Player Modeling for Dynamic Difficulty Adjustment in Top Down Action Games

Thesis Presented
by
Guillermo Romera Rodriguez
to
The College of Arts, Media and Design

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Game Science and Design

Northeastern University
Boston, Massachusetts
May, 2019
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Figures</td>
<td>5</td>
</tr>
<tr>
<td>List of Tables</td>
<td>8</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>9</td>
</tr>
<tr>
<td>2 Dynamic Difficulty Adjustment</td>
<td>13</td>
</tr>
<tr>
<td>3 Player Modeling</td>
<td>18</td>
</tr>
<tr>
<td>4 Domain</td>
<td>21</td>
</tr>
<tr>
<td>4.1 Pommerman</td>
<td>21</td>
</tr>
<tr>
<td>5 First Study</td>
<td>23</td>
</tr>
<tr>
<td>5.1 Goals</td>
<td>23</td>
</tr>
<tr>
<td>5.2 Methodology</td>
<td>24</td>
</tr>
<tr>
<td>5.2.1 Survey</td>
<td>24</td>
</tr>
<tr>
<td>5.2.2 Set Up</td>
<td>25</td>
</tr>
<tr>
<td>5.2.3 Data Collection</td>
<td>28</td>
</tr>
<tr>
<td>5.2.4 Data Analysis</td>
<td>30</td>
</tr>
<tr>
<td>5.2.5 Clustering</td>
<td>30</td>
</tr>
<tr>
<td>5.2.6 Evaluation Methods</td>
<td>33</td>
</tr>
<tr>
<td>5.3 Results</td>
<td>35</td>
</tr>
<tr>
<td>5.3.1 Discussion</td>
<td>40</td>
</tr>
<tr>
<td>6 Second Study</td>
<td>46</td>
</tr>
<tr>
<td>6.1 Goals</td>
<td>46</td>
</tr>
</tbody>
</table>
6.2 Survey ......................................................... 46
6.3 Set Up ......................................................... 47
6.4 Data Collection ............................................... 48
6.5 Data Analysis .................................................. 49
6.6 Results ......................................................... 49
   6.6.1 Discussion ............................................... 50

7 Conclusion ..................................................... 52
   7.1 Recommendations ......................................... 53

Appendices ....................................................... 64
Abstract

In video games, Dynamic Difficulty Adjustment (DDA) typically refers to the technique of adapting the video game difficulty to match the skill level of the player. A common practice is to use predictive algorithms to preemptively set up the difficulty of the game so that the player notices the adjustment minimally, and the algorithm tailors the challenge of the game to match the player’s performance.

This paper aims to provide a different approach on how DDA is conducted. Combining methods of player modeling, data-driven analysis and clustering algorithms to provide a better insight of the interaction between the player and the game. However, the paper not only proposes a new methodology for DDA solely based on player performance but it also measures the engagement of the player once the adjustment has been applied by using a 7-point likert scale survey after the adjustment has taken place.
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Image of In-game Pommerman</td>
<td>22</td>
</tr>
<tr>
<td>2</td>
<td>This figure shows the process we followed to conduct the first study</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>Map 1-1</td>
<td>26</td>
</tr>
<tr>
<td>4</td>
<td>Map 1-4</td>
<td>26</td>
</tr>
<tr>
<td>5</td>
<td>Map 2-1</td>
<td>26</td>
</tr>
<tr>
<td>6</td>
<td>Easy Maps: These were the layouts of the maps we categorized as easy,</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>replicated from the original sections of Bomberman 2</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Map 3-3</td>
<td>27</td>
</tr>
<tr>
<td>8</td>
<td>Map 3-7</td>
<td>27</td>
</tr>
<tr>
<td>9</td>
<td>Map 4-1</td>
<td>27</td>
</tr>
<tr>
<td>10</td>
<td>These were the layouts of the maps we categorized as medium, replicated</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>from the original sections of Bomberman 2</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Map 5-1</td>
<td>27</td>
</tr>
<tr>
<td>12</td>
<td>Map 5-4</td>
<td>27</td>
</tr>
<tr>
<td>13</td>
<td>Map 5-6</td>
<td>27</td>
</tr>
<tr>
<td>14</td>
<td>These were the layouts of the maps we categorized as hard, replicated</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>from the original sections of Bomberman 2</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Bar plot showing the distribution of the answer throughout the pre-game</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>survey for all 35 participants</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Correlation and distribution analysis of the gameplay data, showing</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>only the variables that were later selected to build the classification</td>
<td></td>
</tr>
<tr>
<td></td>
<td>model</td>
<td></td>
</tr>
</tbody>
</table>
17 PCA gameplay screeplot; This visualization allows to get a better understanding on how the variables explained the variance present in the data.

18 PCA mixed data Screeplot. The visualization shows a more evenly distribution of the explained variance across the different dimensions.

19 Selected Hierarchical clustering. The visualization shows how participants were placed across the different clusters based on their data. Clusters 1, 2, and 3 refers to Low, Average, and High Performers respectively.

20 Selected Hierarchical clustering Dendrogram. The blue, red and green clusters refers to Low, Average, and High Performers respectively.

21 Correlations on Mixed Data pt.1

22 Correlations on Mixed Data pt.2

23 This figure shows the process we followed to conduct the second study.

24 Screenshot of Pre-Game survey used for the First and Second Study.

25 Screenshot of Post-Game survey used for the Second Study.

26 Increased Range

27 Bomb-Kick

28 Increased Bomb count

29 These were the visual representations for the three different power-ups found in game. From left to right: Increased Range, Bomb-Kick, Increased Bomb count.

30 Survey Data K-Medoids Clustering Plots

31 Survey Data Hierarchical Clustering Plots

32 Gameplay Data K-Medoids Clustering Plots

33 Gameplay Data Hierarchical Clustering Plots
## List of Tables

1. This table shows the set of questions we use to measure a participant’s perceived competence; questions are adapted from the Intrinsic Motivation Inventory [51].  
2. Summary statistics results of the pre-game survey taken by all 35 participants.  
3. Gameplay PCA results. As it can be seen these six different components accounted for much of the variance within the data.  
4. Mixed data PCA results. These results shows the variance explained by the principal components is more evenly distributed in this new data set than in the previous.  
5. Cluster Evaluation Results.  
6. Selected variables results. The table shows the results the selected variables’ means across the three different clusters.  
7. This table shows the set of questions we use to measure a participant’s interest and enjoyment; questions are adapted from the Intrinsic Motivation Inventory [51].  
10. Gameplay Variables.
1 Introduction

Video games have used a wide variety of techniques for controlling the difficulty of the content presented to the player. One common approach is to give the player the option of choosing the difficulty setting. Here the players are presented with a range of different difficulty settings for them to choose, and it is up to the designers what each difficulty setting implies. Some \cite{7,13} have included difficulty curves in their video games where the difficulty increases as the game progresses and the player learns the mechanics of the game. Others have included manual settings where the player can select certain adjustments within the video game to their perceived skill level. However, these methods can be insufficient as they might not adapt correctly to the player’s skill level and needs, and it can make certain aspects of the video games less engaging. The main issue being that the player is constrained to what the developer of the video game has set up its difficulty bar and where. Hence, the developer may not be able to invest the money nor the time to try to adapt its difficulty for the whole market of its game leaving some players that do not feel the difficulty suits their needs, alienated or bored after a while. That is where Dynamic Difficulty Adjustment or DDA comes into play. What DDA aims is to adjust the difficulty based on the player needs and skill so that the game is constantly adapting to the player feedback that it generates by playing the game, and adjusting its difficulty level accordingly. Some of the research done on DDA have included data-driven analysis with techniques that have included Machine Learning \cite{47,46,73,58} to adapt the difficulty to the player skill and needs based on the data collected within the game. Keeping this in mind, we propose a new approach based on data-driven analysis and player modeling. Following this new approach we try to answer two questions that we believe they need to be taken into account.
In short, these two questions are the ones we are trying to answer:

1. *Can we predict player skill using gameplay features and self-reporting data?*

2. *Does skill prediction model for estimating player skill apply consistently enough to other players? does the model generalize?*

To answer these two questions we divided the work into two separate studies. To answer the first question we used player modeling, data analysis and clustering to see if we can assess the skill level of the player. The main purpose of the first study is through player modeling, data analysis and adjusting the level features, to see if we can assess the skill level of the player.

