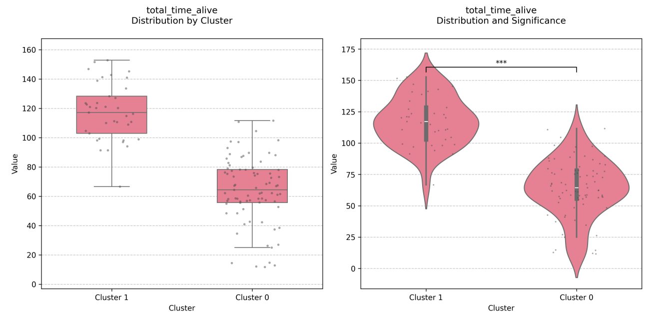

Figure 12: Kruskal-Wallis test result and distribution for the total time alive.

I Ethics and Privacy Scan Application

Section 1. Research projects involving human participants

P1. Does your project involve human participants? This includes for example use of observation, (online) surveys, interviews, tests, focus groups, and workshops where human participants provide information or data to inform the research. If you are only using existing data sets or publicly available data (e.g. from X, Reddit) without directly recruiting participants, please answer no.
Yes

Recruitment
P2. Does your project involve participants younger than 16 years of age?
No
P3. Does your project involve participants with learning or communication difficulties of a severity that may impact their ability to provide informed consent?
No
P4. Is your project likely to involve participants engaging in illegal activities?
No
P5. Does your project involve patients?
No
P6. Does your project involve participants belonging to a vulnerable group, other than those listed above?
No
P8. Does your project involve participants with whom you have, or are likely to have, a working or professional relationship: for instance, staff or students of the university, professional colleagues, or clients?
Yes
P9. Is it made clear to potential participants that not participating will in no way impact them (e.g. it will not directly impact their grade in a class)?
Yes

Informed consent
PC1. Do you have set procedures that you will use for obtaining informed consent from all participants, including

(where appropriate) parental consent for children or consent from legally authorized representatives? (See suggestions for information sheets and consent forms on the website.)
Yes
PC2. Will you tell participants that their participation is voluntary?
Yes
PC3. Will you obtain explicit consent for participation?
Yes
PC4. Will you obtain explicit consent for any sensor readings, eye tracking, photos, audio, and/or video recordings?
Not applicable
PC5. Will you tell participants that they may withdraw from the research at any time and for any reason?
Yes
PC6. Will you give potential participants time to consider participation?
Yes
PC7. Will you provide participants with an opportunity to ask questions about the research before consenting to take part (e.g. by providing your contact details)?
Yes
PC8. Does your project involve concealment or deliberate misleading of participants?
No

Section 2. Data protection, handling, and storage

The General Data Protection Regulation imposes several obligations for the use of **personal data** (defined as any information relating to an identified or identifiable living person) or including the use of personal data in research.
D1. Are you gathering or using personal data (defined as any information relating to an identified or identifiable living person)?
Yes

High-risk data
DR1. Will you process personal data that would jeopardize the physical health or safety of individuals in the event of a personal data breach?
No
DR2. Will you combine, compare, or match personal data obtained from multiple sources, in a way that exceeds the reasonable expectations of the people whose data it is?
No
DR3. Will you use any personal data of children or vulnerable individuals for marketing, profiling, automated decision-making, or to offer online services to them?
No
DR4. Will you profile individuals on a large scale?
No
DR5. Will you systematically monitor individuals in a publicly accessible area on a large scale (or use the data of such monitoring)?
No
DR6. Will you use special category personal data, criminal offense personal data, or other sensitive personal data on a large scale?
No
DR7. Will you determine an individual's access to a product, service, opportunity, or benefit based on an automated decision or special category personal data?
No
DR8. Will you systematically and extensively monitor or profile individuals, with significant effects on them?
No
DR9. Will you use innovative technology to process sensitive personal data?
No

Data minimization
DM1. Will you collect only personal data that is strictly necessary for the research?
Yes
DM4. Will you anonymize the data wherever possible?
Yes
DM5. Will you pseudonymize the data if you are not able to anonymize it, replacing personal details with an identifier, and keeping the key separate from the data set?
Yes

Using collaborators or contractors that process personal data securely
DC1. Will any organization external to Utrecht University be involved in processing personal data (e.g. for transcription, data analysis, data storage)?
No

International personal data transfers
DI1. Will any personal data be transferred to another country (including to research collaborators in a joint project)?
No

Fair use of personal data to recruit participants
DF1. Is personal data used to recruit participants?
No

