INFOMAIGT - Group 8

Vlad-Cristian Puscaru
v.c.puscaru@students.uu.nl
5248973

Vasco Fabiani
v.fabiani@students.uu.nl
2262452

Mehmet Kaan Ozkan
m.k.ozkan@students.uu.nl
5251710

Stavros Spyrou
s.spyrou@students.uu.nl
1568396

Mauro Vazquez
m.vazquezbassat@students.uu.nl
6718809

Themistoklis Andreopoulos
t.andreopoulos@students.uu.nl
0775223

ACM Reference Format:
Vlad-Cristian Puscaru, Vasco Fabiani, Mehmet Kaan Ozkan, Stavros Spyrou, Mauro Vazquez, and Themistoklis Andreopoulos. 2024. INFOMAIGT - Group 8. In . ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

The project aims to analyze player behavior in a Top-Down 2D shooter game and predict the perceived satisfaction. To achieve this, the project consists of the following key components:

- **Top-Down 2D Shooter Game Prototype**: The game mechanics, play style, development, and other characteristics are detailed in Section 3. This prototype serves as one of the environments used to collect player data.
- **Player Experience Questionnaire**: The aim is to asses the player's enjoyment for each level through the use of a questionnaire. This process will be guided by the paper "Development and Validation of the Player Experience Inventory: A Scale to Measure Player Experiences at the Level of Functional and Psychosocial Consequences" by Abeelea et al. More details can be found on **Section 5**.
- **AI Model for Analyzing and Predicting Player Experience**: A supervised AI model will be trained using aggregated player data, including gameplay metrics and responses from the Player Experience Questionnaire. The implementation details are further elaborated in Section 5.

The objective is to structure the experiment in two phases.
The first phase will be focused on building the game prototype, asses its gameplay and use it as a data collection framework.
The second phase envisions a trained AI Model which is able to predict a player's perceived experience by his behaviour in game. This opens up a field of possibilities in which the predictions can be used to further enhance that experience.
However, at the moment of writing, because of time limitations, instead of making the game more adaptable, the goal of phase 2 will be to continue evaluating the AI model against new data sets.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 Related Work

Player behavior and enjoyment research using gameplay metrics such as challenge level, performance feedback, and engagement factors is supported in video game studies. Klimmt et al. [2] examined how perceived difficulty and perceived success of task completion relate to game enjoyment, which the flow theory supports, emphasizing moderate challenge as the optimal level of enjoyment. Their work also illustrates that there are possible adjustments regarding perceived difficulty and success in order to maintain enjoyment, especially in cases where difficulty levels influence success attribution. This observation becomes an important cornerstone for AI-based models on predicting player satisfaction because the integration of challenge-adjusting mechanisms will surely improve predictive accuracy.

A number of works have been done on predictive modeling regarding the enjoyment and performance of players. For instance, Drachen et al. [1] have used telemetry data recorded directly within the game to classify patterns in the behavior of players, increasing predictive capabilities of models predicting player satisfaction. Another approach from Nacke et al. [3] addressed the design of the PXI-a scientifically validated questionnaire of subjective experience, including enjoyment and engagement. Both studies encourage the use of subjective questionnaire data and objective gameplay metrics in predictive models.

3 Game

Our game is a 2D Top-Down Puzzle and Precision Shooter that challenges players to control a blob capable of shooting parts of its mass to eliminate immobile targets across a series of timed levels. The game blends puzzle-solving with quick reflexes, where players must carefully manage their blob's size and mass, while figuring out the optimal sequence of actions to succeed. Levels are designed to encourage both strategic planning and fast execution, with an easy retry button to allow for multiple attempts and experimentation.

3.1 Structure and Goals

The game is structured around a level-based system, where each level contains immobile targets that must be destroyed by shooting mass at them. Players are encouraged to complete each level as quickly as possible, with a timer displaying their progress and time taken. Each level also includes puzzle elements that require players to determine the correct sequence of actions, such as managing their blob's size and taking advantage of the environment. The blob can shoot its own mass to eliminate targets but shrinks as it does so. To maintain its mass and continue shooting, the player can pick up scattered bullets to regain size.