To answer the second question we focused on applying that acquired knowledge and information to a group of participants that have never taken part in the first study and give more insight into how that player model and the data-driven model works with new users and gameplay traces or how can those models generalize outside of the training data that the study is built around.

What this study aims to provide is a different approach for how DDA is conducted, combining the methods of player modeling, data-driven analysis and unsupervised learning to provide better insight of that interaction between player and game. The main difference between the previous studies, such as the one conducted by Teats et al. [34], and the one we are trying to conduct is that we are combining these two methods to provide an extra layer of information about the player and how its perceived skill plays a role in its interaction with the game. This extra layer of information comes in the shape of a survey that tries to identify the player self perceived skill level. Player modeling aims to provide some prior knowledge on the perceived skill level on the player while the data driven modeling aims to cross reference that data provided from
the player’s perceived skill and analyzed which features are important to adjust the
difficulty accordingly.

In the first section of our study we asked our participants to complete a short pre-game survey with the intention of recording their self perceived skill level about playing video games. After completing the survey they were asked to play three different levels from our selected video game, Pommerman [54]. The first study aimed to create an unsupervised learning model, by means of clustering to classify the player into three different skill levels (Low, Average and High Performers). After the evaluation of these different clustering methods, we did select the better fit given our data. In order to differentiate which cluster belonged to which category (Low, Average, High) we performed correlation analysis and difference of means to obtain a greater insight and possibly see meaningful differences that would help us delineate the different clusters. After having differentiated the different clusters we would use the new added variables and create a KNN [19] model to validate its accuracy. The second study on the other hand, will make participants go through the same process as the initial study but with the distinction that after played the three different levels and a brief pause they will be introduced to a level that its based on its perceived skill levels and its gameplay data. In order to do this we will be using the clustering information from the first study and use KNN [19] as a classification algorithm to place the new participant into one of the three different clusters (Low, Average, High Performer). After the participants have finished the whole play session they are required to complete a post-game survey that asks about their enjoyment/interest. Our goals are to see whether there is a difference in enjoyment/interest for the people who had their levels adjusted or not.

The results for the first study showed that certain clustering algorithms and distances clearly work better than others, at least when taking into the consideration
the nature of the data we had. Validating our cluster with KNN [19] proved to be fairly efficient, and at 88% as their highest of the accuracies, the experiment can be called a success. However, results were not as promising with the second study, which might be caused due to the small number of participants. The post-survey results and its analysis proved that there was not significant difference in the results between the people who had their levels adjusted and those who did not (control Group). However, there needs to be kept in mind that although not having results that are statistically significant does not mean these must be overlooked or discarded.

To allow for these questions and studies to be completed we structured our work by doing a review, in Chapters 1 and 2, of previous work done in DDA as well as player modeling. We then proceed to explain the domain, in Chapter 3, were such studies have taken place. After setting up the basics to understand were we stand, we explain our methodology to follow, the set up of the studies and results in Chapters 4 and 5 respectively. As a closing note, Chapter 7 concludes what the studies have accomplished and possible recommendations for future work.
2 Dynamic Difficulty Adjustment

While there is not a clear cut definition of what Dynamic Difficulty Adjustment (DDA), there is indeed a general consensus among researchers about what it tries to do and achieve. In general, DDA is the procedure for automatically balancing the difficulty of a game to adjust it in a way that is optimal to the player’s enjoyment by providing the player with a challenge that is both fun [49] and enjoyable [31]. However, this also includes approaches where the ultimate goal is to make games as engaging as possible [31, 28, 16].

Much research has been done regarding DDA and while newer advancements in machine learning techniques and algorithms allow for the development of new ways for adjusting the difficulty dynamically [39, 73, 34, 27, 58, 70]. The underlying reasons why DDA should be applied still remain an object of much research [4, 58, 39, 2].

One of the most fundamental reasons for DDA can be found in the research conducted by Sha et al. [58]. In their research they study the possibility of creating an enemy AI that produces an appropriate challenge to the player. This reason to use DDA is quite common [10, 34, 6], and it relates back to the study done by Hunicke where the aim is to create a challenge appropriate to the player’s skill level. Inevitably, creating an appropriate challenge for the player also brings another set of reasons for the purpose of DDA. These are maximizing player retention [73] and Player satisfaction [71]. Another reason for DDA can be found in the field of serious games [39, 8]. In this case the application of DDA in serious games contribute to the learning experience, making these games more interesting, enjoyable, and fun to play [37, 48].

Logic dictates that a challenge should be appropriate or match the skill level and performance of the person who is participating in it [35]. Give the participant
A challenge too hard or too easy and eventually it will find it boring 39. Avery and Michalewicz 8 argue that it is the very essence of human adaptation what makes creating challenges in video games a complex dilemma. However, the issue does not stop here as the game should also meet the criteria set by the designer when trying to adapt to the player skill level. Yin et al. 73 tackle this issue by adapting the game scenario difficulties while keeping in mind the conditions set by its designers. Ultimately this has led DDA to focus not solely on the challenge presented to the player but also on its experience and enjoyment 4,31 while also making these adjustment as unnoticeable as possible to the player 31.

A myriad of work has been done regarding dynamic difficulty adjustment. Not only how to but what to adjust has been taken into the equation 74,28,59,16. Done in multitude of environments, such as online games 73, RPG 74, military games 8, Scrabble 28, Endless Runner 68, Tower Defence 65, Checkers and Chinese Chess 33 and many more 70,60. This has lead to DDA being an ever expanding field where a multitude of techniques and approaches have been studied, and new ways to implement those are continuously researched and mixed with new techniques, such as Procedural Content Generation 34,38,68.

Irrespective of the domain where DDA has been applied, adjustments are made using different modalities. Some studies used the benefits of well-known technologies such as Artificial Neural Networks (ANN) to perform their adjustment 73,58. In their studies both lay the groundwork for the use of ANNs on DDA. Yin et al.’s. work builds a neural network based on offline gameplay data and gameplay settings to build a function that later can estimate the difficulty of the scenario for a certain player. Using that data as their foundation to build a function that approximates the difficulty of each scenario when playing online the ANN also gathers the data from the current play session. This approach allows them to change their scenarios and events
in order to match the desired difficulty. On the other hand, Sha et al. research while focusing on the use of ANNs it mixes its power with the capabilities of the Monte-Carlo Tree Search (MTCS) by creating an ANN that approaches the functionality and performance of a MTCS. An advantage of this approach is that where the MTCS fails because being a time and resource heavy technique the Neural Network helps to minimize these constraints. This approach is very similar to the study done by Hao et al. [27] where they also use ANNs made specifically to approach MTCS while managing to keep the performance similar to the former. As a result of the approaches mentioned before, the performance and results tend to be positive. But we should also consider that a main drawback from this approaches is not only the amount of resources needed but the complexity of how those algorithms are constructed and applied. ANNs being a “black box” method might also prove a major drawback when identifying the main qualities on why they work.

However, not every approach for DDA requires complex algorithms or otherwise resource and time consuming techniques. Sutoyo et al. approach [65] for DDA comprises of a series of formulas to adjust the different parameters in game. The advantage of this approach is its simplicity and the ease with which this method can be applied in their scenario without requiring more complex approaches. Other approaches that rely on less complex methods is the study done by Harrison and Roberts [28]. Their approach relies on general data analytics to adjust the difficulty dynamically. This system proves not only useful for adapting the game to the player but also as their paper states, to improve player retention.

