FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Assessing Game Player Profiles Through Agent-Based Simulation

Beatriz de Henriques Martins

DISSERTAÇÃO

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Prof. Rosaldo J. F. Rossetti

Second Supervisor: Tiago Costa

February 1, 2021

Assessing Game Player Profiles Through Agent-Based Simulation

Beatriz de Henriques Martins

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Prof. João Jacob
External Examiner: Prof. João Emílio Almeida
Supervisor: Prof. Rosaldo Rossetti

February 1, 2021

Abstract

Nowadays, data and information has become the main assets for the organizations. It is by exploring collected data every company can get to know their customers, predict their preferences and behaviours in different situations. Casinos are not different and for them it became even more crucial than other companies. Their business relies on all the environment that is surrounding the player. The aim of this dissertation is to profile casino games players using machine learning techniques and an agent-based simulation in order to create a simulation model and understand how different profiles will react to the same game. The target are players whose preferences are slots machine games and explore their influence on the game experience. This project will be divided into two parts. The first one will use machine learning techniques as a tool to conduct and develop a study about the behaviour of both players and games to create the profiles using data collected from real casinos (bwin.party). The second part is to create an agent-based simulation where the agents are the players and the game. This simulation will allow to simulate a game environment of a casino and the player's actions.The expected outcome of this project is to create a simulation model with an agent-based simulation to prove and test the player's behaviour for improvement of the games. With this project, the company will be able to test the most complete, easier and fastest way possible their games before sending a new update to the market as well as to better understand their clients.

Keywords:
Agent-Based Simulation; Simulation; Behavioural Modelling; Serious Games; Machine Learning; Classification

ACM Categories
Computing methodologies→Modeling and simulation→Simulation types and techniques→Agent/discrete models
Computing methodologies→Machine Learning

Resumo

Atualmente, a informação e os dados tornaram-se num dos maiores ativos para as organizações. É através da exploração dos dados recolhidos que as empreas conseguem conhecer os seus clientes, prever as suas preferências e comportamentos em diferentes situações. Os casinos não são diferentes e, para eles, torna-se ainda mais crucial do que para outras empresas. O negócio dos casinos depende de todo o ambiente envolvente ao jogador. O objetivo desta dissertação é criar perfis de jogadores de jogos de casino recorrendo a técnicas de machine learning e uma simulação de agent-based, com o objectivo de criar um modelo de simulação para compreender como diferentes perfis irão reagir ao mesmo jogo. Os alvos são jogadores cujas preferências são jogos de slot para explorar se o comportamento destes tem alguma influência na experiência de jogo. Este projeto será dividido em duas partes. A primeira parte irá recorrer a técnicas de machine learning para conduzir e desenvolver um estudo sobre o comportamento de jogadores e jogos com a finalidade de criar perfis de jogadores usando dados reais recolhidos em casinos. A segunda parte irá criar uma simulação agent-based em que os agentes são os jogadores e o jogo para simular o ambiente de jogo num casino e as acções do jogador no mesmo.O objetivo deste projeto é criar um modelo de simulação agent-based para provar e testar os diferentes comportamento do jogador. A intenção deste projeto é ajudar a empresa reponsável pela produção de jogos para casinos a testar os mesmos de uma maneira mais completa, fácil e rápida os seus produtos antes de enviar uma nova atualização para o mercado, bem como a entender melhor seus clientes.

Keywords:
Agent-Based Simulation; Simulation; Behavioural Modelling; Serious Games; Machine Learning; Classification

ACM Categories
Computing methodologies→Modeling and simulation→Simulation types and techniques→Agent/discrete models
Computing methodologies→Machine Learning

Acknowledgements

Eu gostaria de agradecer à minha família e ao meu namorado, Ricardo, por todo o incentivo, suporte que me deram ao longo de todos estes anos e por acreditarem em mim quando não o fiz. Aos meus pais pela paciência que tiveram, pelo carinho incondicional, por nunca me deixarem desistir e por todas as vezes que não puderam falar apenas para me ouvir. Em especial à minha mãe pela guerreira que é, e por todas as vezes que lutou por mim e por ela. Às minhas irmãs por me fazerem rir quando achava que estava tudo perdido e partilharem as suas visões mais leves. Ao Ricardo pela paciência, carinho e todos os doces que me trouxe para me animar. À minha gata, Kitty, por todos estes anos que me ouviu pacientemente a treinar as minhas apresentações. Aos meus avós por partilharem a sua sabedoria comigo e pelo carinho que sempre me deram. À Beatriz Santos pelos passeios na FEUP. À Elisa Duarte pelos longos telefonemas.

Um agradecimento especial aos meus orientadores, Professor Rosaldo Rossetti e Tiago Costa, e à Fabamaq por terem sempre acreditado em mim e tornado este projecto possível.

Sem vocês este trabalho não seria possível.

Obrigada!

Beatriz

"Well done is better than well said."

Benjamin Franklin

Contents

List of Figures

List of Tables

Abbreviations

AG	Addictive Gambler
AI	Artificial Intelligence
BB	Black-box
BBMG	Browser-Based Multiplayer Game
BDI	The belief-desire-intention
BDS	Bot-Detection System
BRF	Belief-Revision-Function
DG	Digital Game
EG	Electronic Games
FPS	First-Person Shooter
HCI	Human-Computer Interaction
ML	Machine Learning
NPCs	Non-Player Characters
RG	Responsible Gambling
RGC	Responsible Gambling Case
RGI	Responsible Gambling Intervention
RL	Reinforcement Learning
RTP	Return to player
SG	Social Gambler
SP	Synthetic players
SUT	System Under Test
TA	Test Automation
UI	User Interface
VG	Videogames

Chapter 1

Introduction

1.1 Context and Motivation

Nowadays, every company wants to know everything about their clients. Predicting their behaviour in different situations as well as their preferences can result in a huge success as it helps companies adapting their objectives, shifting their vision and accommodate customer's desires. Casinos are not different and for them it became even more vital than other companies, due to the high competitive market they play. Casinos work on emotions, moods, and behaviours stimulated by music, environment (physical and social), game graphics, type of game, and many other factors. All these aspects need to be studied and taken in consideration when improving games. Software houses that develop video games for casinos collect data to better understand the needs of the actual market. They desire to leverage the potential hidden in all data they collect daily and stand out in the sector.

The biggest motivation of this project is to improve the knowledge of the company in regards to their players as well as to increase the quality of their games as well. Currently the organization has awareness to the different types of players but it is unclear how they react to the game features, amount of balance they have and other factors that are being gathered on every event of the players during their stay on the casino.

1.2 Problem and Goals

The aim of this dissertation is to profile casino games players using machine learning techniques and an agent-based simulation in order to create a simulation model to understand how different profiles will react to the same game. The target are players whose preferences are Slot Machine to explore if their behaviour has any influence in the game experience. The intention of this project is to help the company that is producing the games for casinos to test the most complete, easy, and fast way their products before sending a new update to the market as well as to better understand their clients. To achieve this, the company will use this project to incorporate the test scenarios into their current quality-assurance stage and make sure different play-styles don't affect the game's

performance and reliability. After this, with the information we'll create from raw data, there's space for improving the games and innovate according to customer's expectations and will of play.

1.3 Structure of Document

This document is composed of five chapters. The Chapter 1 is about the introduction, in which we describe the context, motivation, problem we are solving and goals. The Chapter 2 shows us an overview of past and different approaches with similar objectives. The Chapter 3 is the explanation of methodological approach for the work will be developed in this dissertations. The Chapter 4 contains all the implementation details and the results from the simulation. Finally, the Chapter 5 is the conclusions and future work of the developed project.

Chapter 2

Literature Review

2.1 Using Machine learning as a tool for Software Testing

2.1.1 Machine Learning

Over the last few years, Machine learning (ML) has received increasing attention from diverse areas, from politics to medicine. This recognition happens because ML is an application of Artificial Intelligence (AI) that provides systems with the ability to accurately learn and improve from past experiences without being explicitly programmed. By W. James Murdoch et al. [18], ML models have demonstrated great success in learning complex patterns that enable them to make predictions about unobserved data in several areas.

2.1.2 Test Automation

Test Automation (TA) is a method in software testing that makes use of specific software tools to control the execution of tests, furthermore compares actual test results with predicted/expected outcomes. TA is used to add additional testing that may be too difficult or too long to perform manually and without human error. By Rajesh Mathur et al. [13, 23], TA is weakened by limitations provoked by current tools and methodologies. They have described four significant challenges of TA, particularly related to user interface (UI) testing:

1. **Test case generation is manual and labour intensive** - there are many methods used for generation of test scripts within current tools. The best part of tools rely in test script what actions should occur and should be the result of each task or to record steps to be replayed in the exact sequence. The tools did not try to understand the intended function of systems and automatic suggest a possible set of test scripts. They stated while a tool may not be able to determine all of the expected behaviour of a system, many aspects are common to all systems or can be learned based in previously scripted tests.

2. **Test scripts are not validated until runtime** - Standard testing tools use an object repository to map user-friendly object name with a list of objects parameters to identify the object.

Scripts are written with a tool test no offer live validation to the script present in the repository, most of the times. For example, if the command to click the object scripted are "OK Button", there is no validation if "OK Button" exists until the script is executed. On the other hand, if the authors write "Enter Text" on the button, this will may not be detected until the scripts run. One way to solve this limitation is verifying the objects used in a script when it is created or modified, still if the item is removed or modified later the script has been broken, and there may not be any indication about it.

3. **Test scripts are brittle** - when an automation script fails most of the frameworks will do one of the two things: skip remaining steps and fail the whole test or fail the single step and continue to next step which will most likely fail as a result of the missed step. Usually, these fails occur when an existing change affects the parameters being used to identify the object. If the element has suffered an alteration, name or type, the tool will no longer be able to identify the element, causing a waterfall of failures. Maintenance of these alterations is high expensive and involves a significant investigation, updating and repetition of sequential tests to verify the correct script. Another cause can be insignificant visual changes, like changing a button to rounded corners or the location of a button by 0.5 pixels or the background in a screenshot comparison. For last, another cause can be unexpected occurrences such as pop-up windows or warnings dialogues or additional steps. Such trivial causes can break the automated script and are difficult to prevent. All these challenges should be brought to the tester's attention.

4. **Maintaining a test suite is expensive** - Rajesh Mathur et al. [13] declares that automation is most commonly applied as a regression check for features that have reached a mature stage in their life cycle. That means the features need updates like adding a new filed to dropdown or changing the text on a paragraph or adding a new hyperlink. However, these trivial changes require a lot of effort behalf to update every script related to the page.

Following these line, Vahid Garousi et al. [7] defined at least six activities related to TA could be automated, as explained below (figure 2.1).

2.1.2.1 Test-Case Design

By Vahid Garousi et al. [7], Test-Case Design means to follow a set of test cases, with the input and the expected output values, or test requirements, control flow paths to cover. This task can be done by human knowledge, exploratory testing, or on criteria, line or requirements coverage. In manual design, the tester takes a look over the requirements and/or code and write the test cases without using any tool, for example, test case design based on human knowledge. In partial automation, the tester uses any of the many available code coverage tools but manually write test cases to increase coverage. In fully automated criteria-based test-case design, the testers need to utilise the knowledge available in search-based test data generation, including inputs and expected outputs.

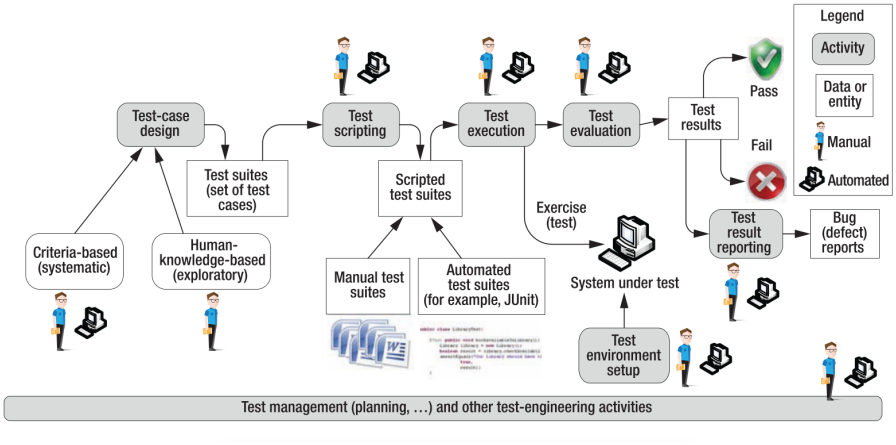

Figure 2.1: Automation across the software-testing process. Each testing activity offers a tremendous opportunity for automation

2.1.2.2 Test scripting

Testers have been performing test scripting for many years, manually or automated, for use in test execution. For partial automation, for example, IBM developed a solution that runs the test and stores the document for later manual execution. That happens because the tool lets developers test through GUI's, called to record-and-playback tools. They made the developers record test cases (test scenarios), while the tool automatically records the test log as a test script in the background. After that, the test can be replayed as many times as developers want. This approach has several limitations related, for example, can requires skilled labour, time-consuming and can produce fragile scripts. Fully automated test scripting came to combat previous faults. This approach uses a novel combination of natural-language processing, backtracking exploration, runtime interpretation and learning to help testers efficiently create automated test scripts from manual test cases.

2.1.2.3 Test execution

Test execution run test cases in the System Under Test (SUT) and store the results and/or observe the SUT's output and/or behaviour. Test execution is affected by previous decisions, because of that if a team runs the test cases manually, the test execution will be manual. If the team runs the cases automatically, the test execution will be fully automated. The only test execution be partially automated is some scripts runs automatically and others manually.

2.1.2.4 Test evaluation

There are three ways to evaluate the test outcome:

1. A human tester make the judgment;

2. The developers add (hard-code) test evaluations as verification points (assertations) in the test code ;

3. The developers build "intelligent" (learning) test oracles, using ML and AI;

The second and third approaches are automated test evaluations but with different levels of intelligence.

2.1.2.5 Test results reporting

Usually, this is the last step. On this phase, a report with the verdicts and deficiencies of the system is made. The report is often manually but exists frameworks and techniques to do automated but reports.

2.1.2.6 Test management and other test-engineering activities

These activities include a set of test for redundancy, regression test selection and repair when SUT has changed.

2.1.3 Related Work

With the increasing dependency on technology for simple and complex tasks, rise the necessity of software be more safety and quick to come out to the market. To achieve these factors, researchers are continually developing new TA solutions, with more concentration on mobile applications. Many apps are tested manually by developers following defined questionnaires. However, this approach is not viable because many of these tests are hand-coded developed for specific scenarios, and it can't be applied to new updates or apps. As a result, it requires spending many resources to reuse these tests and more resources in high maintenance.

With this in mind, Ariel Rosenfeld et al. [21] developed a study to testing several apps and find as many functional bugs as possible. Their approach is based on the premise that different activities in an application share a similar interface structure. They use machine learning techniques to classify each activity in the application into one seven pre-defined activity types. On their studies, they use two different approaches: first, using 26 randomly selected applications, they show that their strategies can successfully instantiate and reuse generic functional tests and uncover functional bugs. Second, in a human study with two experienced human mobile testers, they show that their strategies can automatically cover a large portion of the human tester's work suggesting significant potential relief in the manual testing efforts.

Another example of related works with ML and TA, was developed by Rajesh Mathur et al. [13]. Their project consists of a framework to work on automated user interface-driven testing. This project is a proof of concept to demonstrate self adaptative recovery test script and can also be

applied to testing API, performance test and accessibility testing. Their framework can automatically generate tests directly from an application under test, reducing the cost of implementation. They support their analysis on three different stages explained below by Rajesh Mathur et al. [13]:

- **Test design** is supported with live validation of referenced objects in the test scripts;
- **Test execution** is made robust by adaptive self-recovery, using a combination of techniques to recover from missing elements or unexpected events;
- **Test maintenance** is automated or semi-automated by directly linking changes to the application to updates in the test script.

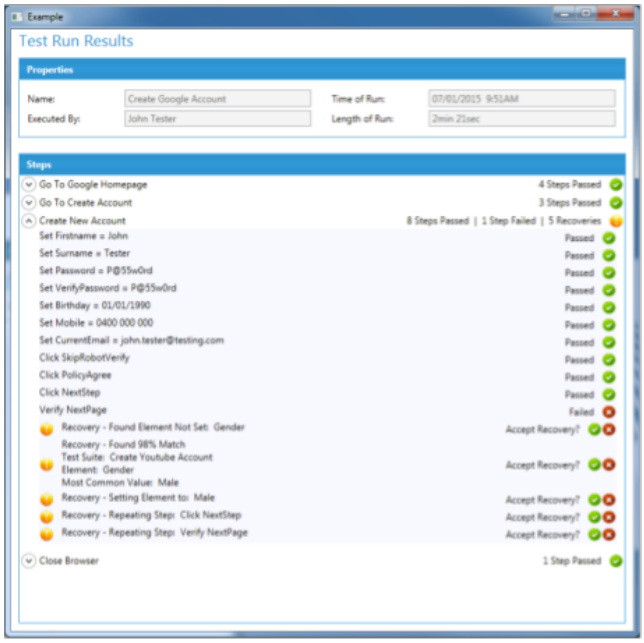

Figure 2.2: A screenshot of the adaptive automation framework prototype showing the results of a test run with adaptive error recovery [13].

ML and TA can be applied to a huge set of areas, including security protocols. An example is a work developed by Guoqiang Shu et al. [23], they conducted a study about how can they discover an internal structure of black-box (BB) implementations and checking security protocols message confidentiality through ML. In this case, they use ML to learn procedures and test security protocols on-the-fly with a conformance test generator serving as a teacher. For BB protocol implementation under test, they maintain a conjecture model of its structure and update it successively as more is learned by applying testing sequences. The process is over when no more new behaviour could be learned, or a security violation is detected. They declare the new testing architecture modifies and applies the classic supervised machine learning algorithm to explore the structure of black-box implementations for checking security properties and claim their approach highly effective.