3.2 Core Mechanics

The primary mechanic of the game is shooting mass, which reduces the player's size. However, the player can regain mass by picking up bullets that are placed throughout the level. A unique mechanic is teleportation: the player can shoot an existing mass (either shot by the player or already in the level), and if the combined size of the player's mass and the existing mass is greater than the player's current size, the player teleports to the larger blob's location and "becomes" the larger blob. This creates opportunities for strategic positioning and fast movement around the map.

3.3 Wall Types

Different types of walls introduce a variety of challenges and possibilities for how players can interact with the environment:

- **Regular walls**: Block both the player and bullets, creating clear boundaries and obstacles.
- **Bouncy walls**: Allow bullets to bounce off them, introducing the potential for trick shots or more complex projectile trajectories.
- **Absorbing walls**: Destroy any bullets that touch them, limiting where the player can shoot. These walls also absorb the player's mass if touched, adding further risk.
- **Player-delimiting walls**: Block the player's movement but allow bullets to pass through, enabling strategic shooting while restricting mobility.
- **Bullet-delimiting walls**: Allow the player to pass through but block bullets, requiring careful consideration of shooting angles and movement.

3.4 Gameplay Depth

Advanced mechanics include moving bullets, which add a timing element to certain levels. For instance, a bullet may slowly bounce toward an absorbing wall, forcing the player to collect it before it is destroyed. Additionally, the size of the bullets varies, providing the player with more or less ammunition depending on the level design.

These mechanics, combined with the teleportation ability and different wall types, create a wide variety of potential level designs, ranging from fast-paced reflex challenges to complex puzzles requiring precise planning and execution.

Despite its simple core mechanics, our game offers a deep and flexible gameplay experience. Players must balance fast decision-making with careful strategy, as they manage their mass, teleport across levels, and deal with the constraints of the environment. This creates a dynamic, evolving challenge where even basic elements can result in intricate, satisfying solutions.

Potential future additions to the game could include different types of shots or one-way collision walls, further enhancing the mechanics and creating even more emergent gameplay. However, for the scope of this project, the game was left as is.

4 Player Modeling Approach

The primary goal of this project is to analyze player behavior within the game we have created and predict the perceived satisfaction of players. To achieve this, we developed a player modeling approach that focuses on capturing gameplay metrics and subjective feedback from players through two distinct data sources: gameplay event data and questionnaire responses. This dual approach enables us to gather a comprehensive view of player experience

4.1 Objectives of the Player Model

The player model is designed to identify relationships between gameplay mechanics and player satisfaction, enabling the prediction of perceived experience based on player actions. Key gameplay features, such as accuracy, time to complete levels, and health percentage, are chosen for their relevance to the player's experience. By pairing these objective metrics with subjective feedback from the Player Experience Questionnaire, we aim to build a model capable of predicting player satisfaction and engagement.

4.2 Data Categories

The model relies on two primary categories of data:

(1) **Player Experience Questionnaire Responses**: This captures subjective feedback on enjoyment, engagement, pacing, and flow, reflecting players' personal assessment of each level.
(2) **Gameplay Metrics**: Observed metrics are logged automatically within the game, including aspects like shots fired, time taken, and health remaining. This data provides an objective record of player behavior that can be analyzed to identify patterns associated with varying levels of player satisfaction.

By modeling these data categories, we can develop a supervised AI model that learns to predict the player's perceived experience based on in-game behaviors.