As mentioned before one of the objectives of DDA is to tailor the challenges to the player while disrupting the experience and the enjoyment of the player to a minimum. The underlying issue, as pointed out by Hunicke [31], lies in the definition of “fun”. Hunicke states that in order to minimize this impact, there is need to be understanding
on how the game is designed and why it is fun. She mentions that the MDA \cite{32} can provide a framework to help understand this process. This issue does not help to tackle DDA from the standpoint of player experience or enjoyment since both terms, similar to “fun”, can be considered purely subjective and what might work for one player it might not work to another. Some of the research conducted by Baldwin et al. \cite{10, 11, 12} try to shed more light on how DDA and player experience (PX) might be correlated. Baldwin et al. initial research \cite{12} focuses on creating a framework to which DDA can be built upon in multiplayer videogames. The framework is broken down into seven different components to which the framework is built. These components are: \textit{Determination, Automation, Recipient, Skill Dependency, User Action, Duration, Visibility}. The advantage of these components, as Baldwin et al. states is that they can be generalized in every scenario. After having built their framework their other research \cite{12, 11} focuses on visualizing the effect that those components have on the PX. This novel approach allows for better understanding of how DDA affects something as subjective as PX might be.

Another approach that tackles the issue of player experience and challenge adjustment is the work done by \cite{22}. In their research they explore the effects of adapting difficulty and how it affects the player’s performance and experience. The approach proposed by them focuses on allowing the player to have longer playtime if they were inexperienced. The results showed that allowing for more time not only increased player’s performance but also the PX as they would feel more immersed into the game. In contrast to Denisova & Cairns research where the manipulation of the playtime session is used to study how the adjustment affected the player, Alexander et al. study \cite{4} studied how changing the mechanics of the game based on the player’s perceived performance and the difficulty chosen affected the enjoyment of the player. The findings of their study led them to the conclusion, that DDA needs to adjust
to the player’s desires and not to the designer’s, something that contrasts with some other approaches on the field.
3 Player Modeling

Video game players have been an object of research for as long as video games have existed. Designers, artists, and programmers alike have been seeking to understand how players interact with their game and how to build a system where the content is tailored to different player personalities, needs, goals, styles and even behaviour [14, 20, 40, 9]. Player modeling is one of those attempts to understand what drives the player to make the choices and take the actions they do in games [21, 17]. Although player modeling has been a big object of research for a while now, the term player modeling remains by itself in constant change. Yannakakis et al. [72] Describe player modeling as “the study of computational models of players in games”. Adam et al. [63] on the other hand, describes it as range of techniques use to understand the relationship between the player and the game. These techniques can range from machine learning models [25, 20, 41], data-driven analysis [66] to surveys [21] to better understand that interaction between player and game.

Video game players are, to some extent, consumers of a product made to fulfill their needs and goals in the entertainment domain, although this does not apply for every case. As it happens with other forms of entertainment these require a substantial amount of research and study about their target audience, as the final goal for video game companies is to maximize their revenue and game sales. The issue, as it is with some other forms of entertainment and art, is that there is no clear guidelines to maximize player enjoyment, and improve player’s experience based on their profile and style. Player modeling is one of those practices that tries to close this gap.

From the academic standpoint player modeling is a technique that can better help understand the behavior of players in video games [57, 63, 9, 30]. However, player
modeling is not limited to understand player’s behavior but use data to predict the
player behavior and adjust aspects of the video game according to the model [17].
Additionally some studies on player modeling such as the one conducted by [21] tries
to understand what is “fun” based on the player opinions and preferences which can
be linked to the study where Hunicke [31] points out that to better understand video
games, the definition of what is “fun” must be understood within the context of the
video games.

From a game company standpoint player modeling is a tool that is often used by
publishers and developers alike to improve the player experience [66, 17], build the
system around the player preferences [21], and understand their behaviour in game.
It is only reasonable that if game companies are trying to understand the individual
interaction between players and games [72], player modeling is an effective technique
that can bring benefits.

Generally player modeling can be divided in two categories, Data-Driven [50, 5, 29]
and Theory-Driven [66, 43, 57]. Although the line between these two approaches
might blur sometimes, the difference in these two categories occurs in the way the
player modeling is conducted. Theory-Driven player modeling involves the creation
of a framework or a theory to build the model around the player. Typically theory
driven approaches involve the use of surveys [21] or the use of already established
methods, such as the Five Factor Model [45, 40] to conduct their study. On the
other hand however, data-driven directly taps into in-game gameplay to create the
player model based on the data footprint left by the player. While this approach can
yield more promising results, and some of them in real time [57] since the model is
created around data generated by the player, it also has an extra level of complexity
due to techniques used such as clustering [5], Hidden Markov Models [15] and Tensor
Factorization [75].
Theory-driven player modeling certainly has its benefits, the study done by Albuquerque and Fialho [21] where they propose a framework with the inclusion of a survey to find the players preferences and their impact of what is fun, is an appropriate example. An approach like theirs has inherently benefits, not only in the field of academia and player modeling but for game companies and as a game design tool. In terms of academia, this type of approaches provide a more in depth look and understanding to what players value when playing video games and how that affects the fun factor of the game itself.
4 Domain

4.1 Pommerman

The software used to conduct the studies has been Pommerman [54]. Pommerman is based on the original console game Bomberman [61]. In the original game Bomberman your objective is to clear levels by destroying the enemy agents with the bombs your character drops. The choice to select this software was both for its reproducible advantages, it being an open source software and a top down perspective video game. In addition to that, a big reason to use this software of choice was the fact that while DDA has been researched in a myriad of environments, top-down video games have still remained a relatively small niche on the field [70, 28]. The program is written in Python and while it allowed for all the customization needed for the study, however, one of the limitations in our study given the time constraints was that the game could not be played remotely as we did not have time for the implementation of said feature. This program while built for Machine Learning and Multi-Agent learning purposes, allowed for a wide array of variation and tweaks that made this work possible. Originally Pommerman has its code set up for Multi-Agent learning hence, some tweaking was needed in order to collect the data necessary for the studies. Thanks to Pommerman’s implementation we were also able to personalize our own game boards as we saw fit. Although the software had a built in function to collect data from gameplay for the purpose of the study new lines of code had to be implemented to allow for extra information to be collected. We modified the player and enemy agents lives to allow for longer session of play, but also limited the number of steps in game to make it short so it would not last too long draining the players attention [24]. Other modifications included slowing down the enemy agents speed.
to allow for better synchronization between enemy agents and players. An important modification that took place and was if not the most important was the possibility to play different maps, one after another and the ability to randomize the selection of maps each time the program was run. This modifications allowed for each play session to have randomized maps and consistency while at the same time removing any possible human bias in the process.

![Pommerman](image)

**Figure 1: Image of In-game Pommerman**

Unfortunately due to Pommerman being a open-source software and being built for a different purpose, there were a few drawbacks to the program. While we slowed down the enemy agents’ speed, the way Pommerman was conceived and written it created a very small but noticeable lag between the players input and the action occurring on screen. Despite this issue the game could be played normally and the lag did not prove an impeding factor to efficiently play the game.
5 First Study

In this first study we try to answer the first question of our paper: Can we predict player skill using game play features and self-reporting data? To this end we perform an analysis of the different types of data obtained from both the game play and the survey, then we proceed to build different clusters as analyze them to overview which one of them is the best fit given our purpose. Once the cluster has been accomplish we proceed to create a classification model in order to predict the player skill level.

5.1 Goals

The first study comprises one of the main goals of this work. Due to the particular characteristics of the first question it is key for the success of the study as a whole and to allow the second study to be effectively conducted. While the question can not be answered in its entirety by the first study it sets the foundation for how the rest of the study will be conducted. However, collecting this initial data enable us to perform critical data driven analysis and clustering techniques to be able to identify the key characteristics that might impact the relationship.