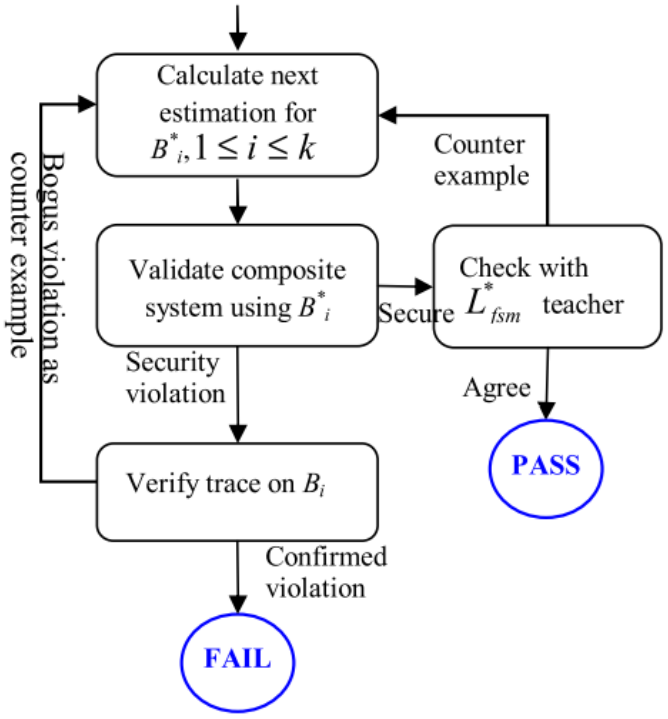

Figure 2.3: Automata used by Guoqiang Shu et al. [23]

2.1.4 Summary

In conclusion, TA and ML are currently two main initiatives for software companies. These technologies can increase the performance of applications and find errors otherwise will be found during the production stage by customers or possible customers. The developers will have more time to solve potential problems and focus on innovating the methodologies used. Despite being recent technologies, ML and TA working together can make significant progress on the "software world".

2.2 Using Agents with Bots on Digital Games

2.2.1 Digital games

Digital games (DG) have a set of definitions related. Frederico S Miranda et al. [16] have compiled as many definitions as they could. An example of a DG definition is, by Salen and Zimmerman [9], a synonym of electronic game. These games are executed by personal computers, consoles (for example PlayStation), portable devices (for example Nintendo, Tablets or Phones) or arcade machines. DG is a system as any other game, with the difference of having hardware and software as essential components. Another example is defined by Kirriemuir [11] as DG being executed by videogames (VG), personal computers and portable devices. These games exhibit the following features:

1. Give digital visual information to players;
2. Receive some input from players;
3. A set of programmed rules processes the entries;
4. Change their digital data.

By Kerr [10], DG refers to all the games executed in arcade machines, computers, consoles and phones. Wolf [29] declares DG is synonymous with VG - consoles or computers run games. Mosca [17] does the differentiation of electronic games (EG), DG and VG:

- **EG**: those who use electricity to operate. For example pinball;
- **DG**: hey only need software to be played, without video output, such as audio games that use only sound resources and textual information;
- **VG**: games that have the software and video output, such as the Tomb Raider game.

In summary, Frederico S Miranda et al. [16] considered the word "Game" as an activity composed by rules well defined and clear objectives, capable of involving the players in the resolution of conflicts and which has a variable and measurable result. They created a definition of all the terms related to the concept of Game.

- **Activity**: a voluntary action, serious (for example a professional game of volleyball) or not (for example volleyball played in schools during physical education classes), with material interest (for example large Las Vegas casinos) or not (for example online poker game or just to pass the time). This activity can be seen as a set of interrelated parts;
- **Rules**: determines what can or cannot be done within the game;
- **Objectives**: clear goal to be achieved;

- **Involved players**: used in the broad sense, it means that it unites people around a common goal, enabling engagement, decision making and interaction with other players or with the game itself;
- **Conflict resolution**: problems and challenges that must be overcome to achieve the objectives;
- **Variable result**: these are uncertain results, which cannot be predicted. A game that before the winner is known, has a good chance of not attracting participants;
- **Measurable result**: metrics to reach the final result, for example, scoreboard, points, fastest time and progress bar.

In summary, Frederico S Miranda [16] defined DG as a voluntary activity, with or without material interest, with serious purposes or not, composed of well-defined rules and clear objectives, capable of involving the players in the resolution conflicts and that has variable and quantifiable results. This activity must be managed by software and performed on hardware.

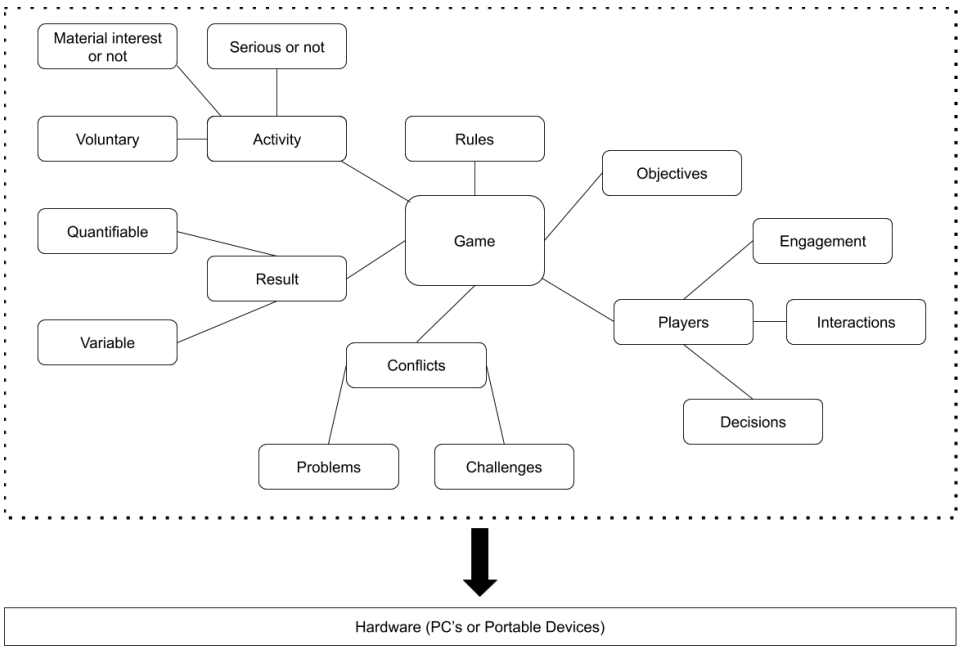

Figure 2.4: Relation between the terms by Frederico Miranda et al. [16]

2.2.2 Related work

DG became one of the biggest and more profitable areas of entertainment in the world. However, bots weren't developed only with "good intentions". A good example is the work developed by a master's student from the Norwegian University of Science and Technology, Erik Wendel [28].

On his master thesis, he studies cheater's bots in massively multiplayer online games. He found out a group of malicious websites selling or distributing subscriptions to cheats, depending on the game gender, despite the fact that some companies have software anti-cheaters. By Eric [28], many cheats are still able to bypass the security mechanisms provided by those companies and it is a continuous arms race between cheat developers and anti-cheat developers - just like in the virus and anti-virus industry. The way he proposes to solve the problem is an approach based in Browser-Based Multiplayer Game (BBMG). He wrote for a non-disclosed BBMG, and reviewed their performance in several tests quantitatively by playing several game accounts in parallel. The primary purpose of Bot-Detection System (BDS) is to detect the use of bots in BBMG by identifying a set of characteristics and relating these to standard human behaviour.

Another application of bots relies on the fact today DG need to be very dynamic to become competitive on the market. The players, professionals or not, have passion in adaptative games. This means games need to be able to grow in difficulty during the gameplay. For this reason, Ushma Kesha Patel et al. [19] conducted a study where they create bots to collect and test news ways of improving the game during the match and develop new methodologies to improve AI in online games. To achieve this result, they created several agents, non-player characters (NPCs), with abilities such as learning, decision-making, prediction, reasoning, planning, and scheduling. The main objective of this study explores techniques to find Nash Equilibrium using fictitious play. By Ushma Kesha Patel et al. [19] fictitious play being a lightweight algorithm can be used in an internet-based virtual environment for three-dimensional games.

Many behaviours on first-person shooters (FPS) were hard-coded making them static and hard to expand. To solve this challenge, Michelle McPartland et al. [15] conducted a study based in reinforcement learning (RL) to multi-purpose bot with several behaviours like navigation, item collection and combat. By Michelle et al. [15] RL can be used to find bugs in the code, for example, the RL bots might learn an easy exploit in killing the enemies, which will be prevalent when replaying games. However, RL had several challenges associated like scaling with increased complexity and stayed trapped in a cycle of repetitious states. Their study tries two different approaches by splitting the problem space into smaller tasks to combine them to produce the overall behaviour set. They tested three different types of RL. The first method would combine a previously trained combat and navigation controller using hierarchical RL. The second method would use simple rules to determine when to use the combat or navigation controller. The third method would use RL to relearn the tasks of combat and navigation simultaneously. For last, they compare the results with a random controller and a state machine controlled bot. They concluded both rule-based and hierarchical RL are good options in different situations. The hierarchical RL must be used to remove predictable behaviours and diversify them while the based-learned showed better results when the action set was hard-coded.

2.2.3 Summary

In conclusion, bots are an exciting technology with a variety of purposes. They can be applied to either bypass security mechanisms or fight malicious purposes, which means that organizations can enter a race battle in which the fastest and more advanced bot wins. We can use bots to conduct a study about the future features of DG before launching on the market, testing new approaches for new methodologies, collecting data or testing theories, resulting in challenging and attractive outputs.

2.3 Modeling agents

Georgios N. Yannakakis et al. [30] conducted a study about human-computer interaction (HCI) to understand how individual players experience the interaction with a game. They argue that the main objective of study players was the understanding of player's cognitive, affective and behavioural patterns. At the same time, we can use games to create arenas for eliciting, evaluating, expressing and synthesising experiences. They believe in conducting a study to achieve tangible results, they need to utilise real players instead of NPCs for two main reasons [30]:

1. an NPC is coded, therefore a perfect model for it exists in the game's code, and is known by the game's developers;

2. one can hardly say that an NPC possesses actual emotions or cognition.

However, according to them, it should be interesting to use NPCs for testing player modelling techniques and compare the results with the actual model and the model founded. Another approach can be an integral component of adaptive AI that changes its behaviour to respond to its opponent dynamically.

They make a difference between player modelling and player profiling: player modelling means modelling complex dynamic phenomena during gameplay interaction such as player cognitive, behavioural and affective states based in data or theories, while player profiling means the categorisation of players based in static information that did not alter during the gameplay such as personality, cultural background, gender and age. Their work was divided into three different stages [30]:

1. **Input**: Game (gameplay, objective and context) Player profile;

2. **Computational (Player) Model**: Model-based (top-down) and Model-free (bottom-up);

3. **Output**: Regression (scalar values), Classification (Classes), Preference Learning (Ordinal data, e.g. ranks, preferences) and Unsupervised Learning/Clustering (no output).

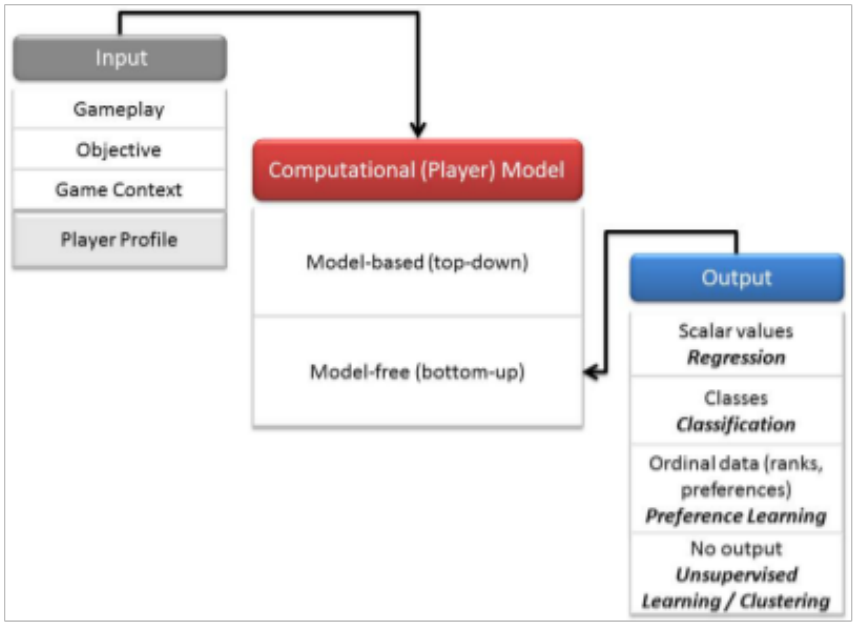

Figure 2.5: Player Modeling: the core components[30].

In regards to the second stage, the computational model was approached into two different ways. The first one was model-based (top-down). This approach was based in emotion theories (e.g. Russell's circumplex model of affect in which emotional demonstration was directly linked to emotional stated like increase heart rate) and general theoretical framework of behavioural analysis and/or cognitive modelling (e.g. usability theory, the belief-desire-intention (BDI) [1, 2, 24] or cognitive theory). Several challenges of a top-down approach were based on measuring the direct link between challenge and fun. According to the authors, note that even though the literature is rich in theories on emotion, caution is warranted with the application of such theories to games (and game players), as most of them have not been derived from or tested in ergodic media such as games. Another author, Calleja [4], for instance, reflects on the inappropriateness of the concepts of 'flow', 'fun' and 'magic circle' (among others) for games. The second was model-free (bottom-up) approaches. This approach refers to the construction of an unknown model (mapping) between the player (input) and a player state representation. In other words, model-free strategies follow the method of working of the exact sciences, where observations are gathered and examined to generate models without an initial assumption on what the model looks like. To create the model, they use annotated players state and players data. With model-free approaches, they try to model and predict player actions and intentions. They also used game data mining to identify different behavioural playing patterns. This approach is common in studies about psychophysiology in games and for facial expression and head pose recognition. To make the results more realistic as possible, the team used a hybrid approach [30].

By the authors, they needed to use three different types of data [30]:

1. **Gameplay input**: anything that a human player is doing in a game environment gathered from gameplay data (i.e., behavioural data). This type of data is linked to the player's cognitive processing patterns and cognitive focus. On the other hand, cognitive processes can influence all player experience. The metrics of the game are statistical spatio-temporal features of game interaction;

2. **Objective input**: objective data collected as bodily responses to game stimuli such as physiology and body movements. These measures are taken by physiological signals, and they obtain through photoplethysmography, electrocardiography, electroencephalography, galvanic skin response, respiration, electromyography and pupillometry. This intrusion through game space will affect the final results and into account when interpreting data;

3. **Game context input**: the game context which comprises any player-agent interactions but also any type of game content viewed, played, and/or created.

To create a player profile, the team needed to gather all the information static or not about the player. They use a stereotype approach, trying to automatically assign a player to a previously defined subgroup of the population, for which key characteristics have been defined. The responses of each player will be chosen according to the stereotype assigned.

According to Georgios N. Yannakakis et al. [30], the model's output is usually a set of particular player states. Each state can be represented for a scalar (e.g. a vector of numbers) or a class to map the state (emotional dimension or a behavioural pattern) or a preference. Still, the final report must be generated by unsupervised learning because existing instances where players states are not available. The team used several ways to annotate the players state like asking directly to the player what they feel in the moment or requiring for the players answer questionnaires.

In summary, the researchers faced several challenges in conducting this study. One of the significant challenges was the lack of public data available. They said they needed a rich multimodal corpus of gameplay, player data and player descriptions. Some examples were detailed gameplay for several games, a large number of players, actions, events, locations, biometrical data and demographic data. As procedural content generation techniques for design and improvement of games reached commercial interest, future features or games will have more user-generated or procedurally-generated content.

2.4 Different types of games

By Jan McMillen [14], chance games are one of the few social activities that occur in nearly all cultures and in every period of time: in this respect, it can be said to be virtually a universal phenomenon in human societies.

2.4.1 Gaming vs Gambling

As defined by the oxford dictionary, **gaming** means playing computer games whether through consoles, computers, mobile phones or another electronic device. Gaming is a term that suggests usual **gameplay**, probably as a distraction and/or hobby. Although traditionally a lonely form of amusement, online multiplayer video games turned into gaming a popular group activity as well.

Defined by the oxford dictionary, gambling means the activity of playing games of chance for money and of betting on horses, etc. With the consciousness of risk and hope of gain, on the outcome of a game, a contest, or an uncertain event whose result may be determined by chance or accident or have an unexpected effect by reason of the punter miscalculation.

2.4.2 Categories

We can summarize all the types into four main categories and divide the classes into subcategories and sub-subcategories in order to add some information about them.

2.4.2.1 Party games

Party games is the name given to the set of games to be played at social gatherings to facilitate interaction and provide entertainment and recreation. Party games are designed to be easy for new players to learn. Some examples of party games are pong, conversation, guessing games, singing games, drinking games and power games.

Figure 2.6: Party guests play a fortune-telling game with cards

2.4.2.2 Tabletop games

Tabletop games are games that are normally played in a table or other flat surfaces, such as board games, card games, dice games, miniature wargame, or tile-based games.

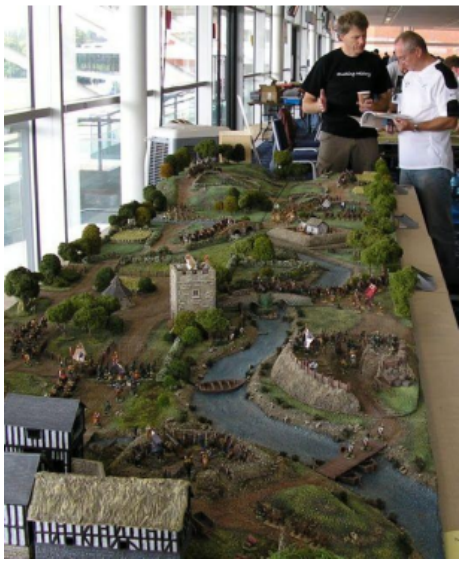

Figure 2.7: Example of a wargame table

As per In-game theory, there are two fundamentally different elements of chance that can play a role:

1. Chance due to outcome uncertainty, e.g. due to dice rolls or due to unknown cards being dealt with during the game;

2. Chance due to state uncertainty, e.g. due to the opponent's position or cards not being visible, or due to the simultaneous move character of the game.

In line with this, games in which outcome uncertainty plays a role are referred to as stochastic games, as opposed to deterministic games. Games in which state uncertainty plays a role are referred to as partial or imperfect information games as opposed to full or perfect information games.