5 Data Collection and Creating the Model

As mentioned above, we collect 2 distinct categories of data from the player:

(1) User inserted data through the use of a Player Experience Questionnaire
(2) User behavior throughout the level, collected by in-game mechanism

The data is collected after each level and it is sent via e-mail to the authors (See **Apendix A**). This process is automatized and implemented directly in the game using the MailJet API [**?**]. Approval for data collection is requested in advance. [1]

5.1 Player Perceived Experience. Data collection via Questionnaire

In order to get the player's perceived experience, the questionnaire (described further in this section) was implemented in game and it is required for the player to complete it in order to advance to the next level.

5.2 Player Behaviour. Data Collection via Event-Driven Mechanism

To track player actions, we implemented an event-based approach. For instance, a `shoot` event is fired whenever the player presses the

[1]Players are notified about data collection as part of the tutorial level. See Section about game.

button to shoot. This event is captured in the player's script and broadcast to various listeners. Each listener responds by updating the relevant statistics (e.g., shot count, accuracy, rate of fire).

For each type of player action (shooting, jumping, taking damage, etc.), a corresponding event is defined in the player's script. External modules such as the statistics logger subscribe to these events and handle the data collection.

This modular architecture allows us to isolate the responsibilities of different game systems.

5.3 Data Structuring and Storage

The data collected during gameplay is stored in a custom class called `PlayerStats`, which includes fields such as:

- Total time to complete the level
- Accuracy (shots hit vs. shots fired)
- Health percentage remaining
- Idle time, power-ups collected, and movement patterns

This `PlayerStats` object accumulates data during gameplay and is sent after each level, through e-mail, as described earlier and allows for efficient handling and storage of data.

5.4 Feature Generation from Raw Data

We designed a set of gameplay metrics that effectively capture player behavior during the game. These metrics were chosen based on their ability to reflect the player's interaction with the game mechanics and their potential influence on the player's perceived experience. Here's a detailed explanation of the features and how they contribute to the player experience model:

(1) **Total Time to Clear a Level**: This measures the amount of time (in seconds) a player takes to finish a level. It serves as a proxy for the difficulty level, as longer times might indicate a more challenging experience or a more strategic approach by the player.
(2) **Total Number of Shots Fired**: This metric records the total number of shots the player takes. A higher number of shots could signify intense engagement with the game, whereas fewer shots might indicate passive gameplay or reliance on non-combat strategies.
(3) **Accuracy of Shots (Shots Hit / Shots Fired)**: Accuracy is calculated as the ratio of shots that hit a target to the total number of shots fired. Higher accuracy may suggest a high skill level or optimal game difficulty, while lower accuracy might indicate either a challenging experience or player inexperience.
(4) **Health Percentage Left at the End of the Level**: This measures the remaining health percentage when the level is completed. A high remaining health percentage might suggest that the player found the level easier, while a low percentage could imply the player struggled or the level was more intense.
(5) **Number of Power-Ups Picked Up**: The number of power-ups collected reflects the player's engagement with the game and their understanding of the game's resource system. Collecting more power-ups may suggest that the player is actively searching for in-game advantages.
(6) **Movement Direction and Distance**: We log the number of ticks where the player presses movement keys (right/left-/up/down). This metric provides insights into the player's mobility and strategy. For instance, a player who moves frequently might be actively engaged in exploration or combat, while less movement could indicate a more defensive or cautious playstyle.
(7) **Percentage of Shots Fired While Moving or Standing Still**: This feature records how much of the combat happens while the player is moving versus standing still. Players who fire more while moving might exhibit a more aggressive or action-oriented playstyle, while standing still and firing could suggest a more methodical or cautious approach.
(8) **Idle Time Before Starting the Level**: This measures the time a player remains idle before initiating the first movement at the start of the level. It can indicate hesitation, strategizing, or player disengagement.
(9) **Time Spent in Different Areas of the Map**: We tracked how much time the player spent in specific areas of the map (e.g., behind walls, in corners, or in open spaces). This feature helps us understand the player's tactical decisions, such as taking cover, exploring, or engaging in combat in specific areas.
(10) **Average Distance from Target When Shooting**: This metric captures the average distance between the player and their target when shooting. It provides insights into whether the player prefers close-range or long-range combat, which can reflect their comfort level with the game mechanics.
(11) **Damage Taken per Level**: This metric tracks the total damage the player receives in each level. It provides a measure of the level's challenge or the player's ability to avoid damage.
(12) **Number of Tries per Level**: The number of attempts required to complete a level reflects its difficulty. A higher number of retries indicates a more challenging experience for the player.