Incidentally, as not many studies have done DDA research on top-down video games [70, 28] a goal of the first study is to have player information in a domain that has not been researched as much as some others [34, 60, 64]. In addition, collecting gameplay data can give us a better insight into how a level layout plays a role in DDA and how players fare based on each map layout. This would not only be helpful for DDA purposes but it could further be applied to some other techniques such as PCG and player modeling. Moreover after some literature review we found that surveys in player modeling are, to a certain extent, widely used [21, 36]. At the same time,
the extent of these surveys is generally limited to do some average data analysis and used them as a part of a theory driven player model. Our purpose for including such survey is to include a player modeling feature into a DDA model and if the results are significant enough include these results in the clustering model for DDA. By collecting self reported and gameplay data we expected to have enough information about the participants’ behaviour and their gameplay patterns to provide insight on how this self reporting data might be related to how the player interacts with the game.

5.2 Methodology

The approach and methodology for the first study comprises of several stages, these being, data collection, data analysis, clustering, evaluation where the participant was asked to complete different tasks. The tasks required for the participant were to complete an initial pre-game survey, and a play session consisting of three different maps. This allowed us to gather distinct levels of information about the participant self-perceived skill level and their gameplay data.

5.2.1 Survey

The player modeling section consists of a survey adapted from the Intrinsic Motivation Inventory (IMI) [51]. The survey, made in a 7-point likert scale and consisting of six different questions, aims to extract an understanding of each player’s self-perceived skill level. The survey takes places immediately before the participant is asked to play the game. The idea for using this survey in our study was to help us understand if the participants’ self-perceived skill could be used to predict their actual in-game skill level. While not relevant to the player modeling section the survey also included a last question where the participant was asked to fill out a Player ID which they got right after playing the game. The addition of this last field on the
survey was so consistency and tracking of each player could be possible since no other identifier was used throughout the study.

The survey questions can be seen in Table 1.

1) I think I will do pretty well compared to other students
2) After playing a few times I think I can feel very competent
3) I think I am pretty good at playing video games
4) I am satisfied with my performance at playing video games
5) I am pretty skilled at playing video games
6) I can’t do very well at playing video games (R)

Table 1: This table shows the set of questions we use to measure a participant’s perceived competence; questions are adapted from the Intrinsic Motivation Inventory [51].

5.2.2 Set Up

The set up was segmented into two different tasks that every participant had to go through. Figure 2 shows the process of which every participant had to go through during the length of each session.

Figure 2: This figure shows the process we followed to conduct the first study

The first part as mentioned before, was the IMI survey which every participant had
to fill out. The participant played three different maps that were randomly assigned to them. These maps were categorized as Easy, Medium, Hard, taken originally from Bomberman 2 [62]. The underlying reason to replicate those maps from the original Bomberman 2 was under the assumption that early game maps were easier while late game maps were harder. The reason for this was that not being professional level designers this would alleviate some of the issues we would have by introduce by designing our own levels. This would not only remove bias on our side but also would give us more validation when using them because their origin from the actual game.

Each difficulty setting had a pool of 3 different maps as seen on Figures 6, 10 and 14 for which the participant had to play one of each difficulty setting. We presented those levels in order of difficulty, that being Easy, Medium, Hard. In addition every map contained three different power-ups Kick bombs, Increased Blast Radius, Additional Bombs, which anyone in the map, including agents, could pick them up. The enemy agents count increased +1 for every level of difficulty, totalling at the highest difficulty level with three agents.

Each difficulty setting had different layouts and characteristics. These were as follows:

![Figure 3: Map 1-1](image1.png)  ![Figure 4: Map 1-4](image2.png)  ![Figure 5: Map 2-1](image3.png)

Figure 6: Easy Maps: These were the layouts of the maps we categorized as easy, replicated from the original sections of Bomberman 2
Figure 10: These were the layouts of the maps we categorized as medium, replicated from the original sections of Bomberman 2

Figure 11: Map 5-1
Figure 12: Map 5-4
Figure 13: Map 5-6

Figure 14: These were the layouts of the maps we categorized as hard, replicated from the original sections of Bomberman 2

- Easy maps: Replicated from the early sections of the game, these maps were taken from levels 1-1, 1-4, 2-1 in game. These maps comprised of only one enemy agent and one power-up of each type.

- Medium maps: The medium difficulty maps were replicated from the mid section of the game. These comprised the sections 3-3, 3-7, 4-1 in game. In contrast to the easy layout maps these maps had two enemy agents as well as two power-ups of each category.
• Hard Maps: The hard difficulty maps were replicated from the late game section of the game. These sections were 5-1, 5-4, 5-6 in the actual Bomberman 2 game. In contrast to the other two previous layouts these maps had three enemy agents and three power ups of each category.

As mentioned before, we modified the original Pommerman code to reach several goals within the study. One of these changes made to Pommerman was a duration limit of each game. The duration of each map was limited to 1 minute 45 seconds, totalling roughly 5.30 minutes for each play session. This value represented 500 iterations of Pommerman’s code for each map. This limit was chosen intentionally to shorten each play session to make sure participants had enough time to familiarize with the basics and play each level comfortably but not enough to fully develop expertise of the game.

5.2.3 Data Collection

The process of collecting participants was done by word of mouth. All participants that took part in the first study were Northeastern students and or faculty members. No confidential data nor demographic data was recorded and the only requirement was for the participant to be 18 years old or above. Prior experience with either the original Bomberman or Pommerman was not required and, incidentally, recruiting participants that did not have prior experience with either would result in a more unbiased data collection. The rules of the experiment as well as the rules of the game were explained beforehand to every participant although not more information was provided. The reason being that letting the participant know what the experiment was about might have tainted the survey results and not being an accurate representation of their perceived skill level. Due to time constraints in the study, the data collection was time boxed to a 3 week period and only 35 participants were used for
the first part of the study.

The retrieval of the collected data was split into different parts mainly due to the data coming from different sources. For the IMI survey the method of collection and creation used was Google Polls. This included a timestamp, marking when the participant completed the survey, the question asked, the answer to it, and the player ID of each participant. The survey data did not present major issues when collecting it and it allowed for a very easy transformation into a format workable in R.

Some of the variables, (Table 10 in the Appendix), captured by Pommerman were, the type of agents used, if the agents and/or the player was alive, the number of power-ups picked up, the number of steps, the location of every agent and player, the state of the board and the board itself, although not all data collected by Pommerman was later used in the analysis.

The last part of the data cleaning and post-processing made sure that all three files where now combined into one with every row being one participant and its gameplay traces across the whole play session. Although the brunt of the data collection was done, in order to make for easier transition into the different modalities of data analysis and cluster modeling, two different versions of this data file was created. One, unaltered with all the recorded data properly formatted and its original values, another with the data scaled normalized and all its categorical variables converted to continuous by the method of one-hot encoding ready to be used with clustering algorithms and more methods forms of data analysis.

The variables collected from the gameplay can be seen in the Table 10 found in the Appendix.
5.2.4 Data Analysis

In order to perform the data analysis we collected previously, the data was separated into three different categories. Survey Data, Gameplay Data, Mixed Data. The survey and gameplay data comprised only of the data that was either collected from the survey or the gameplay traces from each participant. On the other hand the mixed data was the combination of both the survey data and the gameplay traces from each participant.

Different methodologies were followed to analyze each of the data files that were collected. The first to work with was the survey data. For this data collected we performed an analysis on its summary statistics as well as some exploratory graphs to help us get a sense of how the data was distributed and what the main trend among participants was. However, the data from the survey being categorical, there was not much else that could be done to analyze it in a way that would give us meaningful results.

In contrast, both the gameplay and mixed data allowed for a more in depth analysis of its data and greater insight on its characteristics. Same as the survey data, we reviewed the summary statistics for both the gameplay and mixed data. Due to the size of this data compared to the previously mentioned survey, we did make use of the package Psych \cite{psych} to help us visualize its distribution and characteristics. Adding to this, PCA \cite{pca} from the package PCAmixdata \cite{pcamixdata} was performed to the mixed data file. The purpose of using the package PCAmix was due to the fact the data contained not only continuous variables but also categorical.