	Full information	Partial information
Deterministic	- Chess - Go - Checkers	- Battleship - Mastermind
Stochastic	- Monopoly - Roulette - Craps	- Poker - Blackjack - Scrabble

Table 2.1: Some examples of classification of games in game theory

2.4.2.3 Videogames

Videogames are an electronic game that involves interaction with a UI or input devices, such as a joystick, controller, keyboard, or motion-sensing devices, to generate visual feedback on a video display device such as a TV set, monitor, touchscreen, or virtual reality headset. These games are augmented with audio feedback from speakers or headphones, and optionally with other types of feedback systems, including haptic technology.

2.4.2.4 Other games

Other games are all the games that don't fit on any of the above categories. This category had all the different types of games without a specific group for them. Some examples are chance games, educational games, multiplayer games and others.

2.4.3 Casino case

In a casino case, several categories can be applied, such as tabletop games (games with cards or dice) or videogames (slot or video bingo).

Recently, the online casino has revolutionized casino gaming forever. It's now possible to play every single game from the physical casinos, literally, anywhere.

2.4.3.1 Type of games

Slots

A Slot machine is a gambling machine operated by inserting coins into a slot and pulling a handle that activates a set of spinning symbols on wheels, the final alignment of which determines the payoff that is released into a receptacle at the bottom. Exist three types of slot machines:

- mechanical - pulling a handle that activates a set of spinning symbols on wheels;
- electrical - pushing a button that activates a set of spinning symbols on wheels on a screen;
- mechanical electrical - a combination between the two above.

The first slot machine was invented in 1889 by Charles Fey, the Liberty Bell. During several years, slot machines paid the prizes with "gums" or "amusement coupons" to be discounted on cigars, drinks or cash.

Figure 2.8: Example of slots game

Blackjack

Blackjack is a card game where the objective is to have a hand where the value of the set of cards is higher than the dealer's hand and less than twenty-one.

In the past blackjack was a very played game on big saloons. Nowadays, blackjack is mostly played online and when played on a real casino is on defined schedules and rooms.

Figure 2.9: Example of Blackjack table

Roulette

Roulette is a gambling game in which a ball is dropped on to a revolving wheel with numbered compartments, the players betting on the number at which the ball comes to rest. The ball and the wheel spin on different sides. Nowadays, the most common way of playing roulette is on videogames.

Figure 2.10: Example of Roulette table

Craps

Craps is a dice game in which the players make bets on the outcome of a pair of dices. The bets can be placed before a single throw of the dice or on the prediction of the next set of throws. Players may wager money against each other (playing "street craps") or a bank (playing "casino craps"). Because it requires little equipment, "street craps" can be played in informal settings. While shooting craps, players may use slang terminology to place bets and actions.

Figure 2.11: A craps table with a game in progress

Poker

Poker is any of a number of card games in which players wager over which hand is best according to that specific game's rules in ways similar to these rankings. Often using a standard deck, poker games vary in deck configuration, the number of cards in play, the number dealt face up or face down, and the number shared by all players, but all have rules which involve one or more rounds of betting.

Baccarat

Baccarat is a card game played at casinos. It is a comparing card game played between two hands, the "player" and the "banker". Each baccarat coup (round of play) has three possible outcomes: "player" (player has the higher score), "banker", and "tie". There are three popular variants of the game: punto banco (or "North American baccarat"), baccarat chemin de fer (or "chemmy") and baccarat banque (or à deux tableaux). In punto banco, each player's moves are forced by the cards the player is dealt. In baccarat chemin de fer and baccarat banque, by contrast, both players can make choices. The winning odds are in favour of the bank, with a house edge no lower than around 1%.

Bingo

According to Cambridge Dictionary, bingo is a game of chance often played for prizes in which people are given cards and cover each number on their cards as it is called, with the person who is first to cover a whole line of numbers on any card being the winner. Usually, the cards had three lines by five columns, despite existing variants like cards with three lines by nine columns or five lines by five columns. No later than thirty numbers are extract from a set of 1 to 60 or 1 to 90. Usually, the columns follow these rules:

	60	90
Column B	1 to 12	1 to 18
Column I	13 to 24	19 to 36
Column N	25 to 36	37 to 54
Column G	37 to 48	55 to 72
Column O	49 to 60	73 to 90

Table 2.2: Number by column in Bingo

When the player covers a line, column diagonal or the four corners he will receive a prize. The jackpot (Bingo) is determined when all the numbers of the card are marked.

VideoBingo

VideoBingo is very similar to Bingo. The rules and prizes are very similar. The extraction of the thirty numbers occurs in one time. The player can only win the jackpot when the first extraction covers all the numbers of the card. When the player covers all the numbers buying extra balls, the prize will be assigned the award previously defined in a prize table.

2.5 Models of virtual players

Several decades ago, it was impossible to imagine how the game industry would grow and would become one of the more prominent industries of the world. The small and amateur companies become more significant and complex. These days, to launch a game to the market, the team is composed of between 20 to 50 people and needs nearly two years to develop and test the new game. This new age brings new challenges, with the fast growth of the industry, like cheating and crime prevention.

2.5.1 Model-View-Controller

According to James Bucanek [3], Model-View-Controller (MVC) describes the architecture of a system of objects. It can be applied to isolated subsystems or the entire application. Whereas most of the patterns address specific problems. MVC design pattern is also less clearly defined than many other patterns, leaving a lot of latitude for alternate implementations. The MVC pattern is simply stated:

- Data model objects encapsulate information;
- View objects display information to the user;
- Controller objects implement actions;
- View objects observe data model objects and update their whenever it changes;
- View objects gather user input and pass it to a controller object that performs the action.

The key to successfully implementing an MVC design is to pay close attention to the role of objects and the communications between those objects. The first three rules define the role of your objects.

- **Data model** objects store, encapsulate, and abstract your data. They should not contain methods or logic specific to making your application function. For example, a data model class should implement a method that serialises its data, but it should not implement the "Save As..." command. Even if the two functions are nearly identical, the code that deals with the abstract data transformation should be implemented in the data model object and the code that implements the user command should be implemented in the controller object.
- **View** objects display the information in the data model to the user. View class design runs from the extremely generic to the very specific; generic view classes are provided by the framework to display almost any kind of string, number, or image, while you are more likely to implement very specific view objects designed for your application. View objects also interact with the user and initiate actions by interpreting user-initiated events, such as mouse movement and keystrokes. It converts those events into actions that are passed to the controller object for execution. The event and resulting action are often very simple;

clicking the mouse over a button object will send an action message to a controller. Complex gestures, like drag-and-drop, are more involved.

- **Controller** objects implement your application's actions. Actions are usually initiated by view objects in response to user events.

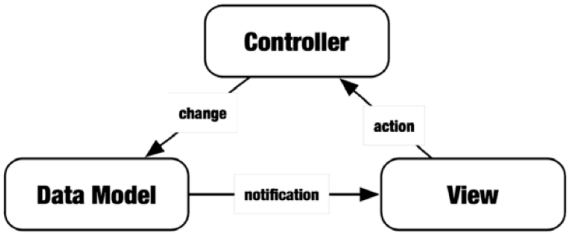

Figure 2.12: Fundamental MVC communications [3]

2.5.2 Pattern Recognition

Pattern recognition is the process of recognising patterns by using machine learning algorithms. Pattern recognition can be defined as the classification of data based on knowledge already gained or in statistical information extracted from patterns and/or their representation. One of the essential aspects of pattern recognition is its application potential.

In a typical pattern recognition application, the raw data is processed and converted into a form that is amenable for a machine to use. Pattern recognition involves classification and cluster of patterns.

- In classification, an appropriate class label is assigned to a pattern based in an abstraction that is generated using a set of training patterns or domain knowledge. Classification is used in supervised learning.

- Clustering generated a partition of the data which helps decision making, the specific decision making activity of interest to us. Clustering is used in an unsupervised learning.

2.5.3 Decision Making

According to Bohanec [25, 26], decision making (DM) is usually defined as a mental process, which involves judging multiple options or alternatives, in order to select one, so as to best fulfil the aims or goals of the decision maker. Therefore, there are two main components involved in decision making: the set of alternatives, judged by the decision maker, and the goals to be satisfied with the choice of one alternative. The output of this process can be an action or an opinion of choice.

DM is a process. This means that in general it takes some time and effort until the choice is made, involving several activities, such as:

- Identification of the decision problem;

- Collecting and verifying relevant information;
- Identifying decision alternatives;
- Anticipating the consequences of decisions;
- Making the decision;
- Informing concerned people and public of the decision and rationale;
- Implementing the selected alternative;
- Evaluating the consequences of the decision.

2.5.3.1 Levels

According to Jouni Smed et al. [25], the classical version of DM has three levels:

- **Strategic level** - decisions made over a long period of time and are based in a large amount of data. It's from speculative nature. The cost of the wrong scenario is very high.
- **Operational level** - is the most concrete and closely connected to the properties of the game world. The number of DMs in this level is very high, choosing short-term actions to give a set of alternatives.
- **Tactical level** - is the intermediary between strategic and operational levels decisions. Usually, use a group of entities and their co-operation. Tactical decisions aim to fulfil the plan made on the strategic level.

2.5.4 Synthesize players

According to Jouni Smed et al. [25], Synthetic players (SP) are the computer-controlled actors in a computer game. They affirm that the game is all about the conflict. So, these players can be used to simulate the behaviour of an opponent or an ally. SP is a NPC with limited participation and with special powers during the game, namely *deus ex machina* sometimes. Their complexity is related to the complexity and overture of the game, that means more complex is the game, more elaborate is the SP. The balance present between the model and the controller is tough to create with SP because when a rule is not present on the core structure, it is mandatorily present in the SP structure. For example, the human-player will hurt themselves during the game when surprised with an attack; an SP will know how to avoid the attack. SP can easily avoid future problems when the path for a trap is previously defined. Consequently, SP will be considered a cheater because of the fact he will receive extra resources or have outside knowledge.

Dataflow between the human player and SP must be similar to be possible to extrapolate human-like features into the SP. In order to reduce to the max the difference between the human-play and SP.

By Jouni Smed et al. [25], we can immediately recognise three distinct components involved in a game:

1. We have players who are willing to participate in the game (e.g. for enjoyment, diversion or amusement);
2. We must have rules which define the limits of the game;
3. There are goals which give rise to conflicts and rivalry among the players.

Between these three components, we have the following relationships. The players are willing to follow the rules of the game. The rules define the goals of the game. The goals motivate the players to participate in the game and drive the game forwards, and achieving a goal in the game gives a player enjoyment.

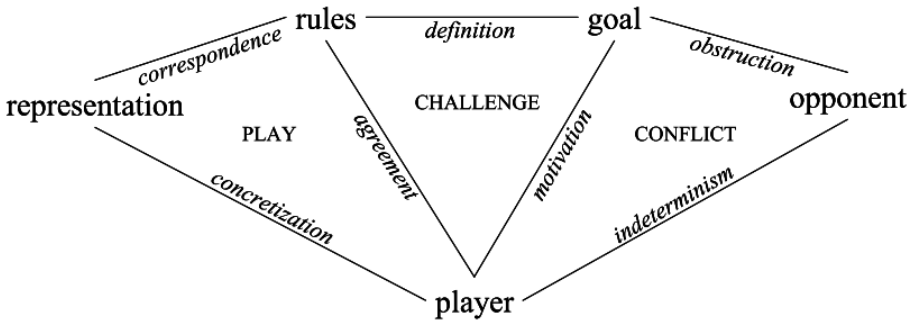

Figure 2.13: Components, relationships and aspects of a game

SP are based in a theory called MVC. Jouni Smed et al. [25] define this theory as the representation of the underlying application domain (model) should be separated from the way it is presented to the user (view) and from the way the user interacts with it (controller). Each part as their tasks well defined:

- **Model** - includes software components responsible for the coordination role (for example, evaluating the rules);
- **View** - handles the illustration role;
- **Controller** - contains the components for the participation role. Affects the model and maintains integrity.

2.5.4.1 Structure

Defined by Jouni Smed et al. [25], the world (or rather the synthetic view of the game world) consists of primitive events and states (phenomena) that are passed to pattern recognition and possibly stored for later use in a history buffer. The information abstracted from the current (and possibly the previous) phenomena is then forwarded to the decision-making system.

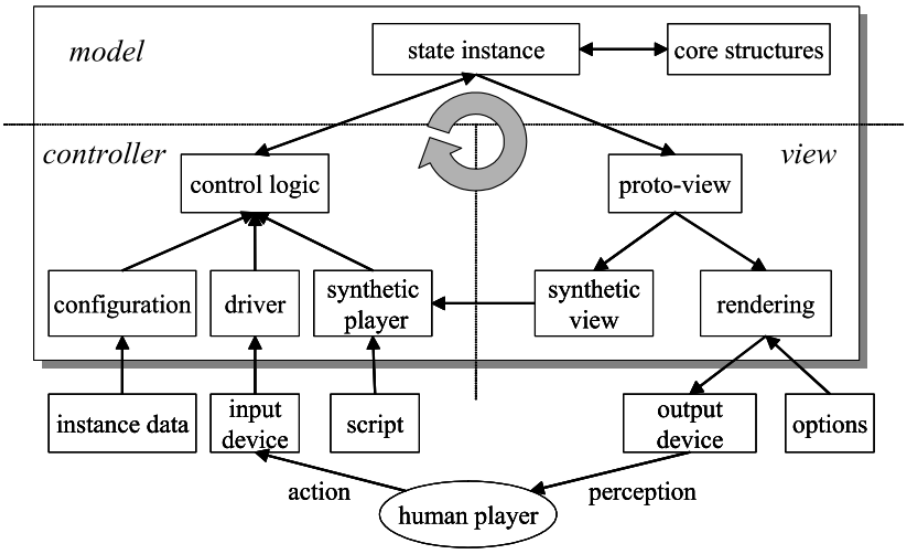

Figure 2.14: Components, relationships and aspects of a game

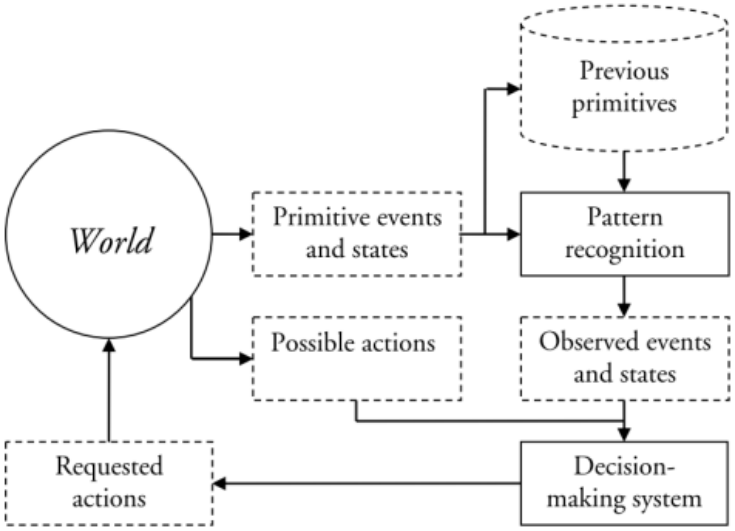

Figure 2.15: Structure by Jouni Smed et al. [25]

2.5.4.2 Behaviour

The environment around the human player is completely over thought to involve the human player, namely an anthropocentric climate. To simulate an SP, this agent needs to have some characteristics as humanness, stance and a life story.

Humanness

Humanness is the ability to approximate an SP to a human. Humans have some aspects, like personalities, psychological and physical characteristics. Their personality changes over the years, influenced by the external environment. These aspects are influenced by their flaws and are represented by emotions like fear, rage, hesitation, compassion, love, sadness and others. According to Jouni Smed et al. [25], we, as human beings, are quite apt to read humanness into the decisions

even when there is nothing but naive algorithms behind them.

Stance or Attitude

Humans always have an attitude or stand towards an opinion or situation. The role of an SP is to reproduce these attitudes from or against humans. The more common position for an SP is to play the enemy. In this role, the SP needs to be able to provide challenging behaviour and demonstrate some intelligence. When the SP plays the role of an ally, he needs to be able to adapt his conduct to the player's action. The SP can act as a neutral position in the game, like as an observer or a reporter. In this case, the SP's behaviour is dependent on the context and protocols from the role and the game. Last, the SP can be an NPC to develop and guide the plot, atmosphere, some extras and gameplay of the game.

Storytelling

Storytelling is the art of narrating a story to others and being able to explain the reasons for the actions taken. In a game, a narrative needs to be interactive and provide real-time events and feelings about free will. Humans tend to "humanise" the stories of the characters to understand their reasons to take some actions. Games are different from others aways from telling a story. In a movie, the storytelling has a restricted and severe way of telling the facts because they are previously determined in contrast to the games. According to Jouni Smed et al. [25], exist some challenges on interactive stories:

1. plot and three-dimensional characters are not enough to produce a high-quality narrative: there must be themes (e.g. betrayal, self-deception, love or revenge) behind them;

2. something is needed to make sure the story stays dramatically interesting;

3. apart from being robust and autonomous, the characters (i.e. SP) have to be memorable personalities by themselves;

4. a character should understand the players – even to the point of inferring other characters' and players' beliefs based on its own beliefs.

2.5.5 MMO vs. MMORPG

An MMO game is a "Massively Multiplayer Online" game. To put it in simpler terms, an MMO is an online multiplayer game which a large number of people can play simultaneously. You don't play with or against just a handful of players, but thousands, sometimes even millions of them at the same time.

MMORPG stands for "Massively Multiplayer Online Role-Playing Game". Some might say that all MMORPGs are easily recognizable by their fantasy settings, but that's not at all the case. Though the majority of RPGs do take place in the realm of elves, orcs, and dragons, a lot of them choose different worlds to explore, like far-away planets or post-apocalyptic wastelands. The keen observers among you may have noticed the two types of games have very similar definitions, but not quite the same.

The difference between MMO and MMORPG is that all MMORPGs are MMOs, but not all MMOs are MMORPGs. MMORPG is by definition a role-playing game, while an MMO can be anything from a Battle Royale action title, a real-time strategy game, or even a new type of interactive experience that defies genres.

2.5.6 Related work

To prove their study about SP, Jouni Smed et al. [26] developed a study where they create a game based on official rules from IIHF of ice hockey. In order to simplify the implementation, the rules about penalties were omitted. Some of the measurements were:

- **Face-offs** - is used to begin every game, period and play. It occurs when an arbitrator drops the puck between the sticks of two opposing players.
- **Offsides** - if a player on the attacking team does not control the puck and is in the offensive zone when a different attacking player causes the puck to completely cross the blue line into the offensive zone until either the puck or all attacking players leave the offensive zone.
- **Icing** - is an infraction when a player shoots the puck over the centre red line and the opposing team's red goal line, in that order, and the puck remains untouched without scoring a goal.