5.5 Model Creation

Once the data was collected, the next step was to pair these features with player feedback to create a supervised learning model. The goal was to predict perceived satisfaction based on the collected metrics.

5.5.1 *Player Experience Questionnaire.*

After each level, the player completed a Player Experience Questionnaire, providing subjective feedback on their experience.The questions included were:

- **Q1:** How enjoyable did you find this level? (1-5 scale)
- **Q2:** How engaging was the level for you? (1-5 scale)
- **Q3:** How would you rate the pacing of the level? (1-5 scale)
- **Q4:** Did the level allow you to maintain a sense of flow? (1-5 scale)

To compute an overall player experience score based on the responses to the questionnaire, we opted to use a weighted average. This method allows us to assign different levels of importance to each question, reflecting the relative significance of various aspects of the player's experience. The four questions in the questionnaire address different components of player experience: enjoyment, engagement, pacing, and flow. The assigned weights are as follows:

- **Q1 (Enjoyability)**: 50% (0.50)
- **Q2 (Engagement)**: 20% (0.20)
- **Q3 (Pacing)**: 15% (0.15)
- **Q4 (Flow)**: 15% (0.15)

To compute the overall experience score, we used the following formula:

$$\text{Overall Experience Score} = (Q1 \times w_1) + (Q2 \times w_2) + (Q3 \times w_3) + (Q4 \times w_4)$$

Where:

- $Q1, Q2, Q3, Q4$ represent the scores provided by the player for each question (on a scale from 1 to 5).
- w_1, w_2, w_3, w_4 are the respective weights assigned to each question.

We chose to prioritize enjoyment (**Q1**) with the highest weight (50%) because it is the most critical factor in assessing the success of a game level. Engagement (**Q2**) follows at 20%, as it reflects how captivated the player was during the experience. Pacing (**Q3**) and flow (**Q4**) were weighted equally at 15%, as these factors, while important, are considered more secondary to the core enjoyment and engagement metrics.

5.5.2 Data Preprocessing. Given that the data originated in a JSON format, we initially loaded and flattened it for easier analysis in `pandas`. Here are the specific steps taken:

- **JSON Data Loading**: The data was loaded from a JSON file and converted into a `pandas` DataFrame. This involved iterating over each player's data and compiling gameplay metrics and survey responses.
- **Filtering Unnecessary Data**: Testing entries were were removed to exclude non-essential records from analysis.
- **Level Extraction**: From the `level` column, the specific level identifier (e.g., "Level10") was extracted by removing the `.unity` suffix and file path.
- **Accuracy Calculation**: Accuracy was calculated as the ratio of `totalShotsHit` to `totalShotsFired`. Missing values were handled by filling NaN with 0 to avoid errors, as a bug in the data originally caused missing values.
- **Overall Enjoyment Calculation**: A composite `overall_enjoyment` score was derived as a weighted combination of enjoyment metrics (Enjoyment, Engagement, Pacing, and Flow) as explained before
- **Duplicate Removal**: Duplicate records were removed based on `name`, `level`, and `totalTimeToClearLevel` to ensure unique entries for each player-level combination.
- **Offset Correction**: All values in the `numberOfTries` column were incremented by 1 since the original data started counting from zero.
- **Capping High Values**: For values in `numberOfTries` exceeding 35, these were replaced with the average number of tries for values under this threshold to address inconsistencies and potential data entry errors, as the data was affected by a bug.