5.2.5 Clustering

An important part of this study was the clustering. Because of this the methodology followed for this section relied on multiple iterations, the use of different al-
gorithms and distance metrics to make sure we would obtain the best clustering possible. Each clustering method was independently done on every data set at our disposal (Survey, Game play and Mixed data) to account for possible patterns, trends or disparities among the different methods. Since the goal of this study was to test whether we could predict player skill by using self-reporting data and gameplay traces we worked under the assumption that there would be three different skill levels, which were based on the maps we provided to the participants. These skill levels were Low Performers, Average Performers, and High Performers. This assumption allowed us set a ground truth for the number of clusters and would further help us define and evaluate these. Based on this assumption also we set up the number of clusters to three, irrespective of what the results were.

The different algorithms and metrics used where as follows:

- **K-Medoids / PAM [44]** :

  We used the K-medoids or Partitioning Around Medoids (PAM) for multiple reasons. First, K-medoids is generally a more robust version of k-means in the presence of outliers and noise, something that we could not afford to remove since our amount of data points was sparse. Additionally, the K-medoids algorithm is parametric, that means it needs some prior knowledge of the number of cluster before performing any operation. This was particularly useful since the intention was to create three different clusters, one for each difficulty setting and then ascribe the participants to any of those based on their data. To perform the K-medoids clustering we used the Cluster [44] package from Rstudio, which provided us with the function Pam() in which we could input either a dissimilarity matrix or the raw data

- **Hierarchical Agglomerative Clustering by Complete Linkage [52]** : 

  Its simple implementation and the ability to output dendrograms without having to
know apriori the number of clusters in our data made this algorithm suitable for this study. Unfortunately one of the disadvantages of this algorithm is its sensitivity to outliers and noise. We also assumed that the data was randomly ordered, which in fact that was the case, so there was not any need to randomize the data before putting through this algorithm. The function used for this was \textit{Hclust()} from the package \textit{stats}. 

For both algorithms used for the clustering we used the following three distance metrics and created the dissimilarity matrix with the function from the package \textit{Cluster} and using its function \textit{Daisy()}.

- **Euclidean Distance:**

The formula for the Euclidean Distance is: \[ \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \]

Where \( p \) and \( q \) represent two different points in Euclidean Space and \( n \) the number of dimensions or neighbors.

A relatively simple distance to calculate, it works by calculating the distance between two points in a euclidean space. We decided to use this distance measure due its simplicity yet its robustness. Although Euclidean distance is known to not work well high either high dimensional data or categorical variables we, however, decided it could give us meaningful results. The function Daisy() calculates the Euclidean Distance by the root sum-of-squares of differences.

- **Manhattan Distance:**

The formula for Manhattan Distance is: \[ \sum_{i=1}^{n} | x_i - y_i | \]

Where \( x \) and \( y \) represents the two different points in a vector space.

The reason to use this distance was its ability to deal better with outliers, noise and in general better results than Euclidean Distance when using High dimensional
data sets [3]. The Manhattan distance in the Daisy() function is calculated by the sum of the absolute differences.

- **Gower’s Distance:**

The formula for Gower’s Distance is:

\[ \sum_{k=1}^{v} \frac{S_{ijk} \times W_{ijk}} {\sum_{k=1}^{v} \delta_{ijk} \times W_{ijk}} \]

Where \( k \) is the \( c^{th} \) column. \( v \) the number of columns in the dataset, \( w_k \) the weight of said \( c^{th} \) column. \( i \) and \( j \) are the different points to be compared. In the formula \( \delta \) accounts for possible missing data.

First introduced by Gower [26], it is a distance that is known to work relatively well with mixed data, and since our data is comprised of both categorical and continuous variables it was an appropriate distance to use.

### 5.2.6 Evaluation Methods

Choosing a distance metric was not an easy task, and the factors involved were multitude and may vary depending on the data at hand, hence why we decided to try these three different distances and once we had the clusters we could evaluate their results and decide which one suited best our needs. In order to evaluate our clusters we decided to use three different metrics (**Principal Component Analysis of the clusters** [1], **Average Silhouette Width** [56], and **Dunn’s Index** [23]) of internal evaluation for clustering.

**Principal Component Analysis** [1]

One of our data analysis techniques was PCA. We decided to use PCA because we wanted to have a prior understanding of what the important features were and how much those features explained the variance in the data. This would not only give us more insight about our data but would also help later in the study to select the appropriate variables to be selected when filtering. In addition we also used PCA
to evaluate the clusters and see which ones explained most of the variance within the data.

**Average Silhouette**

For clustering, the average silhouette measures how well a certain data point has been clustered. In other words, it measures the average distance between clusters from one observation to another. We decided to use this method because it would allow us to see how well shaped the clusters were. While we understood there could be some overlapping in the clusters we wanted to optimize those that had their observations as differentiated as possible.

**Dunn's Index**

Coined after J.C. Dunn, the Dunn index (DI) identifies those clusters that are well separated apart from each other but compact enough within the members of each cluster. The higher the DI value the better as it denotes that clusters have good separation and compactness. We used this internal evaluation metric because we wanted to value the clusters that were more separated and compact from each other. However, as previously mentioned, because we knew the clusters could overlap this could lead us into better suited clusters.

**K-nearest neighbors algorithm (KNN)**

We used KNN during the last part of this study as a validation method for our cluster results. This would test the accuracy of the final clustering method we selected and inspect whether the model was sound or not. Approaching this as a classification problem we run KNN from the package `class` on the final model to test the accuracy of the results from the clustering. We set up the training and testing data such as: Training = m - Row_n
Testing = n
Where Row_n represents the number of the row to be classified and m the total
number of rows in the data set. Because of this the value for K was set up as K=1. The package used was the class package, both for its quick running time and its simplicity on its implementation.

5.3 Results

Because the clustering methodology was applied to the three different subsets of data we divided its results independently.

Survey Data

First, the exploratory analysis on the survey data showed that the distribution of the survey results were far from having a normal or uniform distribution, instead and did not seem to follow a predefined pattern. Figure 15 and Table 2 shows that the mean between questions was not too far off from each other. What is interesting, is too see that for Maximum and Minimum scoring both Question 2, and Question 5 did not attain the whole range of possible answers. While Question 6 was a control question, since we were interested in their actual scoring and not the overall results we did not perform any further transformations on it. Keeping this in mind choosing PAM proved specially useful with this subset of data since the data was not normally distributed.

<table>
<thead>
<tr>
<th>Question</th>
<th>Mean</th>
<th>Median</th>
<th>Max</th>
<th>Min</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>3.8</td>
<td>4</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>Q2</td>
<td>5.075</td>
<td>5</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>Q3</td>
<td>4.114</td>
<td>4</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>Q4</td>
<td>4.8</td>
<td>5</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>Q5</td>
<td>3.886</td>
<td>4</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>Q6</td>
<td>3.457</td>
<td>3</td>
<td>7</td>
<td>1</td>
</tr>
</tbody>
</table>

**Sum Total** 25.11 26.00 34 14

Table 2: Summary statistics results of the pre-game survey taken by all 35 participants
The clustering results for the survey can be seen in Table [5]. It showed some interesting results in terms of its evaluation metrics. With an average of 38.7% of variance explained by the principal components and an Avg. Silhouette of 0.0932, Hierarchical clustering performed better on those two aspects than K-medoids. In terms of the differences between the distances, Euclidean performed better under the Dunn’s Index, however, we believed this is due to the way the Dunn’s Index is calculated and while the metric showed better results than the other two distances it could be misleading focusing only on one metric alone. As a result we did concluded that Gower’s and Manhattan distance where a better fit for the clustering of the survey data.

Gameplay Data

The gameplay data showed similar results to those of the survey, despite the fact that this time there was a mix of both categorical and continuous variables. First, as mentioned before, we performed some initial exploratory analysis that allowed us to get a better sense on our data, we were working with.

However, since for the final clustering we only used a handful of selected features the distribution and correlation can be appreciated in Figure [16]. In this figure it can
be seen that throughout all of the selected features only two of them had co-linearity. These being lives_e which recorded the number of lives each participant had at the end of the game on the easy maps, and is_alive_e that recorded whether the participant was alive at the end of the easy maps session.