The physical model implemented on the game engine obeys Newtonian particles physics on a 2D plane. The particles are represented by circles, arches and lines and have some features like mass, size, position, velocity and acceleration. RoboCup inspired this model.

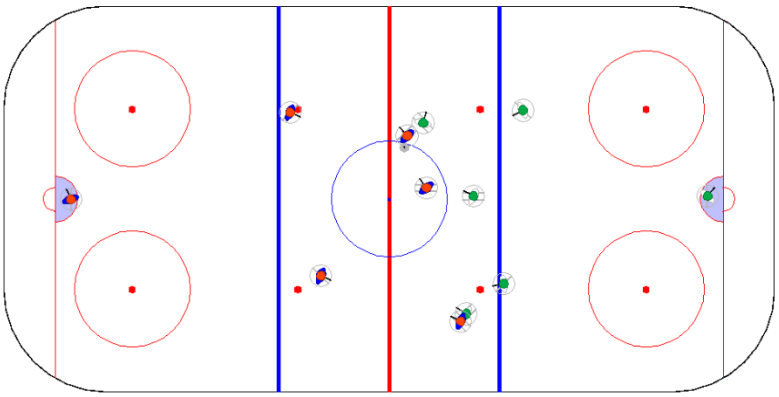

Figure 2.16: A screenshot from AIsHockey

The project (2.16) was implemented with java two platform. The SP was an instance of java class and inherited methods for receiving information on the game states and sending its actions (for example moving shooting and turning). The game engine runs on a server, which sends update messages to the clients regularly. To reduce network traffic, this is done by using multicasting. Each client has at least one SP and each SP runs in a different thread.

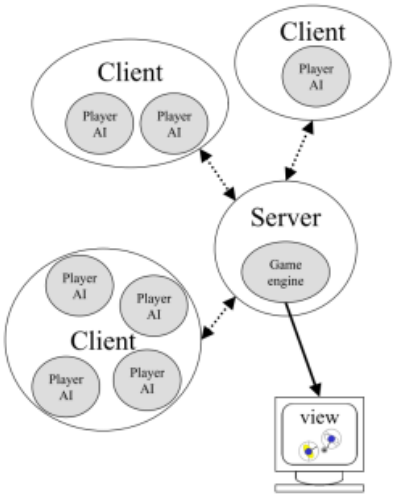

Figure 2.17: Synthetic players are distributed among the clients, while the game engine resides in the server.

With this work, the team wants to be able to study their theory about SP (real-time response, distribution, autonomy and communication) and understand how they can distribute the SP for several computers (cumulative computer power of network nodes). They try to answer this question: "how autonomous SP should be?". They defend that it is possible to rely on the network since the SP had the capability to drop and join. This technique will allow for creating smaller and better designs. In their words [26], SP must cross the gap of autonomy and communicate with each other and share their plans and negotiate between them as humans do.

AIsHockey was tested on graduated students. They had the opportunity to play alone and in groups. During the experience, the students could choose any basic approach they think was better to win. The researchers were very surprised because the students didn't copy each other. The testing group shows results with a high diversity and high quality. And, everyone was very alert to what the colleagues did. They concluded that the problem when they team up was the fact the team was a collection of personalities and opinions.

2.6 Using Synthetic players with Machine Learning

According to A. Drachen et al. [6], the best way to create an SP is using these data mining (DM) approaches:

- **Descriptive data mining** is used to describe the general properties of existing data in a concise way. In addition, it presents any interesting characteristics of the data without having a predefined target.
- **Predictive data mining** is used to forecast explicit value, based on patterns determined from known data. In other words, it is used to attempt to predict something based on inference in the data at hand.
- **Supervised learning** (SL) - originates in machine learning – a branch of artificial intelligence science that is concerned with the design and development of algorithms that allow computers to evolve behaviors based on data. A learning algorithm takes advantage of a test dataset, "training data" (observed examples), to capture characteristics of interest of the underlying, unknown probability distribution of data and make intelligent decisions based on their properties. In supervised learning, training data is combined with knowledge of desired outputs. The output of the algorithm can be a continuous value (regression) or a prediction of a class label of the input object (classification). The task of the supervised learning algorithm (the "learner") is to predict the value of the function for any valid input, after seeing a number of training examples (i.e. pair of input and target output). In order to achieve this ability, the algorithm has to generalize from the training data to unknown situations in a way that is reasonable. In the context of digital games, predictive data mining can be used to forecast when a player will stop playing, if a player will convert from a non-paying to a paying user, what types of items players will purchase, classify player behavior, etc.
- **Unsupervised learning** (UL) - also originates in machine learning, and also focuses in fitting a model to observations. However, unlike supervised learning, there is no "a priori" output. The input objects are generally treated as random variables, and a density model built for the dataset.

Some DM methods are:

- **Description** is when analysts are simply trying to describe patterns of trends in game data, and is usually accomplished using Exploratory Data Analysis (EDA), which is a graphical method for exploring data in search of trends or patterns. Descriptions of patterns often suggest possible explanations for them. EDA is particularly useful for basic analysis and for obtaining an understanding of the game data prior to the application of advanced algorithms.
- **Characterization** is simply the summation of general features of objects in a target class (or sample), producing a characteristic rule.

- **Discrimination** is when features of objects across two classes (or samples) is com- pared. Discrimination is identical to characterization except that discrimination results include comparative measures.

- **Classification** is used to organize data into classes, which is hugely useful to game development. Classification uses class labels to order objects in a data collection, normally using a training data set where all objects are already associated with known class labels (e.g. playtime per level associated with character class). The classification algorithm uses leanings from the test data and builds a model that can be applied to other or future data.

- **Estimation** is similar to classification, but the target variable is numerical, not categorical. In statistics, methods such as regression and correlation are estimation methods. In estimation, models are built using training data of complete records, which provide the value of the target variable as well as the predictors (causal variables). For new observations, estimates of the value of the target variable are made, based in the values of the predictors.

- **Prediction** is reminiscent of classification and estimation, but with prediction, we want to know about the future. The core idea is to use a large number of known values to predict possible future values. There are many approaches to prediction, from traditional statistical methods to more specialized knowledge discovery methods, such as neural networks, decision tree analysis, and k-nearest neighbors. Prediction is one of the most widely applied data mining methods in the analysis of data from multiplayer and massively multiplayer persistent games, where predicting the effect of design changes or the behavior of the player community, is important for revenue. Prediction can be used to forecast in many contexts around game development and publishing.

- **Clustering** is a lot like classification, in that the aim is to order data into classes. However, the class labels are unknown and it is up to the clustering algorithm to discover what the classes are and evaluate their acceptability. The core goal of clustering algorithms is to group or segment objects (e.g. players, asset, items, games or any observation, case or record) in such a way that the similarity between objects in one group (cluster) is high (intra-cluster similarity), while between groups is dissimilar (intercluster similarity).

- **Association (affinity)** when performing an association analysis, the goal is to find features (attributes) that “go together”, thus defining association rules in the data. An association rule specifies that if X, then Y. The association rule is accompanied by a measure of support, and of confidence. The support threshold identifies the frequency of the features occurring, and the confidence threshold defines the probability one appears when the other does.

2.6.1 Unsupervised learning

In the book "Game Analytics", A. Drachen et al. [6] explain that in UL there is no "a priori" defined output, i.e. we are not trying to predict target values, but rather focus on the intrinsic

structure of and relations in the data. Clustering is a classic example of UL. For classifying player behaviour, causes for game crashes.

2.6.1.1 Clustering

Continuing with the analysis of A. Drachen et al. [6] and apply to the context of behaviour analysis in computer game development, cluster (and classification) analysis provides a means for reducing the dimensionality of a dataset to find the most important features, and locate patterns which are expressed in terms of user behaviour as a function of these features, which can be acted upon to test and refine a game design (or specific) parameters of a design.

Some algorithms used to apply:

- **K-means clustering** - A cluster refers to a collection of data points aggregated together because of certain similarities.
- **Principal component analysis (PCA)** - is a technique for reducing the dimensionality of datasets, improving interpretability, simultaneously minimizing information loss. It is possible to create new uncorrelated variables that successively maximize the variance.
- **Non-negative matrix factorization (NMF or NNMF)** - is a group of algorithms in multivariate analysis and linear algebra where a matrix V is factorized into (typically) two matrices W and H, with the property that all three matrices have no negative elements.
- **Archetypal analysis (AA)** - according to Adele Cutler et al. [5], AA represents each individual in a data set as a mixture of individuals of pure type or archetypes. The archetypes themselves are restricted to being mixtures of the individuals in the data set. Archetypes are selected by minimizing the squared error in representing each individual as a mixture of archetypes. The usefulness of archetypal analysis is illustrated in several data sets. Computing the archetypes is a nonlinear least-squares problem, which is solved using an alternating minimization algorithm.

2.6.2 Supervised learning

According to A. Drachen et al. [6], SL methods for data mining are drawn from ML, a branch of artificial intelligence science that is concerned with the design and development of algorithms that allow computers to evolve behaviours based in data, notably with the purpose of prediction. Popular SL techniques include artificial neural networks, decision tree learning, support vector machines and Bayesian learning. The primary use of SL within games has been so far for the imitation of player behaviour, the analysis of player behaviour in online games, prediction of player behaviour on massively multiplayer online games, and notably for analysis of player behaviour towards driving revenue in online social games. A particular area of SL in games is to imitate human playing behaviour. Given a sufficiently large set of behavioural metrics data derived from players, SL can be used for both imitating human playing behaviour but also for predicting various aspects

of the behaviour. Essentially, SL can be used for the prediction of any player attribute given in the dataset. These kinds of questions are important to get answered during the development and testing of a game, and even after release. SL methods can thus be used to, e.g. test and adjust designs, or even form the basis for systems controlling real-time adjustment of game features during play.

2.6.2.1 Artificial Neural Networks

According to Sun-Chong Wang [27], an artificial neural network (or simply neural network) consists of an input layer of neurons (or nodes, units), one or two (or even three) hidden layers of neurons, and a final layer of output neurons. The figure below shows a typical architecture, where lines connecting neurons are also shown. Each connection is associated with a numeric number called *weight*. The output, h_i ,of neuron i in the hidden layer is (2.1), where $\sigma()$ is called activation (or transfer) function, N the number of input neurons, V_{ij} the weights, x_j inputs to the input neurons, and T_i^{hid} the threshold terms of the hidden neurons. The purpose of the activation function is, besides introducing nonlinearity into the neural network, to bound the value of the neuron so that divergent neurons do not paralyse the neural network.

$$h_i = \sigma(\sum_{j=1}^{N} V_{ij} * x_j + T_i^{hid}) \tag{2.1}$$

2.6.2.2 Decision Tree Learning

According to Lior Rokach et al. [20], a decision tree is a classifier expressed as a recursive partition of the instance space. The decision tree consists of nodes that form a rooted tree, meaning it is a directed tree with a node called "root" that has no incoming edges. All other nodes have exactly one incoming edge. Each leaf is assigned to one class representing the most appropriate target value. Alternatively, the leaf may hold a probability vector indicating the probability of the target attribute having a certain value. Instances are classified by navigating them from the root of the tree down to a leaf, according to the outcome of the tests along the path.

2.6.2.3 Support Vector Machines

According to Michael Doumpos et al. [8], Support Vector Machines (SVMs) have become an increasingly popular non-parametric methodology for developing classification models. In a dichotomous classification setting, the development of a classification model begins with a training sample $T = x_i, d_i, i = 1,2,...,n$, where $x_i \in R_m$ is the input information for the training object i on a set of m independent variables and $d_i \in \{1,+1\}$ is the corresponding outcome (dependent variable). The objective of the analysis is to construct a decision function $f(x) \rightarrow d$ that distinguishes the two classes. In the simplest case, $f(x)$ is defined by the hyperplane $xw = \gamma$ as equation 2.2, where w is the normal vector to the hyperplane and γ is a constant. Since f is invariant to any positive rescaling of the argument inside the sign function, the canonical hyperplane $d(xw\gamma)1$ is defined as separating the classes by a 'distance' of at least unity. The analysis of the generalisation

performance of the decision function *f(x)* has shown that the optimal decision function f is the one that maximises the margin induced in the separation of the classes, which is $2/||w||_2$.

$$f(x) = sgn(xgamma - \gamma) \tag{2.2}$$

2.7 Using Synthetic player for game testing

Methods based in UL and SL are used to study the evolution of the player's behaviour during and on the game. The experts apply these approaches in order to understand some questions like:

- How will the player react to the game?
- When will particular players drop the game? And why?
- What are the critical points of the game to the players? And why?
- Which players will convert from non-paying to paying?

In the book "Game Analytics" by A. Drachen et al. [6], they used the game "Tomb Raider: Underworld". This game has seven levels plus a prologue level. With this example, the team is trying to predict when the player would stop playing the game, based on their early play behaviour. This way, the experts identify the players who drop the game earlier and explain why this happens and find solutions to avoid players dropping the game. The analysed data was drawn from the native metrics suite of Square Enix Europe, which contains data from approximately 1.5 million players of Tomb Raider: Underworld. The population was divided into sub-samples. A sub-sample of 10 000 players was selected, randomly drawn from a larger sample of over 200 000 players from which the metrics data was captured within the period of 1st December 2008 - 1st January 2009. The original sample 10 000 players was cleaned thoroughly, removing instances where the metrics suite had missing data reported for a player. The aim of the analysis was to predict when the players drop the game, because they selected players who had completed level where included. The features used by them were selected from the core mechanisms with a strategy which helps ensure that the features are relevant to player behaviour analysis. Each features was measured for map or level, resulting on a total feature set up over 400 variables (number of features* level / map unit):

- **Playing time** : the time that each player spent playing the game. This includes a number of features, notably the playing time spent for each sub-segment of each level in the game (there are over 70 such segments).
- **Total number of deaths**: how many times the player died.
- **Help-on-Demand**: how many times the player requested help from the native Help-on-Demand system in the game, which assists players with their progress in the game.
- **Causes of death**: the game features various ways in which a player can die. These were classified into four groups:
  - Death by melee enemies;
  - Death by ranged enemies;

  - Death by environmental causes;
  - Death by falling (by far the most common cause of death in Tomb Raider: Underworld – 62.92%).

- **Adrenalin**: the number of times the adrenalin feature was used. Using adrenalin allows the player to temporarily slow down time while performing special attacks.
- **Rewards**: the number of rewards collected (the average is 112.08). Treasure: The number of treasures found. Each level has one or a few of these major finds, which take particular exploration to locate.
- **Setting changes**: players can change various parameters of the game, and four of these impact directly on gameplay and were therefore included: Ammo adjustment, enemy hit points, player hit points, and saving grab adjustment (which adjusts the time a player has to secure a handhold after a jump).

The sub-set was divided into three sub-datasets:

- players at least complete the first level;
- players at least complete the second level;
- players complete all the levels.

Several classification algorithms were used on the two problems: completion time and last level played. The best results were found using logistic regression, a relatively simple algorithm which could predict when a player would stop playing Tomb Raider: Underworld, with a success rate of 77.3%.

Chapter 3

Methodological Approach

3.1 Problem Formalization

This dissertation aims to create a simulation to allow the study of gambler player's behaviour. There are several types of gamblers, and these are influenced by the game, the environment, the music and other internal and external factors. On this work, we will focus on slots games and the major groups of gambler - the social and addictive gamblers. We will also study factors such as the will of play, the amount of wage, return to player, and the methods used to detect a gambler addiction.

3.2 Proposed Method

In order to be able to create the simulator, we need to know the existing types of players. Consequently, this work was divide into two different main stages. The proposed method followed has six different stages, all interconnected to each other. To explain the different steps, we will use the diagram 3.1 below.

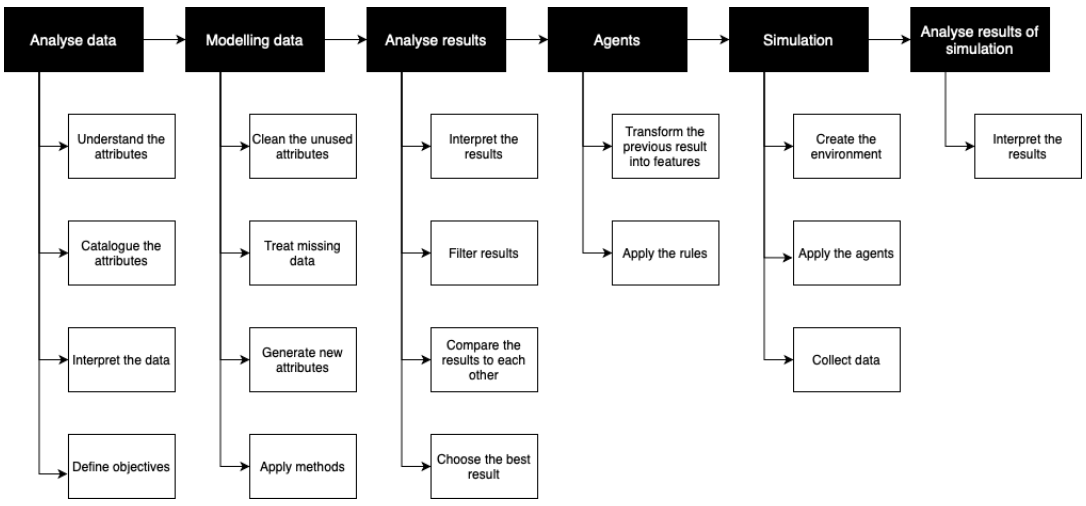

Figure 3.1: The block diagram.

The first three steps are related to the database and the last three to the simulator. To create a simulation, we had to gather real data from the field to understand what we are looking for, the type of data we are working on, and what kind of relations we are looking for. The first stage is to analyse the data. We need to understand the attributes available, catalogue them to reduce the possible doubts about the meaning of features, and interpret the data to define objectives for the following stages. The second stage is about modelling the data. In this stage, we need to guarantee that data can be modelled. So we have to correct the missing values, clean the unused attributes, find relations between the attributes to generate new features and apply different predictive models. The third phase is about analysis and interpretation of the results obtained from previous steps. This is the most critical stage of the proposed method. We have to be very cautious and critic towards results interpretation, to filter the most realistic ones and compare them until we have the best solution. As a fourth step, we have to create the simulator. The fourth step is about creating the agent's, players and game. In this stage, we need to transform the previous result into rules and features for the several types of players and define the answers from the game side. The fifth step is creating the environment, the game and players perspective and marks to collect the data results. Designing the environment means making the actions from players into something perceptible to the user's eyes and creating a real game in the background for data generation. The sixth and last step is to analyse several types of players' behaviour for the same game.