Outliers were handled in multiple ways to evaluate the impact on model performance:

1. **Z-Score Method**: Removed outliers with Z-scores beyond a threshold of 3.
2. **IQR Method**: Filtered out records beyond 1.5 times the IQR from the 25th and 75th percentiles.
3. **Percentile Capping**: Capped extreme values at the 1st and 99th percentiles.

These three datasets allowed for experimentation with different versions of outlier handling to assess which method yielded better model performance.

5.5.3 Feature Engineering. After cleaning and preprocessing the data, we transformed the raw gameplay metrics into features that were more informative for our predictive model. As described in Section 5.4 , these features included metrics such as:

- Total time to clear the level.
- Accuracy of shots.
- Health percentage left.
- Movement patterns and engagement in different areas of the map.
- Number of retries per level, etc.

These features were paired with the responses from the player experience questionnaire (enjoyment, engagement, pacing, and flow ratings) to serve as the labels for the model.

5.5.4 Model Selection and Training. We experimented with several machine learning algorithms to determine which model performed best for predicting player experience. The primary models considered were:

- **Random Forest**: A robust and interpretable model that works well with both numerical and categorical data.
- **Gradient Boosting Regressor**: A more powerful model that builds on weak learners to minimize prediction errors and handle complex relationships in the data.
- **Support Vector Regression (SVR)**: A regression model that extends Support Vector Machines (SVMs) to predict continuous values, such as player experience scores. SVR aims to minimize the prediction error by creating a margin of tolerance and predicting values that fall as close as possible to the true scores.

The chosen models were trained using a supervised learning approach, with the gameplay features as the input data and the questionnaire responses as the target variables. For each target variable (e.g., enjoyment, engagement, pacing), a separate model was trained. We utilized 80% of the data for training and reserved 20% for testing to evaluate the model's performance.

We attempted stacking by combining the above models as base models with an XGBoostRegressor as the meta-model. This stacked model aimed to capture different aspects of the data by leveraging each base model's strengths. However, it was found that the simpler Gradient Boosting Regressor model was more efficient without requiring the added complexity of stacking.

6 Evaluation

After extensive testing, the Gradient Boosting Regressor was selected as the final model. We trained this model on the capped dataset (also selected as the best dataset), and evaluation on the test set yielded the following metrics:

- **Mean Absolute Error**: 0.782
- **R^2 Score**: 0.106

These results indicate that the model's performance is somewhat limited, as shown by the low R^2 score. This suggests that the model explains only a small portion of the variance in player experience. One potential reason for this mediocre performance is that the game itself may be too complex to establish a clear correlation between gameplay metrics and perceived satisfaction. The game's combination of intricate mechanics and diverse gameplay elements may require a larger dataset to effectively capture meaningful patterns in player behavior.

6.1 Feature Importance Analysis

To further understand the model's predictions, we conducted an analysis of feature importance. The most significant features and their respective importances are shown in Table 1.

Feature	Importance
Accuracy	0.365
Total Time to Clear Level	0.359
Idle Time Before Level	0.110
Total Shots Hit	0.077
Shots While Moving	0.040
Number of Tries	0.028
Resource Collected	0.015
Total Shots Fired	0.005
Health Left at End	0.000
Damage Taken	0.000

Table 1: Feature importance values for the final Gradient Boosting Regressor model.

The analysis reveals that `accuracy` and `totalTimeToClearLevel` are the most influential features in predicting player experience. However, features like `healthLeftAtEnd` and `damageTaken` have zero importance, suggesting that they have minimal or no impact on the model's predictions. This indicates that not all metrics collected may contribute meaningfully to the model, possibly due to a limited dataset or weak correlations with player satisfaction.

7 Discussion

The model grants a good insight into what features contribute the least and the most to enjoyment in a video game. In a simple, top-down shooter we can see that `accuracy` is the most significant features, indicating that having an engaging game is important to satisfaction. This reveals that players are satisfied as they hit the marks, and feel discouraged or frustrated when they miss the mark, whether it is due to their poor aim or difficulty in bouncing the shots or solving the puzzles in various scenarios in the levels.