Figure 16: Correlation and distribution analysis of the gameplay data, showing only the variables that were later selected to build the classification model

The other analysis performed on the gameplay data in distinction to the survey data was PCA analysis, as previously mentioned we used the package PCAmix [15] for this task because it allowed us to perform principal component analysis on mixed data. The results showed, seen in Table 3 and Figure 17, that out of 30 variables most of the variance could be explained by six components. By reducing the dimensionality of the data we hoped to better understand how and what components explained the variance within the data, however, since our interest was into ascribing players into a pre-defined number of clusters (3) we deemed necessary to use all possible variables
in order to have all results unaltered and increased readability.

<table>
<thead>
<tr>
<th>PCA Analysis</th>
<th>Dim 1</th>
<th>Dim 2</th>
<th>Dim 3</th>
<th>Dim 4</th>
<th>Dim 5</th>
<th>Dim 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variance</td>
<td>6.39</td>
<td>4.77</td>
<td>3.71</td>
<td>3.08</td>
<td>2.79</td>
<td>2.40</td>
</tr>
<tr>
<td>% Variance</td>
<td>15.99</td>
<td>11.93</td>
<td>9.27</td>
<td>7.72</td>
<td>6.98</td>
<td>6.02</td>
</tr>
<tr>
<td>Cumulative % of var.</td>
<td>15.99</td>
<td>27.93</td>
<td>37.21</td>
<td>44.93</td>
<td>51.92</td>
<td>57.94</td>
</tr>
</tbody>
</table>

Table 3: Gameplay PCA results. As it can be seen these six different components accounted for much of the variance within the data.

Figure 17: PCA gameplay screeplot; This visualization allows to get a better understanding on how the variables explained the variance present in the data.

In a similar way to the survey data we performed all different clustering algorithms onto the gameplay data in order to distinguish whether there was a relationship between these to subsets of data (Survey and Gameplay) or not. The results had certain resemblance to those obtained with the previous subset of data. Hierarchical clustering on average performed better than K-medoids. For the percentage of variance explained within the clusters, K-medoids only provided an average of 27% variance
explained, compared in contrast to the hierarchical algorithm where it averaged out 42.16% of variance explained. In terms of the other metrics (Average silhouette Width and Dunn’s Index) on average both algorithms performed similarly with k-medoids having the best average silhouette width $0.131$ but the hierarchical clustering having the best Dunn’s index with $0.4402$.

Distance evaluation however, showed again that euclidean distance had the best results above all, which further proved that despite its better results over the other two, it might not be an accurate representation of the actual performance of the distance alone. Manhattan and Gower’s distance scored in their individual evaluations quite similar to each other, not surprising since both metrics tend to work relatively well on portions of mixed data.

**Mixed Data**

The mixed data was the main focus of our study. It consisted on the data collected from each play session and the survey from each participant. We then performed a visual inspection of the data. Even as this data sat was the combination of the survey data and the gameplay, the newly added variables combined could reveal new results and gives a further insight in their collinearity and distribution. Additionally PCA was also performed as the new subset of variables from the pre-game survey and gameplay data combined could reveal new findings into the variance and principal components of the data.

The visual analysis revealed that the colinearity between variables was not too high with the exception of the variables from the survey data. As expected those variables present in the survey contained more colinearity due to the set up of the survey. However, as we accounted for possible issues with it some of our methods were not as dependant on that. In contrast the PCA analysis, as seen in Table 4 and Figure 18, revealed a difference from the previous subset of data. This time it
revealed that around 7 variables explained most of the variance present in the data, however, the distribution of the variance explained was more uniform throughout this subset of data.

<table>
<thead>
<tr>
<th>Variance</th>
<th>Dim 1</th>
<th>Dim 2</th>
<th>Dim 3</th>
<th>Dim 4</th>
<th>Dim 5</th>
<th>Dim 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.55</td>
<td>6.05</td>
<td>5.17</td>
<td>4.37</td>
<td>4.16</td>
<td>3.99</td>
<td></td>
</tr>
<tr>
<td>% Variance</td>
<td>11.53</td>
<td>8.17</td>
<td>6.99</td>
<td>5.90</td>
<td>5.62</td>
<td>5.39</td>
</tr>
<tr>
<td>Cumulative % of var.</td>
<td>11.53</td>
<td>19.70</td>
<td>26.70</td>
<td>32.61</td>
<td>38.24</td>
<td>43.63</td>
</tr>
</tbody>
</table>

Table 4: Mixed data PCA results. These results shows the variance explained by the principal components is more evenly distributed in this new data set than in the previous

![PCA mixed data Screeplot](image)

Figure 18: PCA mixed data Screeplot. The visualization shows a more evenly distribution of the explained variance across the different dimensions

5.3.1 Discussion

Based on the evaluation of our clusters, as seen in Table 5 we were able to discern the different characteristics of these. At first, it can be seen that the K-medoids
(PAM) algorithm perform rather poorly in comparison to the Hierarchical. This could be caused by the different approaches that both algorithms have in creating clusters. K-medoids algorithm resembles k-means \([42]\) in certain aspects, it creates partitions of the data and use each observation as the center. On the other hand, hierarchical clustering treatment of data is slightly different. It treats every data points as it own cluster until it identifies the closest to each other. Given the previous analysis, and as PCA showed, the variables in the Mixed Data subset showed that the variance explained in the data was more spread out throughout the different components. These two reasons could be what caused K-medoids to perform more poorly than hierarchical. This lead us to the assumption that using Hierarchical was the better suited for the study. The results between the different clusters helped to make the decision as for which clustering algorithm we would end up using, However, as mentioned before, choosing a distance metric has its own complexity by itself.

As mentioned previously we decided to evaluate every cluster and distance based on three different metrics \(\text{PCA, Avg. Silhouette Width and Dunn’s Index}\). Euclidean distance performed better than the other two distances when it comes to the evaluation by the Dunn’s Index. Since Dunn’s Index tends to look for the clusters that are well separated and compact it is no surprise that Euclidean performed above the other two distances. Besides, the other two distances performed better in both PCA and Avg. Silhouette Width than Euclidean distance. In Fact Manhattan and Gower’s performed so similarly that it was not until the last subset of data (Mixed Data) we deemed fit to use Gower’s instead of Manhattan. They both performed very similarly, in fact having the same results for the Survey Data, although we explained before that might be caused due to the lack of enough observations. On the Mixed Data, Gower’s performed better than Manhattan both at the Dunn’s Index and Avg. Silhouette Width. Despite the fact that they scored the same in PCA we decided to
choose Gower’s distance over Manhattan due to its better performance on the last subset of data and its performance across the other two subsets.

<table>
<thead>
<tr>
<th></th>
<th>K-Medoids</th>
<th>Hierarchical</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Survey</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Euclidean</td>
<td>26.65</td>
<td>31.6</td>
</tr>
<tr>
<td>Manhattan</td>
<td>25.32</td>
<td>42.3</td>
</tr>
<tr>
<td>Gower’s</td>
<td>25.32</td>
<td>42.3</td>
</tr>
<tr>
<td><strong>Gameplay</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Euclidean</td>
<td>28.36</td>
<td>35.50</td>
</tr>
<tr>
<td>Manhattan</td>
<td>26.77</td>
<td>45.60</td>
</tr>
<tr>
<td>Gower’s</td>
<td>26.55</td>
<td>45.40</td>
</tr>
<tr>
<td><strong>Mixed</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Euclidean</td>
<td>20.26</td>
<td>24.40</td>
</tr>
<tr>
<td>Manhattan</td>
<td>21.13</td>
<td>35.30</td>
</tr>
<tr>
<td>Gower’s</td>
<td>21.24</td>
<td>35.30</td>
</tr>
</tbody>
</table>

Table 5: Cluster Evaluation Results

Once we had our clustering method selected, that is, Hierarchical Clustering with Gower’s distance, we still needed to differentiate the three different clusters. In order to do this we performed a correlation analysis to investigate which variables had more impact into the clustering. The set up for this correlation analysis was to filter out all those variables that had less than 0.40 points of correlation. We decided to use 0.4 so we could cast a bigger net of variables since we understood that given the nature and scarcity of our data high correlation could be a privilege. According to the correlation Figures 21, 22 these ten variables accounted for much of the correlation for the clustering. After we reviewed this information we decided to perform a Wilcoxon Test [69] to see whether the differences of means in these variables between clusters were significant enough. The results as seen in Table 6 showed that there were four variables between cluster 1 and 2 whose difference in means was significant. Adding to this there were 9 variables between cluster 1 and 3 whose difference in means was significant enough.
Figure 19: Selected Hierarchical clustering. The visualization shows how participants were placed across the different clusters based on their data. Clusters 1, 2, and 3 refers to Low, Average, and High Performers respectively.