3.2.1 Rapidminer Studio

We used RapidMiner Studio, a visual data science workflow designer accelerating the prototyping validation of models to analyse the data set. Easy to use visual environment for building analytics processes: Graphical design environment makes it fast and straightforward to design better models. This tool allows data mining, text mining and predictive analytics. The program will enable us to enter raw data, including databases and text, which is then automatically and intelligently analysed in a large scale.

3.2.2 BDI Agent

Michael Bratman, an American philosopher, developed a human behaviour model based in intentions in practical reasoning, called belief-desire-intention (BDI) model. In the 1980s, when Standford starts the research in the area, they created a BDI architecture. The main idea of a BDI architectures concept lies on the though that we can talk about computer programs as if they have a "mental state". Then, we define the differences between beliefs, desires and intentions:

- **Beliefs** are information the agent has about the world. They represent environment characteristics, which are updated accordingly after the perception of each action. They can be seen as an informative component of the system.

- **Desire** is all the possible states of affairs that the agent might like to accomplish. It is a potential influencer of the agent's actions. We often think and talk of desires as being options

for an agent. They store the information about the goals to be achieved and properties and costs associated with each goal. They represent the motivational state of the system.

- **Intentions** are the states of affairs that the agent has decided to work until achieving. We can think of them as a goal. They are delegated to the agent, or may result from considering options: we think of an agent looking at its options and choosing between them. They represent the current action plan chosen. They capture the deliberative component of the system.

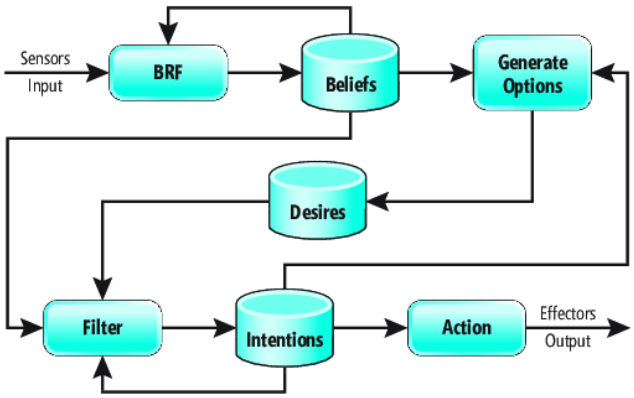

Figure 3.2: The BDI Agent Architecture from "Leveraging the Beliefs-Desires-Intentions Agent Architecture" by Microsoft

3.2.3 Eclipse Jason

The language interpreted by Jason is an extension of AgentSpeak, which is based on the BDI architecture. A component of the agent architecture is a belief base. An example of one of the things that the interpreter will constantly be doing, without it being explicitly programmed is to perceive the environment and update the belief base accordingly, even though this can be customised. Another critical component is the agent's goals, which are achieved by the execution of plans. The way reactive planning systems react to such events is by executing plans. They are courses of action that agents commit to executing to handle such events. The actions change the agent's environment, in such a way that we can expect the agent's goals to be achieved. The practical reasoning part of the agent's cyclic behaviour, in an AgentSpeak agent, is done according to the plans that the agent has in its plan library. Eclipse Jason is a plugging that allows us to work with Jason in an environment of Java.

3.3 Assessment Metrics

With this study, we are trying to respond questions such as:

- Is the theoretical RTP similar to the real one?
- Which are the types of gamblers?
- Which are the main differences of a social gambler and an addictive gambler?

The main components of a slot game include the reels, the pay lines and the paytable. The slot game has a predefined number and table of symbols. The symbol 100 represents the WILD, a symbol that assumes the value of the number is left. The symbol 200 represented the SCATTER, the symbol with the biggest prizes and counted on the matrix and not the line.

```
[0, 1, 2, 3, 4, 100, 200]
```

Listing 3.1: Set of symbols for the reels

The reels can represent by a vector with combinations of these symbols. The more often games vary between three and five reels.

```
[
    [0, 1, 0, 2, 2, 4, 1, 1, 4, 0, 2, 1, 1, 4, 200, 0, 3, 1, 2, 1],
    [2, 1, 0, 1, 1, 2, 3, 4, 0, 100, 2, 0, 3, 0, 3, 0, 200, 1, 1, 1],
    [4, 3, 2, 200, 3, 3, 4, 2, 3, 4, 1, 100, 0, 0, 2, 3, 0, 2, 2, 4],
    [2, 1, 2, 0, 4, 0, 4, 1, 100, 0, 0, 1, 3, 2, 200, 0, 0, 1, 0, 0],
    [2, 2, 3, 100, 2, 3, 0, 4, 3, 200, 1, 0, 200, 2, 2, 4, 3, 3, 2, 3]
]
```

Listing 3.2: The reels from the slot game (3 rows for five columns).

The pay lines are all independent between each other. They always are counted from the left to the right and from the first position until the end. The paytable is a predefined table with the counter of symbols with a prize associated.

Symbols\N of symbols	1	2	3	4	5
0	0	0	5	20	50
1	0	0	5	20	50
2	0	0	10	50	100
3	0	0	15	75	150
4	0	0	25	150	400
200	0	0	1	3	5

Table 3.1: Prize table

Return to player (RTP) is an international measure to explain the public how much will be the return for the game to a player. On a games with an RTP equal to 79% this means in 100 credits the player will receive 79 credits. The RTP of a game is dependent on the number of reels (n), the number of different symbols (s) present on the game, the number of lines with a prize and the number of WILD. To calculate an RTP from a slot game, we need to calculate the probability of a common symbol (s) gets out of on a reel with n positions. If the game has the WILD symbols, we need to sum the number of wilds to the number of common number symbol.

$$P_s(n) = \frac{\#Symb_s(n) + \#Symb_{100}}{\#Total(n)} \tag{3.1}$$

To calculate the probability of a SCATTER get out, we need to count the number of rows multiply for the number of times of a SCATTER on the reel and divide for the total number of symbols on a reel.

$$P_{200}(n) = \frac{\#row \cdot \#Symb_{200}(n)}{\#Total(n)} \tag{3.2}$$

The contrary of a probability from a certain symbol is one less the probability of that symbol.

$$\neg P_s(n) = \frac{\#row \cdot \#Symb_{200}(n)}{\#Total(n)} \tag{3.3}$$

After these two steps, we need to estimate the RTP by line. This probability can be calculated doing the product from the prize of 3 equals symbols multiplied by chance to get the same symbol on the first, second and third column and the contrary of getting out the same symbol on the fourth column; plus the prize of 4 equals symbols multiplied by chance to get the same symbol on the first, second, third and fourth column and the contrary of getting out the same symbol on the fifth column; plus the prize of 5 equals symbols multiplied by chance to get the same symbol on the first, second, third, fourth and fifth column. This formula demands to be applied to each row for each symbol.

$$\begin{aligned} RTP_{line} = \sum (&\$_{Symb_s}(3) \cdot P_{Symb_s}[0] \cdot P_{Symb_s}[1] \cdot P_{Symb_s}[2] \cdot (1 - P_{Symb_s}[3]) + \\ &+ \$_{Symb_s}(4) \cdot P_{Symb_s}[0] \cdot P_{Symb_s}[1] \cdot P_{Symb_s}[2] \cdot P_{Symb_s}[3] \cdot (1 - P_{Symb_s}[4]) \\ &+ \$_{Symb_s}(5) \cdot P_{Symb_s}[0] \cdot P_{Symb_s}[1] \cdot P_{Symb_s}[2] \cdot P_{Symb_s}[3] \cdot P_{Symb_s}[4]) \end{aligned} \tag{3.4}$$

Following, we need to estimate the SCATTER's RTP by line. This formula is very similar to the previous but applied to the SCATTER symbol. This probability can be calculated doing the product from the prize of 5 equals SCATTERS multiplied by chance to get the same symbol on all the columns; plus the prize of 4 equals SCATTERS multiplied by the chance of not getting the same symbol on the fifth column and to get the same symbol on first, second, third and fourth column; plus the prize of 3 equals SCATTERS multiplied by the chance of not getting the same symbol on the fifth and fourth column and to get the same symbol on first, second and third

column.

$$RTP_{SCATTER} = \sum(\$_{Symb_{200}}(5) \cdot P_{Symb_{200}}[0] \cdot P_{Symb_{200}}[1] \cdot P_{Symb_{200}}[2] \cdot P_{Symb_{200}}[3] \cdot P_{Symb_{200}}[4] + \\ + \$_{Symb_{200}}(4) \cdot \sum_{c=0}^{4}((1 - P_{Symb_{200}}[c]) \cdot \prod_{k=0;c\neq k}^{4} P_{Symb_{200}}[k]) + \\ + \$_{Symb_{200}}(3) \cdot \sum_{c_1=0}^{3}\sum_{c_2=c_1+1}^{4}((1 - P_{Symb_{200}}[c_1]) \cdot (1 - P_{Symb_{200}}[c_2]) \cdot \prod_{k=0;k\neq c_1;k\neq c_2}^{4} P_{Symb_{200}}[k])) \quad (3.5)$$

The total RTP are determined by the sum of the RTP byline and the SCATTER's RTP by line.

$$RTP_{Total} = RTP_{line} + RTP_{SCATTER} \quad (3.6)$$

3.4 Dataset Description

During this work, the dataset used to find each gambler's main characteristics belongs to *bwin.party* and was collected between November 2008 and November 2009. This dataset contains all the data relating to all the users who triggered a responsible gambling case (RGC) and/or a responsible gambling intervention (RGI). RGC (A.1) means the cases of users triggered the flag but don't need any intervention from *bwin.party*. RGI (A) means the cases of users need to receive intervention from the *bwin.party* team and they were reported to the Division on Addiction with headquarters in Vienna, Austria. The analytic dataset includes betting behaviour for two sports betting products (fixed odds and live-action), a combination of five casino-type products, five poker- type products, and twelve other games (A.3). The authors give us an alert to be careful with the data related to the number of bets and the missing values on the raw dataset II on the codebook [22] because these values belong to vendors. By this reason, we need to exclude all the games present on the original dataset marked with a star (*). Four different tables compose the dataset:

- **Raw dataset I** - contains all the information about demographics like User ID, a variable indicating group case versus control, country of residence, preferred language for interacting with bwin.party, gender, year of birth, bwin.party registration date, and bwin.party first deposit date[22];

- **Raw dataset II** - contains all the information about daily aggregates like user ID, date of transaction, product type, turnover or amount staked for given day/product, hold or total amount lost for given day/product, and the number of bets for given day/product. All transactions are in Euros[22];

- **Raw dataset III** - contains all the details about responsible gambling (RG) like user ID, number of RG events experienced, First RG date, Last RG date, First RG Event type and intervention type[22];

- **Analytic dataset** - contains gambling behaviour data that was available for fixed odds, live-action, casino-type, poker type and other games betting for each participant included in the Daily Aggregates raw dataset aggregated over the entire history of betting[22].

Chapter 4

Result and Analysis

4.1 Dataset

For the dataset analysis, it was used the tool Rapidminer, explained in chapter 3. As explained before, there were some standard tasks to do such as cataloging the attributes from all the tables available in the dataset (Raw I, II, II and analytics), handling the missing values, cleaning the unused attributes, and testing several models. The catalogue with all the attributes from the original and final tables is available in the appendix A.

The original dataset was a result of online gambling at the European bwin.party webiste, during the years of 2008 and 2009. The Raw I contains information from the user such as the year of birth, userID, gender, registration date and first deposit date. Some data such as the attributes year of birth and registration date were missing. For the year of birth, it was used the average age from all the users years to fill the blank spaces. It was decided that the first deposit date was equal to the registration day's date for the registration date's empty spaces. The attributes related to the country name and language name were removed as they were not adding relevant information for the research. Finally, two other attributes were considered, namely user's age, evaluated as the time elapsed between the user's year of birth until 2009, and the number of days between the registration date and first deposit date.

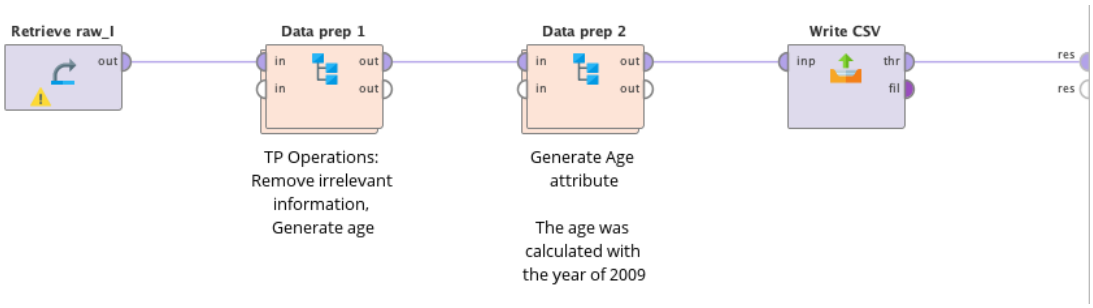

Figure 4.1: Rapidminer excert

```
    age = 2009-[YearofBirth]
    days_passed_registration_1deposit = date_diff([Registration_date],[
        First_Deposit_Date])*(1.15740741*10^(-8))
```

Listing 4.1: Formulas to calculate the age and the days between the registration date and the first deposit date

Raw II contains betting information associated with each product. In this table, the data was organised by userID and contains date transaction, product type, turnover, hold and number of bets. There was information missing in the attributes such as turnover and hold. The authors of the dataset explained that they needed to hide the information related to the vendors. Consequently, we had to select the types of games that were not associated with vendors. A consequence of this was the number of data available which was decreased significantly.

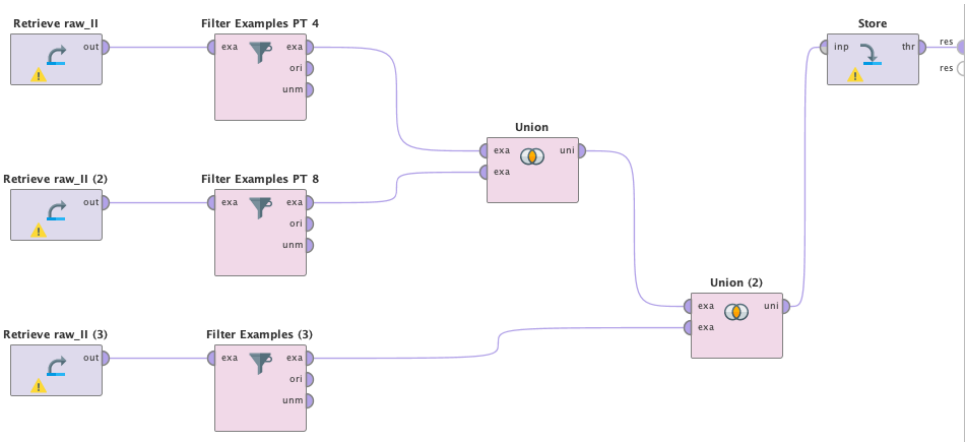

Figure 4.2: Rapidminer excert

Raw III contains all the information from RGC and RGI. In other words, this table has a summary of the types of RGC and RGI of each user with the respective dates of cases and interventions. It was missing just a few values such as RGC first date, RGC last date and intervention type. As this was not enough to interfere with the results, we decided to filter out all the missing value. After this, we changed the labels of RGC and RGI for the correspondent ID's and create a new feature to store the number of days between the first and last RGC.

```
if([Event\_type\_first]=="3rd person contact", 1,
  if([Event\_type\_first]=="account closure/reopening", 2,
    if([Event\_type\_first]=="cancel outpayment", 3,
      if([Event\_type\_first]=="change of limit: manual change", 4,
        if([Event\_type\_first]=="fair play - heavy complainer", 6,
          if([Event\_type\_first]=="inpayment block request", 7,
            if([Event\_type\_first]=="minor case", 8,
              if([Event\_type\_first]=="partial block", 9,
                if([Event\_type\_first]=="problem reported", 10,
```

```
10                  if([Event\_type\_first]=="user request higher limit", 11,
11                    if([Event\_type\_first]=="two events on the same day", 12, 13 )
                          ))))))))))
```

Listing 4.2: Formula to change the RGC label to ID

```
1  if([Interventiontype_first]=="bwin.party gives additional RG advice to subscriber."
       , 1,
2    if([Interventiontype_first]=="Subscriber's account is re-opened upon customer
         request after closure or self-exclusion period had elapsed.", 2,
3      if([Interventiontype_first]=="The customer's request was technically not
           possible.", 3,
4        if([Interventiontype_first]=="Account blocking for investigation and asking
             the customer for a detailed statement about his situation.", 4,
5          if([Interventiontype_first]=="Change of personal deposit limit from 'VIP'
               subscriber accepted.", 5,
6            if([Interventiontype_first]=="Request for partially blocked account block
                 could not be conducted completely. ", 6,
7              if([Interventiontype_first]=="The third party who has made a request is
                   advised about privacy regulation; the customer is contacted and
                   asked for a statement; the customer account remains open until
                   further evidence indicates confirms suspections.", 7,
8                if([Interventiontype_first]=="Request for partially blocked account
                     block could be conducted completely.", 8,
9                  if([Interventiontype_first]=="Inpayment method not blocked as
                       requested.", 9,
10                   if([Interventiontype_first]=="Inpayment method blocked", 10,
11                     if([Interventiontype_first]=="Request for higher personal
                           deposit limit denied.", 11,
12                       if([Interventiontype_first]=="Request for higher personal
                             deposit limit accepted.", 12,
13                         if([Interventiontype_first]=="Daily/weekly personal deposit
                               limit is changed.", 13,
14                           if([Interventiontype_first] == "Imposed account block by
                                 bwin.", 14,
15                             if([Interventiontype_first]=="Betting limit changed.",
                                   15,
16                               if([Interventiontype_first]=="Account remains closed.
                                     ", 16,
17                             if([Interventiontype_first]=="Account is blocked and
                                   reimbursement is given.", 17, 18))))))))))))))))))
```

Listing 4.3: Formula to change the RGI description to ID

After cleaning and treating the first three tables, we started to join Raw I and III to create the "PlayerType" table and find the user with a flag on RGC or RGI. Then, we analyse all the RGC events and RGI cases to find the flags referring to gambling cases and classify the player type

as a "Social" gambler and and "Addictive" gambler. These terms are used to refer to the main categories of players in gambling research thus creating the label for classification. We chose to eliminate the attributes date of the first deposit, registration date, RGC date and RGI date as these pieces of information were more relevant represented by the time gap than the specific date itself.

```
if(
  [RG\_event\_type\_first\_ID] == 1 ||
  [RG\_event\_type\_first\_ID] == 2 ||
  [RG\_event\_type\_first\_ID] == 9 ||
  [RG\_event\_type\_first\_ID] == 12 ||
  [ID\_interventionType\_first] == 1 ||
  [ID\_interventionType\_first] == 4 ||
  [ID\_interventionType\_first] == 8 ||
  [ID\_interventionType\_first] == 14 ||
  [ID\_interventionType\_first] == 16,
  "Gambler",
  "Social"
)
```

Listing 4.4: Formula to classification of the gambler

Raw analytics contains all the generated data from the previous table. They developed attributes such as the first active game of each type of product, first active product in 31 days for every kind of product, the total stakes, bets and active days in 31 days, average and size of betting by-product on active days and per day, total stakes and bets for each week and weekend and other statistical metrics. Raw analytics has plenty of attributes related to the missing data in Raw II whereas we decided to filter all the features associated with all the types of products except for the casino type and other games.