A second critical feature was the `totalTimeToClearLevel`, which reveals that levels that take too long, or the player is unable to solve the puzzles directly affect their experience, again showing that players are likely to be disappointed if the levels take too long for various reasons.`idleTimeBeforeLevel` wasn't very impactful but still contributed to a meaningful degree. This feature is correlated with level difficulty as the player is expected to analyze surroundings before attempting in harder levels. However, the player could be idle for various other reasons, and `totalTimeToClearLevel` or `numberOfTries` are a better indicator of difficulty.

On the other hand, there are certain features that would be expected to have greater impact than they did. It is surprising that `numberOfTries` did not have a significant impact, this is an interesting finding that it is not the number of attempts in the process of trial and error, but the total time taken to complete a level that significantly affects a player's experience.

On a similar note, `healthLeftAtEnd` and `damageTaken` also do not affect much, although we would expect them to be correlated with the difficulty of the level, and expect an emotional response when a player wins with a small amount of health, or after great struggle. One possible explanation for this is that health is a finite resource, and thus players that managed to actually complete the level would have similar values for these features.

Finally, it is expected that `totalShotsFired` and `totalShotsHit` do not impact the model much, attacking is a finite resource in this game that uses health, so there is not much variety in the data here.

7.1 Limitations and Future Work

The limited amount of data likely contributed to the model's reduced accuracy, as machine learning models generally require extensive datasets to capture complex patterns. Additionally, the intricate nature of the game mechanics may demand a more nuanced model, potentially involving more sophisticated feature engineering or alternative algorithms. Future work could focus on gathering a larger dataset, refining the feature set, and exploring advanced modeling techniques to improve predictive performance.

8 Conclusion

The project was successful in formalizing an abstract concept such as *enjoyment* in video games and running tests to isolate and try to predict the sources affecting it negatively and positively. This was accomplished by combining a questionnaire after each level with various metrics gathered from the player model during their gameplay experience. Our model grants insight into what makes video games fundamentally enjoyable and engaging, and opens the door for future work on optimizing games to be more satisfactory experiences, which is a common goal whether the objective is for recreational purposes, or educational purposes for a game.

9 Conclusion

At the time of writing this section is still ongoing.

References

[1] 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

[2] Christoph Klimmt, Cynthia Blake, Dorothe Hefner, Peter Vorderer, and Christian Roth. 2009. Player Performance, Satisfaction, and Video Game Enjoyment. In *Entertainment Computing – ICEC 2009 (Lecture Notes in Computer Science, Vol. 5709)*, Stephane Natkin and Jean-Claude Dupire (Eds.). Springer, Berlin, Heidelberg, 1–12. https://doi.org/10.1007/978-3-642-04052-8_1

[3] Lennart E Nacke, Anders Drachen, and Stefan Göbel. 2010. Methods for evaluating gameplay experience in a serious gaming context. *International Journal of Computer Science in Sport* 9, 2 (2010), 1–12.

Appendices

A Example of data structure sent via e-mail


```
{
    "name": "NOT-FILLED",
    "level": "Assets/Scenes/Levels/Level3.unity",
    "inGameData": {
        "totalTimeToClearLevel": 39.22,
        "healthLeftAtEnd": 30,
        "totalShotsFired": 29,
        "totalShotsHit": 7,
        "shotsWhileMoving": 0,
        "idleTimeBeforeLevel": 1.40289307,
        "numberOfTries": 0,
        "resourceCollected": 12,
        "damageTaken": 10,
        "accuracy": 0
    },
    "questionAnswers": {
        "How enjoyable did you find this level?": 4,
        "How engaging was the level for you?": 4,
      "How would you rate the pacing of the level?": 3,
     "Did the level allow you to maintain a sense of flow? ": 3
    }
}
```