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>lives_e</td>
<td>0.5714</td>
<td>1.875</td>
<td>2.833</td>
</tr>
<tr>
<td>is_alive.1_e</td>
<td>0.761</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>stepstaken_m</td>
<td>92.333</td>
<td>174.375</td>
<td>137</td>
</tr>
<tr>
<td>block_destroyed_m</td>
<td>0.507</td>
<td>0.805</td>
<td>0.787</td>
</tr>
<tr>
<td>is_alive.1_h</td>
<td>0.761</td>
<td>0.375</td>
<td>0</td>
</tr>
<tr>
<td>Q1</td>
<td>3.333</td>
<td>4</td>
<td>5.166</td>
</tr>
<tr>
<td>Q2</td>
<td>4.714</td>
<td>5.125</td>
<td>6.166</td>
</tr>
<tr>
<td>Q3</td>
<td>3.380</td>
<td>5</td>
<td>5.5</td>
</tr>
<tr>
<td>Q5</td>
<td>3.380</td>
<td>4.375</td>
<td>5</td>
</tr>
<tr>
<td>Q6</td>
<td>4</td>
<td>3</td>
<td>2.166</td>
</tr>
</tbody>
</table>

Table 6: Selected variables results. The table shows the results the selected variables’ means across the three different clusters.
With these results we assumed the following order for clustering:

- Cluster 1: Low Performers
- Cluster 2: Average Performers
- Cluster 3: High Performers

Despite this, we still had to evaluate and validate our clustering model and its predictive power. To this end using the Mixed Data, as it was to be our training data set for the second study, we performed KNN. In order to be careful we decided to use both the whole data set and also those variables that more than 0.40 correlation to see whether the model changed a lot. The results were of 88% accuracy with all variables, and 80% with the subset of correlated variables. Not being a huge difference between these two models and our study being oriented at identify those variables that make most of the impact in the participant’s skill level we decided to use the later model with the subset of correlated variables.
Figure 21: Correlations on Mixed Data pt.1

Figure 22: Correlations on Mixed Data pt.2
6 Second Study

In this second study we tried to answer the second question of our paper: *Does skill prediction model for estimating player skill apply consistently enough to other players? does the model generalize?*. To this end we used our classification algorithm (KNN) to classify where the new pool of participants would be placed based on their skill level and gameplay traces by using the clustering model previously built. Once this was done we analyzed the results to see whether there was statistical significance between the control and the experimental group.

6.1 Goals

The goals for the second study were two fold. First our goal is to test whether the model generalized with newly acquired participants or not. Additionally another goal from this study was to understand and see if that adjustments in the game would have any impact into the participant enjoyment or interest.

6.2 Survey

Much like the first study the participants were required to complete the Pre-Game survey. Same as before, participants were required to complete this survey before each play session and submit it right after once they got their player id. However, the change in the second study was a second added survey to be completed right after each play session. The idea behind this survey was to asses whether participants had interest and enjoyed their adjusted levels. While this survey would not provide a full assessment of their player experience it definitely gave us an insight on the model we constructed and its efficacy.

The survey was designed as follows:
1) I enjoyed doing this activity very much
2) This activity was fun to do
3) I thought this was a boring activity (R)
4) This activity did not hold my attention at all. (R)
5) I would describe this activity as very interesting
6) While I was doing this activity, I was thinking about how much I enjoyed it
7) I thought this activity was quite enjoyable

Table 7: This table shows the set of questions we use to measure a participant’s interest and enjoyment; questions are adapted from the Intrinsic Motivation Inventory [51].

6.3 Set Up

![Diagram showing the process of the second study](image)

Figure 23: This figure shows the process we followed to conduct the second study.

In the same note as the first study, the participants had to complete the Pre-Game survey, then play the game from the pool of three different difficulties. Now the change from the first study is that once they played the first three levels, the play session would be interrupted briefly. We would run our model with the data from the first study as our training data and the newly acquired data. After collecting data the current participant we then proceed to run the model we created on the first study and the participant would be presented with a level that matched its performance.
It is important to note, that despite giving the participant a map according to its performance it would not play the same level twice as once they place that level it would be discarded from the pool of maps.

After having played the map suggested by the model, the participant was then asked to complete the post-game survey, as seen on Table 7. In order to validate the experiment and see the results unbiased we split up the participants between experimental and control. Those who were part of the control group followed the aforementioned set up of the experiment with the slight difference that they would not receive an adjustment based on their performance. Instead they would get a random map from the pool of maps they did not play.

The prediction of the new participants was made using the KNN algorithm from the package \texttt{class} [67]. Because of this, we used the previous data for the first study (Mixed Data subset) and used only those variables that we filtered with a correlation of more than 0.40. The value for K in this prediction was set up using the rule of thumb of $K = \sqrt{n}$, hence the value of K was set up to 6.

### 6.4 Data Collection

Participants were, for the second study, gathered from people who did not play the game before. This would allow us to give more validation to the experiment, and remove any possible bias by allowing people who had some prior experience with the game. The pool of participants, due to time constraints in the study, was constrained to 10. All participants were above 18 years old and had no prior experience to Pommerman as well as no prior understanding of the study that was taking place. The process of collecting the newly acquired data followed them same pattern as the first study did. The small difference this time was we only collected the variables that were needed for the model to work, as seen in Figures 21 and 22. However, since
the implementation of the adjustment occurred after the participant had played the three initials maps we had to minimize the data collection, processing and analysis to a minimum in order not to skew or bias the results. With this in mind another R script was built from the scratch to allow the data to be retrieved, formatted and input into the model once the participant finished playing the initial phase of the experiment. The only delay occurred where downloading the data directly from Google Polls. As we anticipated that would cause a small delay the script involved an automatic unzipping tool to allow the data be collected without any setbacks.

6.5 Data Analysis

In order to test if our model generalized across the new set of participants and if did it had an impact in their enjoyment we made used of the post-game survey results to test our hypothesis. Following a similar methodology we performed a basic review of the summary statistics for the results and reviewed if the difference in their scoring was statistically significant. However, we took into account that while the results might have not been statistically significant, in part due to the sample size size of this second study, it would not mean that the adjustment had no impact.

6.6 Results

As expected due to the sample size not statistical significance was encountered at all. However, as stated before that does not mean there would be no difference in their means. A quick summary statistics review of both the control group and the actual adjusted grouped showed little difference in the means of both groups. Although in part that is caused due to the sample size, it is worth noting however, that six out of seven questions indeed report a difference in their means. One can speculate that with a bigger sample group the difference could be higher.
6.6.1 Discussion

We attempted to answer our second research question in this study: *Does skill prediction model for estimating player skill apply consistently enough to other players? does the model generalize?*. Now the answer to that question might not be fully answered if we only looked at the statistical significance. Seeing the results we do get an overview of how the model behaved and what the results were. We noticed there was a difference in the means of both groups, experimental and control. However, given our sample size it is hard to discern whether that difference was purely coincidental or the results of the model. While this could be attributed to randomness, the difference in questions, Q3 and Q4 presented a small gap in their answers between both groups. This difference while not statistically significant it cannot be denied and completely discarded.