After all the tables' experiences, we concluded that the Raw II was not helpful because all the information we can extract from there was in the Raw Analytic. The final step was to join the tables PlayerType and Raw Analytics and eliminate all the attributes related to a specific game or product. We were only keeping the general features concerning to total about all the products. The eliminated attributes were the UserID, the first game played, all the frequencies week and weekend features, and the most frequent game related to casino and other products. It was decided to change some names to make them more explicit for reading and analysing. Some examples are NumberofGames31days to totalNumberOfGames, pcsumstake31days to totalWagered and pcavgbetsize to avgBetSize.

After analysing all the predictive models available on RapidMiner, and watching some of the Rapidminer Academy videos to understand some of the best ways to model the problem, it was concluded that the most reliable method to resolve the modelling problem was using the cross-validation vote and boosting. Since we used two methods, it was used the operator multiply applied to vote and boosting. This technique was the most trustworthy as it returns the results ordered from the best to the worst outcome possible. The final result was the same decision tree

on both methods. We can conclude that the essential attributes to define a Social and Addictive Gambler were related to the type of events and interventions they had with bwin.party Customer Services. In conclusion, we can define some rules simplified for the different players based on the events and intervention of bwin.party. The Addictive gambler has two opportunities to ask for a bank loan because the max of events a gambler can have on bwin.party before suffering an intervention are also two events. Often, the events and the intervention are related to wallets limits. The rules we can apply to Social Gambler weren't possible for implementation because they usually are related to system failures or age problems. For simplification, it was decided that all the variable related to money will be measured in credits and we found out that the max value for the wallet is 255 credits. The limit of credits value was calculated through the total wagered max.

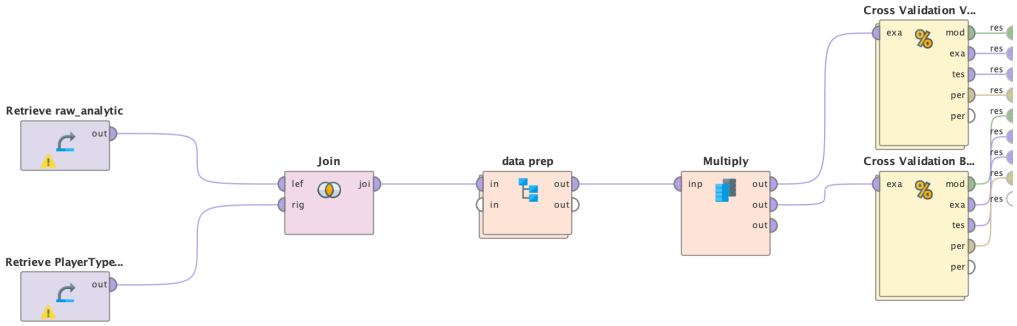

Figure 4.3: Main process for cross validation vote and boosting

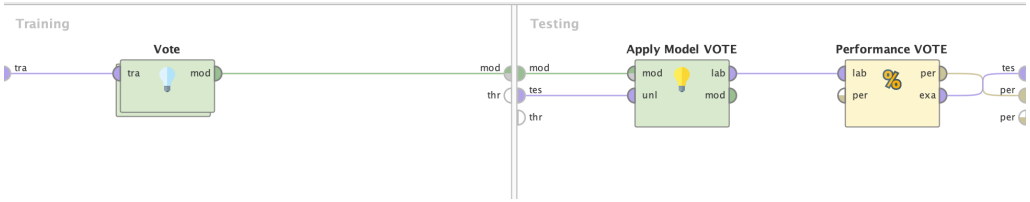

Figure 4.4: Excerpt from Cross Validation Vote

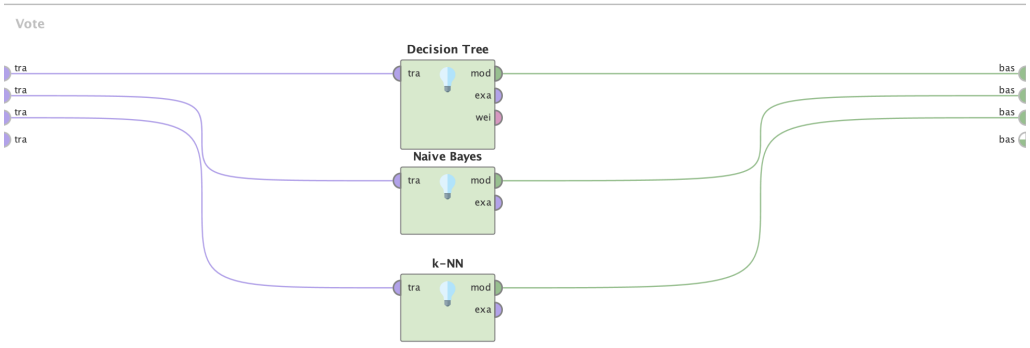

Figure 4.5: Inside of Cross Validation Vote

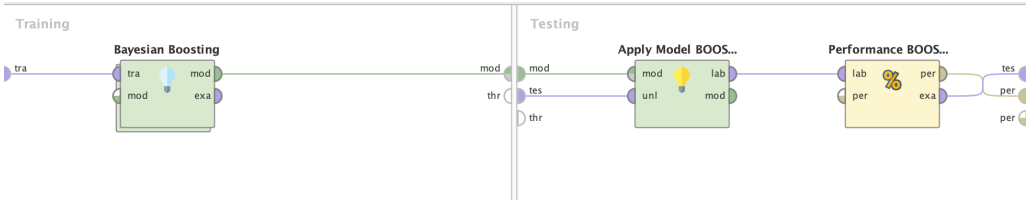

Figure 4.6: Excerpt from Cross Validation Boosting

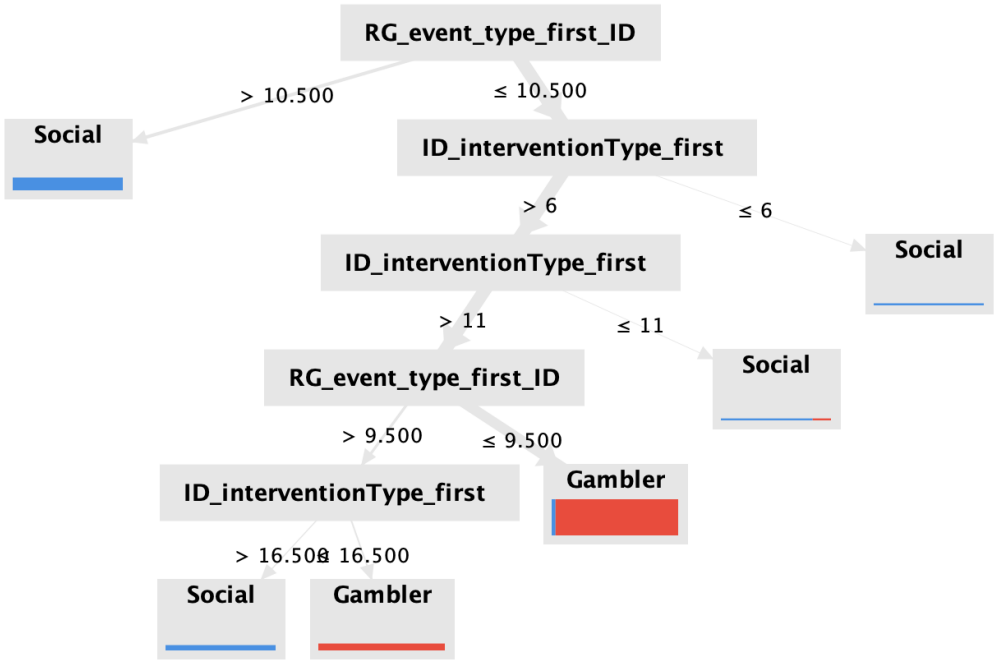

Figure 4.7: Decision Tree

4.2 Simulation

The simulation is composed of two main groups, the agents and the environment. We started to create the agents, following the slot game and finally, the casino environment.

4.2.1 The agents

We have three main types of agents, the game, the bank and the players.

The game agent is an agent responsible for responding to the action of the player agent. It is responsible for receiving the money, making a bet while the player had money and informing the player agent how much he has lost or won and when he hasn't more money.

The player agent could be of two different types: Social Gambler or Addictive Gambler. In gambling studies coordinated by psychologists, they talk about a variable called "Will to Play". This variable represents the intention of the player to continue playing.

The Social Gambler represents the type of player play just and only as entertainment, pleasure, and social context. This type of player controls the emotions and only spends the amount of money

he has on that specific time. This player lost his intention to play when started to lost more than he wins or in the case he wins a jackpot or a big prize to cover at least the initial investment. He keeps and increases his will to play when he is winning.

The Addictive Gambler represents the player who has compulsive gambling or gambling addiction. This type of player desires to continue to gamble without taking any attention to another part of their lives. They are constantly searching for more money to bet, thinking of winning, and can't stop playing. He is no control of his emotions or desires. Even when he is losing a huge amount of money, he will have an even deeper desire for playing. As explained before, this type of player has the opportunity to use the bank two times. When he does that, they feel guilty and sad, and these emotions had a good impact in his will to play (decrease the intention). Instead, he will continue to play until he hasn't more money and can't asking for another loan.

The bank agent is the agent responsible for representing the bank where the addictive gambler asks for a loan. This agent calculates a random amount of credits between three and 255 after checking if the addictive gambler can receive the loan. The min value represents the bet size, and the max was calculated before through data analysis.

4.2.2 Slot Game

To make this simulation even more real, we created a slot game based in information available in chapter 3. The slot has three lines for five columns. It means we have three lines with a prize and all the lines are independent. The columns represent the reel in the previous chapter. The first line ([0][0], [0][1], [0][2]. [0][3], [0][4]) represents the random positions we will calculate from a zero to nineteen. The next line ([1][0], [1][1], [1][2], [1][3], [1][4]) is calculated by add one to the position before. And, the last line ([2][0], [2][1], [2][2]. [2][3], [2][4]) is calculated by add two for the first-line positions. The reels are always circular, that means when we are on the nineteenth position, the next position will be the first one.

```
1 public int randomPos() {
2     int pos = 0;
3     int min = 0;
4     int max = 19;
5
6     pos = (int) (Math.random() * (max - min + 1) + min);
7
8     return pos;
9   }
```

Listing 4.5: Code for determine the random position

After calculating the positions from the reels, we needed to create the final matrix to represent the slot game. So, we built a matrix with the results from the positions matrix.

```
public void createPosSpin() {
    for (int i = 0; i < 1; i++) {
      for (int j = 0; j <  spinPos[i].length; j++) {
        spinPos[i][j] = randomPos();
        if( spinPos[i][j] == 19) {
          spinPos[i+1][j] = 0;
          spinPos[i+2][j] =  spinPos[i+1][j] + 1;
        } else if( spinPos[i][j] == 18){
          spinPos[i+1][j] =  spinPos[i][j] + 1;
          spinPos[i+2][j] = 0;
        } else {
          spinPos[i+1][j] =  spinPos[i][j] + 1;
          spinPos[i+2][j] =  spinPos[i][j] + 2;
        }
      }
    }
}
```

Listing 4.6: Code to assign the random position

Before we can determine the prize present in the matrix game, we needed to search for the WILD symbol represented by 100. The WILD symbol will be substituted by the symbol present in the previous position.

```
    int aux = row.length;
    for (int i = 0; i < row.length; i++) {
      if(row[i] == 100 && i != 0) {
        row[i] = row[i-1];
      }
    }
```

Listing 4.7: Excert of the search for the WILD symbol

For a line to have a prize, the count of equal symbols must start in the first position and need to have at least three equal symbols. This cycle only has to run four times to make all the required comparisons. We started to define the token as "-1" as this symbol doesn't exist on the reels. The count is equal to "1" because we are counting the symbols and not the comparisons.

```
    aux = row.length-1;
    for (int i = 0; i < aux; i++) {
      if(row[i] == row[i+1]) {
        count++;
        token = row[i];
      }else {
        if(count < 3) {
          count = 1;
```

```
9             token = -1;
10          }
11          break;
12        }
13      }
```

Listing 4.8: Excert of the search of the prize in game matrix

For the SCATTER symbol represented by the 200, we needed to count in the matrix how many times it appears and multiply it by the number of rows of the matrix.

4.2.3 The environment

Being a casino game simulation, the environment must contain the actions a person experience on a casino's ground. With this in mind, we tried to simulate some actions such going to the wallet to check the existence of money there and if it is true, do the cash in in the game machine and start to play while we have money. In case there is no balance available in the wallet, we need to simulate going to the bank and asking for a loan and go back to play until we have money left.

There were designed two different environments, one for each type of player. The only difference between them was the existence of the bank. As Social gambler doesn't need to do bank loans, we removed this part from his environment.

For the Social Gambler environment, on the top of the left corner, we have the wallet's position and the representation of his will to play. On the bottom of the right corner, we have the game position.

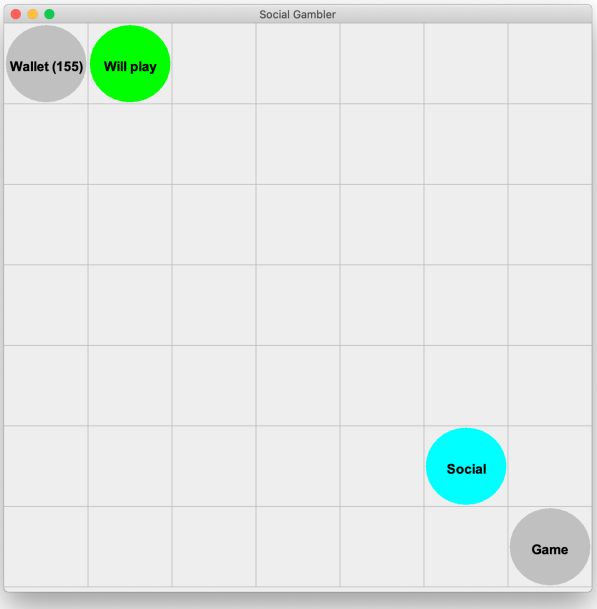

Figure 4.8: Social Gambler environment

For the Addictive Gambler environment, the wallet, will to play and game are in the same position as Social Gambler. On the bottom of the left corner, we can find the bank position.

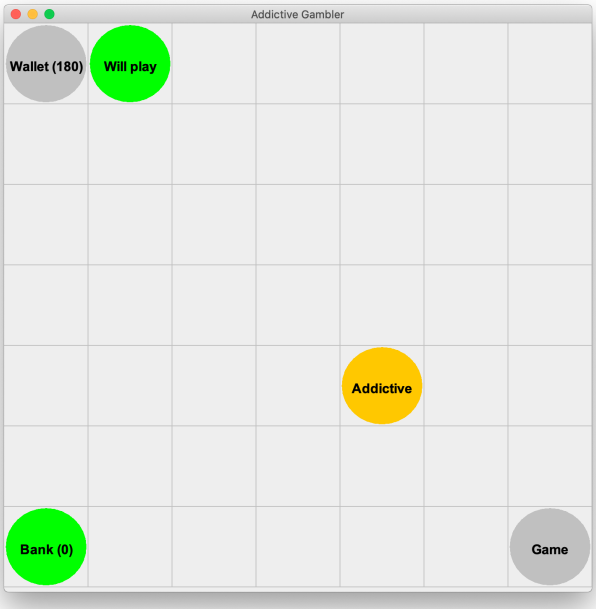

Figure 4.9: Addictive Gambler environment

The number between parentheses on the wallet represents the number of credits available, and it's on continually update during the game. The number between parentheses on the game represents the number of times the player can play depending on the available credits (available credits/denomination). The number between parentheses on the bank represents the number of times that addictive gambler receives a loan. When the game starts, each player's will (social or addictive) is characterised by 0.5. For the Social Gambler, each victory represents an increment of 0.001, each lost represents a decrease of 0.002 and winning a jackpot represents lost of will to play. When the will to play on Social Gambler is bigger than 0.35, the representation is green. If Social Gambler wins a jackpot, the representation is pink. If Social Gambler loses his will to play, the representation is red. For the Addictive Gambler, each victory represents an increment of 0.001, winning a jackpot means an increment of 0.01 and each time they ask for a loan has a cost of 0.15 on his will to play. When the will to play of the Addictive Gambler is less than 0.65, he controls his will. Otherwise, he is completely addicted to the game.

The only agent moving through the environment is the player agent. If the player agent represents the Social Gambler, he starts in the centre of the environment, goes to the wallet to get the money and then go to the game machine while he had the intention to play or had credits available.

If the player agent represents the Addictive Gambler, he starts in the centre of the environment, goes to the wallet to get the money then go to the game machine while he had credits available. When he doesn't have enough credits, he goes to the bank and asks for a loan. When the loan is rejected, the Addictive Gambler return to the wallet.

4.3 Results

First of all, we needed to calculate the RTP of our slot, applying the formulas explained in the previous chapter. We used Excel to perform the calculation of the probabilities of each symbol on the reels. We introduced the reel's table in excel and created a new one with the results of the formula 3.1 and 3.2.