Participants' data rights and privacy information
DP1. Will participants be provided with privacy information? (Recommended is to use as part of the information sheet: For details of our legal basis for using personal data and the rights you have over your data please see the University's privacy information at www.uu.nl/en/organisation/privacy.)
Yes
DP2. Will participants be aware of what their data is used for?
Yes
DP3. Can participants request that their personal data be deleted?
Yes

DP4. Can participants request that their personal data be rectified (in case it is incorrect)?
Yes
DP5. Can participants request access to their personal data?
Yes
DP6. Can participants request that personal data processing is restricted?
Yes
DP7. Will participants be subjected to automated decision-making based on their personal data with an impact on them beyond the research study to which they consented?
No
DP8. Will participants be aware of how long their data is being kept for, who it is being shared with, and any safeguards that apply in case of international sharing?
Yes
DP9. If data is provided by a third party, are people whose data is in the data set provided with (1) the privacy information and (2) what categories of data you will use?
Not applicable

Using data that you have not gathered directly from participants
DE1. Will you use any personal data that you have not gathered directly from participants (such as data from an existing data set, data gathered for you by a third party, data scraped from the internet)?
No

Secure data storage
DS1. Will any data be stored (temporarily or permanently) anywhere other than on password-protected University authorized computers or servers?
No
DS4. Excluding (1) any international data transfers mentioned above and (2) any sharing of data with collaborators and contractors, will any personal data be stored, collected, or accessed from outside the EU?
No

Section 3. Research that may cause harm

Research may cause harm to participants, researchers, the university, or society. This includes when technology has dual-use, and you investigate an innocent use, but your results could be used by others in a harmful way. If you are unsure regarding possible harm to the university or society, please discuss your concerns with the Research Support Office.
H1. Does your project give rise to a realistic risk to the national security of any country?
No
H2. Does your project give rise to a realistic risk of aiding human rights abuses in any country?
No
H3. Does your project (and its data) give rise to a realistic risk of damaging the University's reputation? (E.g., bad press coverage, public protest.)
No
H4. Does your project (and in particular its data) give rise to an increased risk of attack (cyber- or otherwise) against the University? (E.g., from pressure groups.)
No
H5. Is the data likely to contain material that is indecent, offensive, defamatory, threatening, discriminatory, or extremist?
No
H6. Does your project give rise to a realistic risk of harm to the researchers?
No
H7. Is there a realistic risk of any participant experiencing physical or psychological harm or discomfort?
No
H8. Is there a realistic risk of any participant experiencing a detriment to their interests as a result of participation?
No
H9. Is there a realistic risk of other types of negative externalities?
No

Section 4. Conflicts of interest

C1. Is there any potential conflict of interest (e.g. between research funder and researchers or participants and researchers) that may potentially affect the research outcome or the dissemination of research findings?
No
C2. Is there a direct hierarchical relationship between researchers and participants?
No

Section 5. Your information.

This last section collects data about you and your project so that we can register that you completed the Ethics and Privacy Quick Scan, sent you (and your supervisor/course coordinator) a summary of what you filled out, and follow up where a fuller ethics review and/or privacy assessment is needed. For details of our legal basis for using personal data and the rights you have over your data please see the University's privacy information. Please see the guidance on the ICS Ethics and Privacy website on what happens on submission.
Z0. Which is your main department?
Information and Computing Science
Z1. Your full name:
Jin Kai Huang
Z2. Your email address:
j.k.huang@students.uu.nl
Z3. In what context will you conduct this research?
As a student on a course with course coordinator: Dr. Julian Frommel
Z6. Email of the course coordinator or supervisor (so that we can inform them that you filled this out and provide them with a summary):
j.frommel@uu.nl

Z8. Title of the research project/study for which you filled out this Quick Scan:
AI4GT Player Types Study
Z9. Summary of what you intend to investigate and how you will investigate this (200 words max):
In our project, we are creating a game inspired by shoot 'em up game Luftrausers. In the game, the player controls an airplane that can only fire from the front, meaning that hitting enemies requires a combination of movement and aim. The scoring system will be based on how long the player stays alive and how many opponents the player is able to defeat. We plan on modelling the playstyle of players by collecting various features while they are playing. Using these features, we want to use an unsupervised learning approach to classify different player types. For example, we think that players that gain a higher percentage of their total score from staying alive might be classified as pacifists, while someone that has snappy aim, high accuracy, and mostly scores by defeating enemies might be classified as a sharpshooter.

Scoring

- Privacy: 0
- Ethics: 0

Received 25 October 2024; revised 8 November 2024; accepted 8 November 2024