Now, has the model generalized across new participants? Unfortunately due to sample population and the scope of the study we cannot to reach a definitive conclusion. However, we do believe this work open new venues for player modeling and DDA
and how this approach can help to get a better understanding of how the perceived skill level of players can play a role in their in-game performance.
7 Conclusion

In this study we tried to answer two questions; These being *Can we predict player skill using gameplay features and self-reporting data?* and *Do those identified features for estimating player skill apply consistently enough to other players?* We tried to follow a different approach for DDA as a means to explore the possible relationship between DDA and player modeling and how these two could be combined could lead to more insight into the interaction between the player and the game. The first study focused on identifying those potential features and what impact the self-reporting data had. Conducting a User Study where participants were required to fill out a survey and complete a play session with three different levels gave us enough data to start investigating what features might had a bigger impact on the participants performance. We focused our efforts on clustering participants following a specific set of metrics and distances, while also assuming that there were three difficulty levels for any given participant. After much evaluation of the different clustering methods, correlation analysis, it was clear that certain variables have more impact on predicting the skill level of a participant than other.

The second study however, focused on leveraging those variables to apply our clustering model, by using the KNN algorithm for classification, into new participants and see if the model generalized through a this new pool of participants. This study differentiated itself from the first one as participants had to play and adjusted level after the first play session, and complete a post-game survey that recorded their interest and enjoyment in reference to the level that was adjusted based on their previously self-reported and gameplay data. However, and although the results displayed in the second study showed some difference between the participants who had their last map adjusted and those who had not, due to the sample size of the second study
the results could not be deemed as conclusive.

We believe that while the current work has provided a new approach for DDA in which information is filtered and self-reported data gives an extra layer of information, the sample size has been a drawback. This however, does not invalidate the work done. The clustering methodology has proven useful and insightful on analyzing the different aspects of gameplay and self-reported data.

7.1 Recommendations

While this study has tried to account for a lot of the complexity involved with clustering, it certainly had its limitations. Future work should focus on increasing the pool of participants and test other clustering algorithms and distances. Possibly mixing the approach of this study with Procedural Content Generation (PCG) could yield very interesting results given that the study adjusted the difficulty by the means of level layout. Additionally, while Pommerman \cite{54} has proved a very useful tool, it might require prior modification of its code if used for DDA, hence using another software that it is more oriented towards this type of study would be less straining and would reduce the workload.

We believe future work should focus on a more in depth User Study where more participants can be gathered and different metrics and algorithms can be tested to review their results. Additionally, given the nature of this study, we believe it can provide a deeper insight into the process of evaluating PCG.
References


## Appendices

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Type</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Categorical</td>
<td>Randomly generated ID</td>
<td>AGCFC</td>
</tr>
<tr>
<td>map_{e,m,h}</td>
<td>Categorical</td>
<td>Map the participant played</td>
<td>2</td>
</tr>
<tr>
<td>steps_all_{e,m,h}</td>
<td>Continuous</td>
<td>Total number of in-game steps</td>
<td>250</td>
</tr>
<tr>
<td>bombs_laid_{e,m,h}</td>
<td>Continuous</td>
<td>Number of bombs laid by participant</td>
<td>18</td>
</tr>
<tr>
<td>step_taken_all_{e,m,h}</td>
<td>Continuous</td>
<td>Total number of participant steps</td>
<td>254</td>
</tr>
<tr>
<td>blast_strength_{e,m,h}</td>
<td>Categorical</td>
<td>Bomb radius strength</td>
<td>4</td>
</tr>
<tr>
<td>can_kick_{e,m,h}</td>
<td>Boolean</td>
<td>Ability to kick bombs</td>
<td>0</td>
</tr>
<tr>
<td>block_destroyed_{e,m,h}</td>
<td>Continuous</td>
<td>Number of destructible walls destroyed</td>
<td>0.375</td>
</tr>
<tr>
<td>lives_{e,m,h}</td>
<td>Categorical</td>
<td>Participant lives</td>
<td>2</td>
</tr>
<tr>
<td>is_alive.1_{e,m,h}</td>
<td>Boolean</td>
<td>Agent 1 state (0-dead 1-alive)</td>
<td>0</td>
</tr>
<tr>
<td>is_alive.2_{m,h}</td>
<td>Boolean</td>
<td>Agent 3 state</td>
<td>0</td>
</tr>
<tr>
<td>is_alive.3_{h}</td>
<td>Boolean</td>
<td>Agent 3 state</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 10: Gameplay Variables
Pre Game Survey

I think I will do pretty well compared to other students

1 2 3 4 5 6 7

After playing a few times I think I can feel very competent

1 2 3 4 5 6 7

I think I am pretty good at playing video games

1 2 3 4 5 6 7

I am satisfied with my performance at playing video games

1 2 3 4 5 6 7

I am pretty skilled at playing video games

1 2 3 4 5 6 7

I can't do very well at playing video games

1 2 3 4 5 6 7

Please enter your player ID

Your answer

Figure 24: Screenshot of Pre-Game survey used for the First and Second Study
Post Survey

When answering these questions think about the last level you played.

I enjoyed doing this activity very much

1  2  3  4  5  6  7
☐  ☐  ☐  ☐  ☐  ☐  ☐

This activity was fun to do

1  2  3  4  5  6  7
☐  ☐  ☐  ☐  ☐  ☐  ☐

I thought this was a boring activity

1  2  3  4  5  6  7
☐  ☐  ☐  ☐  ☐  ☐  ☐

This activity did not hold my attention at all

1  2  3  4  5  6  7
☐  ☐  ☐  ☐  ☐  ☐  ☐

I would describe this activity as very interesting

1  2  3  4  5  6  7
☐  ☐  ☐  ☐  ☐  ☐  ☐

While I was doing this activity, I was thinking about how much I enjoyed it

1  2  3  4  5  6  7
☐  ☐  ☐  ☐  ☐  ☐  ☐

I thought this activity was quite enjoyable

1  2  3  4  5  6  7
☐  ☐  ☐  ☐  ☐  ☐  ☐

Player ID

Your answer

Figure 25: Screenshot of Post-Game survey used for the Second Study
Figure 26: Increased Range    Figure 27: Bomb-Kick    Figure 28: Increased Bomb count

Figure 29: These were the visual representations for the three different power-ups found in game. From left to right: Increased Range, Bomb-Kick, Increased Bomb count

(a) K-medoids Cluster Plot for Survey Data with Euclidean distance  
(b) K-medoids Cluster Plot for Survey Data with Manhattan distance

(c) K-medoids Cluster Plot for Survey Data with Gower’s distance

Figure 30: Survey Data K-Medoids Clustering Plots
(a) Hierarchical Cluster Plot for Survey Data with Euclidean distance  
(b) Hierarchical Cluster Plot for Survey Data with Manhattan distance  
(c) Hierarchical Cluster Plot for Survey Data with Gower’s distance  

Figure 31: Survey Data Hierarchical Clustering Plots
Figure 32: Gameplay Data K-Medoids Clustering Plots

(a) K-Medoids Cluster Plot for Gameplay Data with Euclidean distance
(b) K-Medoids Cluster Plot for Gameplay Data with Manhattan distance
(c) K-Medoids Cluster Plot for Gameplay Data with Gower’s distance
(a) Hierarchical Cluster Plot for Gameplay Data with Euclidean distance

(b) Hierarchical Cluster Plot for Gameplay Data with Manhattan distance

(c) Hierarchical Cluster Plot for Gameplay Data with Gower’s distance

Figure 33: Gameplay Data Hierarchical Clustering Plots
(a) K-Medoids Cluster Plot for Mixed Data with Euclidean distance
(b) K-Medoids Cluster Plot for Gameplay Data with Manhattan distance
(c) K-Medoids Cluster Plot for Gameplay Data with Gower’s distance

Figure 34: Mixed Data K-Medoids Clustering Plots
(a) Hierarchical Cluster Plot for Mixed Data with Euclidean distance

(b) Hierarchical Cluster Plot for Gameplay Data with Manhattan distance

(c) Hierarchical Cluster Plot for Gameplay Data with Gower’s distance

Figure 35: Mixed Data Hierarchical Clustering Plots