POS/REELS	Reels 1	Reels 2	Reels 3	Reels 4	Reels 5
1	0	2	4	2	2
2	1	1	3	1	2
3	0	0	2	2	3
4	2	1	200	0	100
5	2	1	3	4	2
6	4	2	3	0	3
7	1	3	4	4	0
8	1	4	2	1	4
9	4	0	3	100	3
10	0	100	4	0	200
11	2	2	1	0	1
12	1	0	100	1	0
13	1	3	0	3	200
14	4	0	0	2	2
15	200	3	2	200	2
16	0	0	3	0	4
17	3	200	0	0	3
18	1	1	2	1	3
19	2	1	2	0	2
20	1	1	4	0	3

Table 4.1: Reel's table

```
1    (CONTAR.SE(B2:B21;0)+CONTAR.SE(B2:B21;100))/CONTAR(B2:B21)
```

Listing 4.9: Example of code in excel

	P(0)	P(1)	P(2)	P(3)	P(4)	P(200)
Reel 1	0,2	0,35	0,2	0,05	0,15	0,15
Reel 2	0,3	0,35	0,2	0,2	0,1	0,15
Reel 3	0,2	0,1	0,3	0,3	0,25	0,15
Reel 4	0,45	0,25	0,2	0,1	0,15	0,15
Reel 5	0,15	0,1	0,35	0,35	0,15	0,3

Table 4.2: Results of the probability of each symbol on each reel

As a next step, we needed to calculate the negation of the previous probability. We only needed to apply this formula for the fourth and fifth position because they are the only two positions where we can not have the same symbol.

```
1 1-J10
```

Listing 4.10: Example of code

	∼P(0)	∼P(1)	∼P(2)	∼P(3)	∼P(4)	∼P(200)
Reel 4	0,55	0,75	0,8	0,9	0,85	0,85
Reel 5	0,85	0,9	0,65	0,65	0,85	0,7

Table 4.3: Results from the contrary of the previous probability

The last step is to determine the RTP byline for each symbol. We created a table where the lines represent the RTP for each symbol, and the columns represent the possible combination. We applied the formulas 3.4, 3.5 and 3.6.

Symb	1	2	3	4	5
0	0	0	5	20	50
1	0	0	5	20	50
2	0	0	10	50	100
3	0	0	15	75	150
4	0	0	25	150	400
200	0	0	1	3	5

Table 4.4: Prize table

```
1 1-J10
```

Listing 4.11: Example of code

	3 Symb	4 Symb	5 Symb	Total
RTP LINHA[0]	0,03300000	0,09180000	0,04050000	0,16530000
RTP LINHA[1]	0,04593750	0,05512500	0,01531250	0,11637500
RTP LINHA[2]	0,09600000	0,07800000	0,08400000	0,25800000
RTP LINHA[3]	0,04050000	0,01462500	0,01575000	0,07087500
RTP LINHA[4]	0,07968750	0,07171875	0,03375000	0,18515625
RTP LINHA scatter	0,00200813	0,00129094	0,00075938	0,00405844

Table 4.5: Results to calculte the RTP byline

RTP	0,79976469
RTP Social Gambler	0,86125474
RTP Addictive Gambler	0,869686088

Table 4.6: Results of RTP, RTP Social Gambler and RTP Addictive Gambler

The difference between the total RTP determined from the slot and the total RTP obtained from the simulator happens because the amount of data collected in the simulator hasn't enough.

After this, we analysed the results from the simulation itself. we collected the following information by each play for each player:

- Type;
- Player;
- Denomination
- The initial amount of money;
- Wagered credits;
- The last prize;
- Available credits after the spin;
- Number of bets;
- Will to play;
- Number of loans (only for addictive gambler);
- the amount loaned (only for addictive gambler).

We collected a total of 13 377 results from the simulation. What we could observe is that, even though we had more social gambler players, the addictive gamblers play much more.

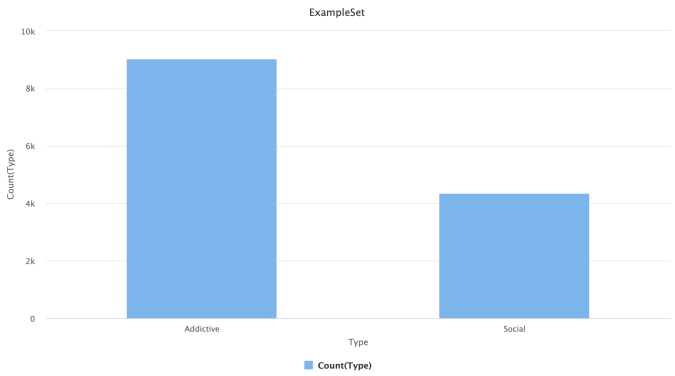

Figure 4.10: Plays of each type

To count the number of wins and lost games, we define that zero will represent the lost games, and one will represent the wins games. The number of games lost or won by an addictive gambler is almost double the number of social players.

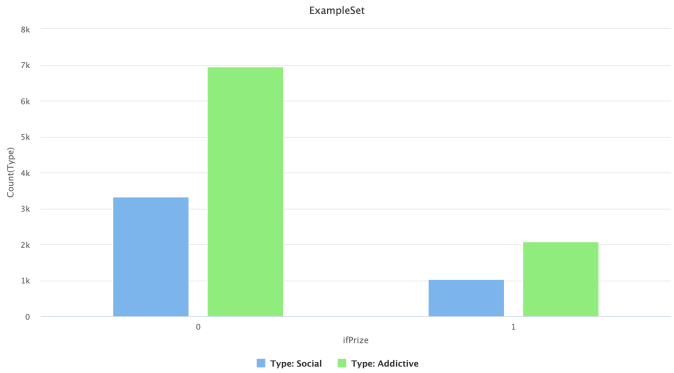

Figure 4.11: Winning and loses

We can observe a social gambler's will who plays until they had no money to wager in this graphic.

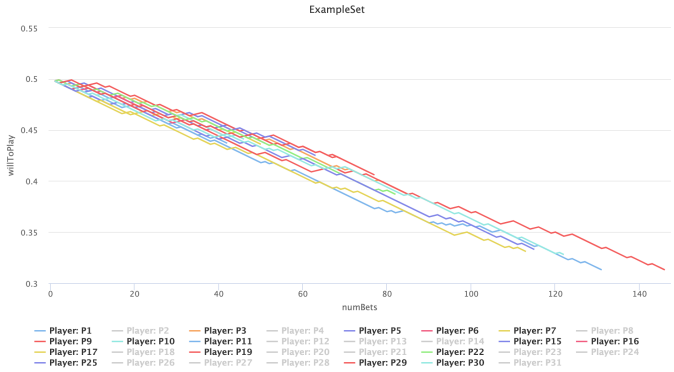

Figure 4.12: Will To Play Until No Money Social

We can observe a social gambler's will who plays until they have won a jackpot or a big prize.

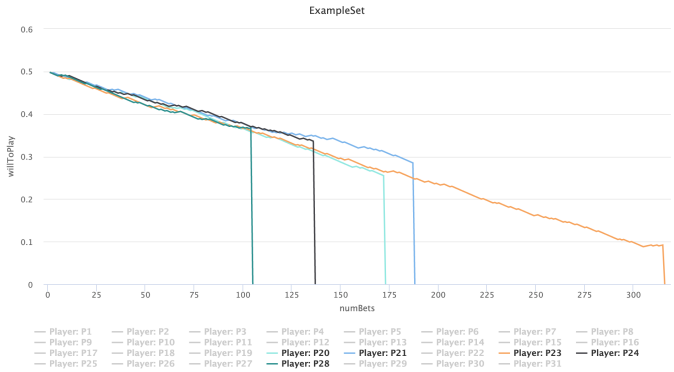

Figure 4.13: Will To Play Win Jackpot Social

We can observe a social gambler's will who plays until they had no more intention to play.

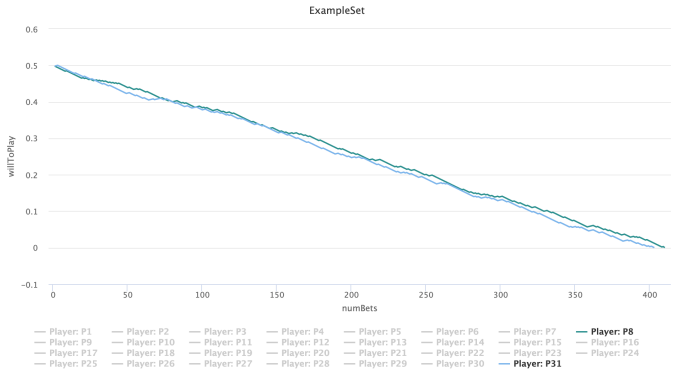

Figure 4.14: Will To Play Losted Social

We can observe an additive gambler's will during the gameplay, and the negative emotions representing the player's moment for a loan.

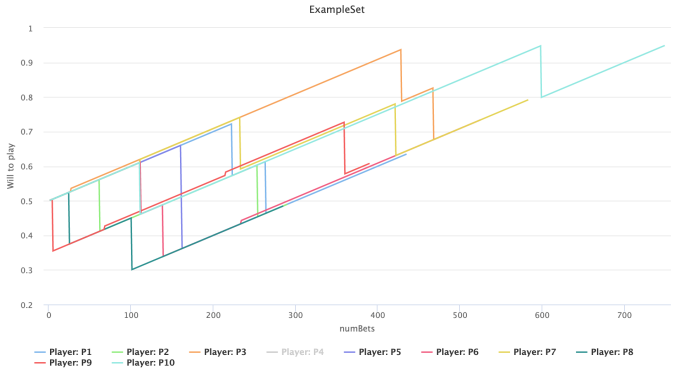

Figure 4.15: Plays of each type

Chapter 5

Conclusions and Future Work

This project will enable the company to better understand the final users' profiles, final customers, and use the results to evolve the game aligned with its vision and intent for innovation. There is the opportunity to create new features for the games, based in the gamblers' behaviours and industry feedback.

This work will also be beneficial for the quality assurance (QA) team, as it will allow the organization to perform tests quicker, with higher quality and security. Incorporating the current test scenarios on this project's developments will enable the QA team to extract more data from the tests, improve them, and innovate. The mathematics department will also benefit from this simulation because it allows conducting studies directed for specific markets. On the team we were allocated, Tech Consultancy, they are also developing a tool for automated tests where they can apply the players' profile and in which this project will play a significant role.

The main challenge we faced during the development phase was the fact that the access to real data was firstly denied, which delayed the entire project development. Initially, we were supposed to use the company's data, however, there were some data privacy questions such as "who's the data owner?" or "can we share the data we collected from our clients?" that were raised and made it impossible to progress on the development of the project. Finding data related to poker or other card games was easy as many poker players use programs to track data from all the players at the table, which is not the case for slot machine games.

5.1 Future work

Considering the scope of the project there are lot of opportunities ahead for its applicability. The next step is to create an interface to manage the parameters of the game and the players such as the denomination of the game, the initial available balance, and other player agent's rules gathered from new studies on the gamblers' habits. After this, since the company has multiple types of games (slot machines, bingo, baccarat, and others) the engine would need to be modified to allow extra data attributes and make this project functional for the entire company's portfolio.

On top of this, the results of this project were retrieved using the Decision Tree, as we already had a pre-classification of the data[1], however, this might not be the case for other real data sets. Thus, for future data to be analyzed it will be needed to also apply k-NN[2] method which will firstly prepare and group the data, adding an extra layer of complexity to the project.

[1]The dataset applied on this project, was real data used by the Responsible Gambling commission and already had flags for addictive players.

[2]K-NN (k-nearest neighbors algorithm) is a non-parametric classification method first developed Evelyn Fix and Joseph Hodges in 1951 that is used for classification and regression. In both cases, the input consists of the k closest training examples in data set [12].

Appendix A

Tables

A.1 Responsible Gambling Event IDs, Labels, and Descriptions

ID	Label	Description
1	3rd person contact	A 3rd party (e.g., a relative) contacts bwin.party customer service (CS) to get the account blocked due to RG reasons.
2	account closure/reopening	The subscriber CS to have his account closed due to problem gambling or reopened after a closure due to problem gambling.
3	cancel outpayment	The subscriber issues an out-payment from the bwin.party account but then calls CS to cancel.
4	change of limit: manual change	The user requested a change in the personal deposit limit and was advised by CS to make the limit change in the interface.
6	fair play - heavy complainer	The subscriber complains heavily about fair play.
7	inpayment block request	The subscriber requests to block one method of in-payment to the bwin.party account (e.g., payment from bwin.party to the subscriber's credit card).
8	minor case	The subscriber or someone else identifies the subscriber is identified as a minor.
9	partial block	The subscriber requests bwin.party; CS to block one or more (but not all) products due to problem gambling.
10	problem reported	The subscriber reports a problem, which may or may not be justified.
11	user request higher limit	The subscriber requests a personal deposit limit which is above the standard permitted deposit limit.
12	two events on the same day	Subscriber experienced two RG events on the same day.
13	not reported by bwin	Event type not reported by bwin.party.
0	no event	-

A.2 Customer Service Intervention Codes and Descriptions

ID	Description
1	bwin.party gives additional RG advice to subscriber.
2	Subscriber's account is re-opened upon customer request after closure or self-exclusion period had elapsed.
3	The customer's request was technically not possible.
4	Account blocking for investigation and asking the customer for a detailed statement about his situation.
5	Change of personal deposit limit from 'VIP' subscriber accepted.
6	Request for partially blocked account block could not be conducted completely.
7	The third party who has made a request is advised about privacy regulation; the customer is contacted and asked for a statement; the customer account remains open until further evidence indicates confirms suspections.
8	Request for partially blocked account block could be conducted completely.
9	Inpayment method not blocked as requested.
10	Inpayment method blocked
11	Request for higher personal deposit limit denied.
12	Request for higher personal deposit limit accepted.
13	Daily/weekly personal deposit limit is changed.
14	Imposed account block by bwin.
15	Betting limit changed.
16	Account remains closed.
17	Account is blocked and reimbursement is given.
18	Partial block of products requested by subscriber is not possible
0	no event

A.3 Product Type IDs, Names, and Types

ID	Name	Type
1	sportsbook fixed odds	Fixed odds
2	sportsbook live action	Live action
3	poker boss media 4*	Poker
4	casino boss media 2	Casino
5	Supertoto*	Games
6	game VS*	Games
7	games bwin*	Games
8	casino chartwell	Casino
9	race book*	Games
10	poker bwin*	Poker
14	games football*	Games
15	Minigames*	Games
16	mobile games*	Games
17	mobile casino	Casino
19	gamArena*	Games
20	Gratta e Vinci*	Games
21	Fantasy Sports*	Games
22	Paradice*	Games
23	Backgammon*	Games
24	Sidegames*	Games
25	Bingo*	Games

* For these third-party (vendor) products, we consider turnover and hold information included in the daily aggregates file sent from bwin.party invalid due data storage procedures. In Raw Datset II.Daily aggregates, we have set to "missing" the turnover and hold values for these products. For these products, we urge caution in interpretation of the variable "number of bets." In Braverman, LaPlante, Nelson and Shaffer (2013), we do not analyze wager-related data for all Games and Poker product type [22].

A.4 Raw Dataset I

Variable	Type	Description
USERID	Numeric	User id
RG_case	Numeric	Subscriber's group
CountryName	String	Subscriber's country of residence
LanguageName	String	Subscriber's preferred language
Gender	String	Subscriber's gender
YearofBirth	Numeric	Subscriber's year of Birth
Registration_date	Date	Subscriber's bwin.party registration date
$First_Deposit\_Date$	Date	Subscriber's bwin.party first deposit date

A.5 Raw Dataset II

Variable	Type	Description
UserID	Numeric	User id
Date	Date	Date of transaction
ProductType	Numeric	Product type
Turnover	Numeric	Amount staked
Hold	Numeric	Amount lost
NumberofBets	Numeric	Number of bets

A.6 Raw Dataset III

Variable	Type	Description
UserID	Numeric	User id
RGsumevents	Numeric	Number of RG events
RGFirst_Date	Date	First RG event date
RGLast_date	Date	Last RG event date
Event_type_first	Numeric	Type of first RG event
Interventiontype_first	Numeric	Type of first customer service intervention

A.7 Raw Dataset Analytic

Variable	Type	Description
USERID	Numeric	user ID
age	Numeric	Age
gender	Numeric	Gender
RG_case	Numeric	RG case
ValidationSet	Numeric	Validation or Training set
first_active_product1_31days	Date	First active date for fixed odds
first_active_product2_31days	Date	First active date for live action
first_active_product4_31days	Date	First active date for casino type products
first_active_games_31days	Date	First active date for other games
first_active_poker_31days	Date	First active date for poker
p1sumstake31days	Numeric	Total wagered in fixed odds in 31 days since the first deposit date
p1sumbets31days	Numeric	Total number of bets in fixed odds in 31 days since the first deposit date
p1totalactivedays_31days	Numeric	Total active days in 31 days since the first deposit date for fixed odds
p1SDStakes31days	Numeric	Variability of wagers in fixed odds in 31 days since the first deposit date
p1SDBets31day	Numeric	Variability of number of bets per day in fixed odds in 31 days since the first deposit date
p2sumstake31days	Numeric	Total wagered in live action in 31 days since the first deposit date
p2sumbets31days	Numeric	Total number of bets in live action in 31 days since the first deposit date
p2totalactivedays_31days	Numeric	Total active days in 31 days since the first deposit date for live action
p2SDStakes31days	Numeric	Variability of wagers in live action in 31 days since the first deposit date
p2SDBets31days	Numeric	Variability of number of bets per day in live action in 31 days since the first deposit date
pcsumstake31days	Numeric	Total wagered in casino in 31 days since the first deposit date
pcsumbets31days	Numeric	Total number of bets in casino type games in 31 days since the first deposit date
casino_totalactivedays_31days	Numeric	Total active days in 31 days since the first deposit date for casino

Variable	**Type**	**Description**
pcSDStakes31days	Numeric	Variability of wagers in casino type games in 31 days since the first deposit date
pcSDBets31days	Numeric	Variability of number of bets per day in casino type games in 31 days since the first deposit date
pgsumbets31days	Numeric	Total number of bets in other games in 31 days since the first deposit date
games_totalactivedays_31days	Numeric	Total active days in 31 days since the first deposit date for other games
pgSDBets31days	Numeric	Variability of number of bets per day in other games in 31 days since the first deposit date
firstGamePlayed	Numeric	First game played
p3totalactivedays_31days	Numeric	Total active days in 31 days since the first deposit date for poker
p1avgbetsperactiveday	Numeric	Average number of bets per active day of fixed odds
p2avgbetsperactiveday	Numeric	Average number of bets per active day of live action
pcavgbetsperactiveday	Numeric	Average number of bets per active day of casino
p1avgbetsperday	Numeric	Average number of bets per any day during the period of observation of fixed odds
p2avgbetsperday	Numeric	Average number of bets per any day during the period of observation of live action
pcavgbetsperday	Numeric	Average number of bets per any day during the period of observation of casino
p1avgbetsize	Numeric	Average bet size in fixed odds
p2avgbetsize	Numeric	Average bet size in live action
pcavgbetsize	Numeric	Average bet size in casino type games
wk1frequency	Numeric	Number of active days during the 1st week
wk2frequency	Numeric	Number of active days during the 2nd week
wk3frequency	Numeric	Number of active days during the 3rd week
wk4frequency	Numeric	Number of active days during the4th week
wk1_p1sumstakes	Numeric	Total wagered during 1st week for fixed odds
wk1_p2sumstakes	Numeric	Total wagered during 1st week for live action
wk1_pcsumstakes	Numeric	Total wagered during 1st week for casino type games
wk2_p1sumstakes	Numeric	Total wagered during 2nd week for fixed odds

Variable	**Type**	**Description**
wk2_p2sumstakes	Numeric	Total wagered during 2nd week for live action
wk2_pcsumstakes	Numeric	Total wagered during 2nd week for casino
wk3_p1sumstakes	Numeric	Total wagered during 3rd week for fixed odds
wk3_p2sumstakes	Numeric	Total wagered during 3rd week for live action
wk3_pcsumstakes	Numeric	Total wagered during 3rd week for casino
wk4_p1sumstakes	Numeric	Total wagered during 4th week for fixed odds
wk4_p2sumstakes	Numeric	Total wagered during 4th week for live action
wk4_pcsumstakes	Numeric	Total wagered during 4th week for casino
wk1_p1sumbets	Numeric	Number of bets during 1st week for fixed odds
wk1_p2sumbets	Numeric	Number of bets during 1st week for live action
wk1_pcsumbets	Numeric	Number of bets during 1st week for casino type games
wk1_pgsumbets	Numeric	Number of bets during 1st week for other games
wk2_p1sumbets	Numeric	Number of bets during 2nd week for fixed odds
wk2_p2sumbets	Numeric	Number of bets during 2nd week for live action
wk2_pcsumbets	Numeric	Number of bets during 2nd week for casino type games
wk2_pgsumbets	Numeric	Number of bets during 2nd week for other games
wk3_p1sumbets	Numeric	Number of bets during 3rd week for fixed odds
wk3_p2sumbets	Numeric	Number of bets during 3rd week for live action
wk3_pcsumbets	Numeric	Number of bets during 3rd week for casino type of games
wk3_pgsumbets	Numeric	Number of bets during 3rd week for other games
wk4_p1sumbets	Numeric	Number of bets during 4th week for fixed odds
wk4_p2sumbets	Numeric	Number of bets during 4th week for live action
wk4_pcsumbets	Numeric	Number of bets during 4th week for casino type games

Variable	**Type**	**Description**
wk4_pgsumbets	Numeric	Number of bets during 4th week for other games
wkend_p1sumstakes	Numeric	Total wagered in fixed odds during the weekends
wkend_p2sumstakes	Numeric	Total wagered in live action during the weekends
wkend_pcsumstakes	Numeric	Total wagered in casino type games during the weekends
wkend_p1sumbets	Numeric	Number of bets placed in fixed odds during the weekends
wkend_p2sumbets	Numeric	Number of bets placed in live action during the weekends
wkend_pcsumbets	Numeric	Number of bets placed in casino type games during the weekends
wkend_pgsumbets	Numeric	Number of bets placed in other games during the weekends
wkday_p1sumstakes	Numeric	Total wagered in fixed odds during the weekdays
wkday_p2sumstakes	Numeric	Total wagered in live action during the weekdays
wkday_pcsumstakes	Numeric	Total wagered in casino type games during the weekdays
wkday_p1sumbets	Numeric	Number of bets placed in fixed odds during the weekdays
wkday_p2sumbets	Numeric	Number of bets placed in live action during the weekdays
wkday_pcsumbets	Numeric	Number of bets placed in casino type games during the weekdays
wkday_pgsumbets	Numeric	Number of bets placed in other games during the weekdays
mostFrequentGame	Numeric	Most Frequent Game
playedFO	Numeric	Played fixed odds at least 3 times
playedLA	Numeric	Played live action odds at least 3 times
playedPO	Numeric	Played poker at least 3 times
playedCA	Numeric	Played casino type games at least 3 times
playedGA	Numeric	Played other games at least 3 times
NumberofGames31days	Numeric	Number of games during the first 31 days since the first deposit date

Variable	**Type**	**Description**
weekfrequencytraj	Numeric	Trajectory by weekly frequency
FOBetsWeeklyTraj	Numeric	FO. Trajectory by weekly bets
FOstakesWeeklyTraj	Numeric	FO. Trajectory by weekly stakes.
LABetsWeeklyTraj	Numeric	LA. Trajectory by weekly bets.
LAstakesWeeklyTraj	Numeric	LA. Trajectory by weekly stakes
CasinoBetsWeeklyTraj	Numeric	Casino. Trajectory by weekly bets.
CasinostakesWeeklyTraj	Numeric	Casino. Trajectory by weekly stakes
GamesBetsWeeklyTraj	Numeric	Games. Trajectory by weekly bets.
p1wkendsumbetssratio	Numeric	ratio of number of bets made during weekends compared to weekdays for fixed odds
p2wkendsumbetsratio	Numeric	ratio of number of bets made during weekends compared to weekdays for live action
pcwkendsumbetsratio	Numeric	ratio of number of bets made during weekends compared to weekdays for casino type games
pgwkendsumbetsratio	Numeric	ratio of number of bets made during weekends compared to weekdays for other games
p1wkendsumstakesratio	Numeric	ratio of total wagers made during weekends compared to weekdays for fixed odds
p2wkendsumstakesratio	Numeric	ratio of total wagers made during weekends compared to weekdays for live action
pcwkendsumstakesratio	Numeric	ratio of total wagers made during weekends compared to weekdays for casino
period_tillDeposit	Numeric	Number of days since registration till the first deposit date (31 max)
RiskGroup1	Numeric	Risk Group 1: Many games, active casino players
RiskGroup2	Numeric	Risk Group 2: Two games, ActiveLA players
RiskGroupCombined	Numeric	Risk group 1 or 2 combined
totalactivedays_31days	Numeric	Total active days in 31 days since the first deposit date
totalactivedaystillDeposit_31_days_max	Numeric	Total active days till the first deposit date

A.8 Players

Variable	Type	Description
Type	String	Type of player, Social or Addictive
Player	String	Player Id
denomination	Numeric	The value of a bet line in euros
initialMoney	Numeric	The initial amount of money
wageredCredits	Numeric	The total credits wagered so far
earnedCredits	Numeric	The total credits earned so far
lastPrize	Numeric	The prize from each bet
ifPrize	Numeric	Variable to count the winnings and loses
availableCredits	Numeric	The available credits after a bet
numBets	Numeric	Bet counter
loanCount	Numeric	Loan counter
loanCredits	Numeric	The amount of the loan
willToPlay	Numeric	The will to play after each bet

Appendix B

Images

B.1 Images from Social Gambler Simulation

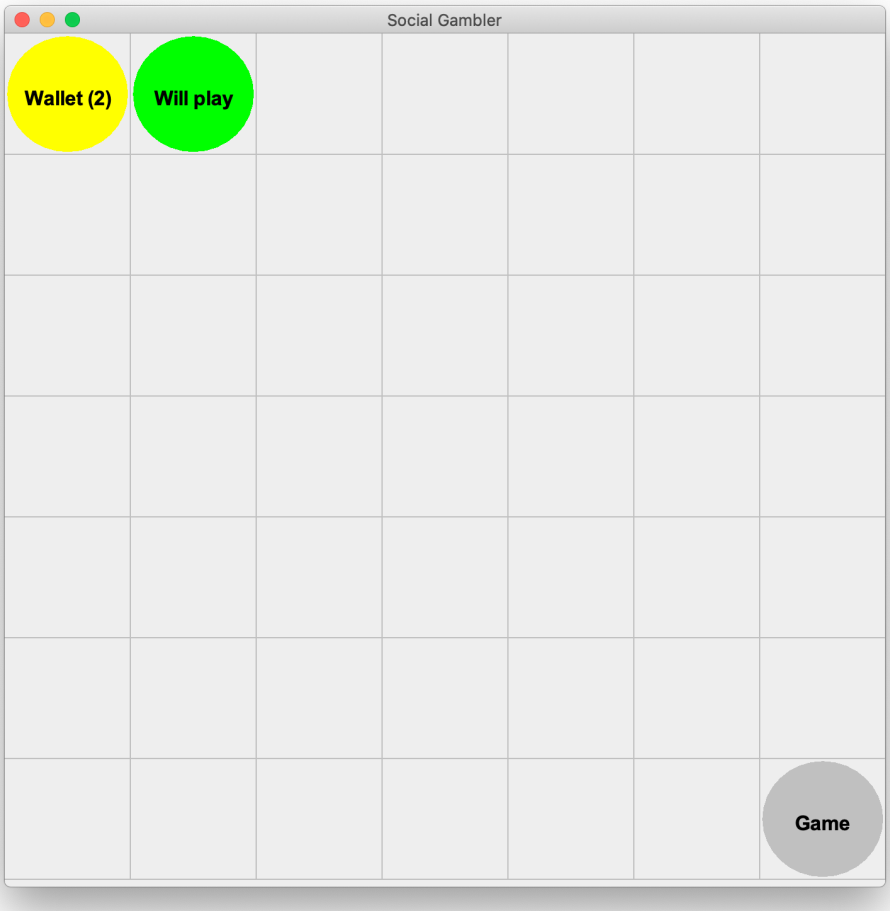

Figure B.1: Final state without money and yet Will to Play positive

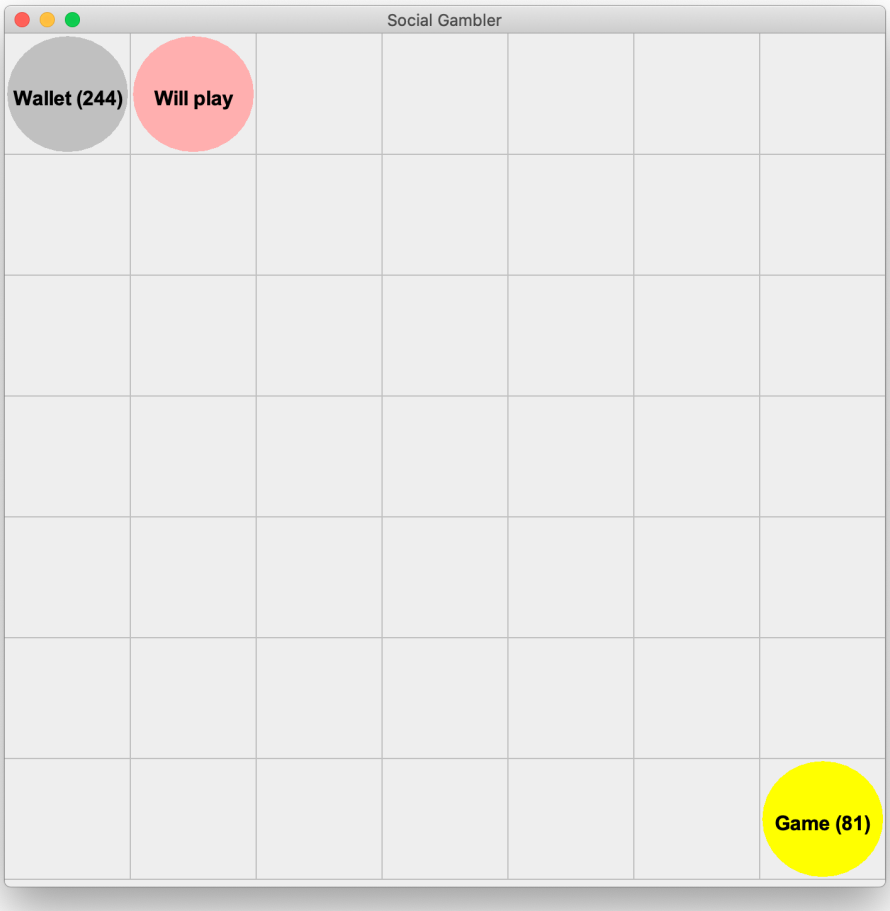

Figure B.2: Final state without will to play because SG wins a jackpot

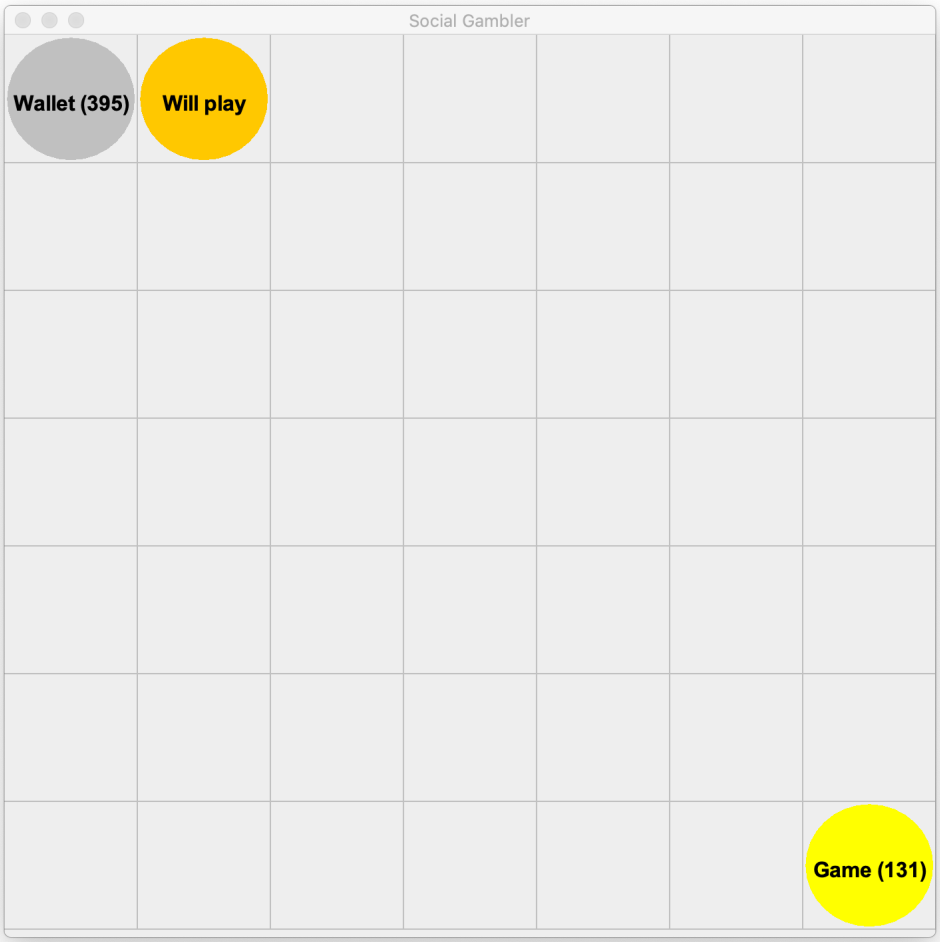

Figure B.3: SG are losting their will to play

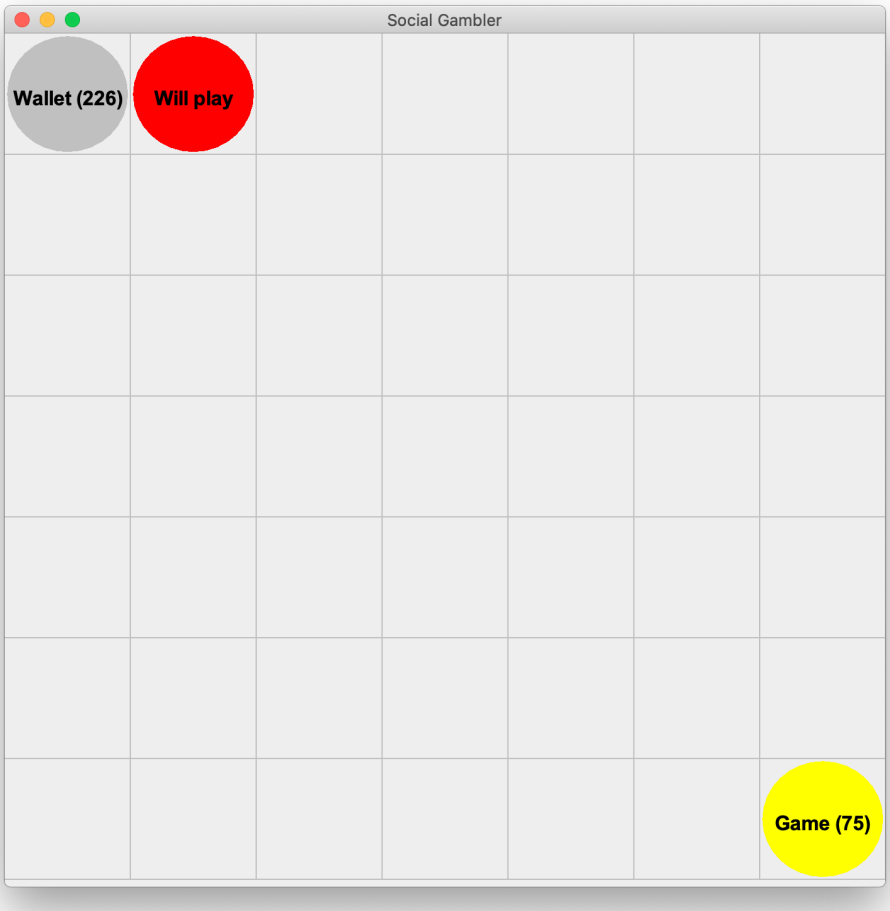

Figure B.4: SG lost their will to play

B.2 Images from Addictive Gambler Simulation

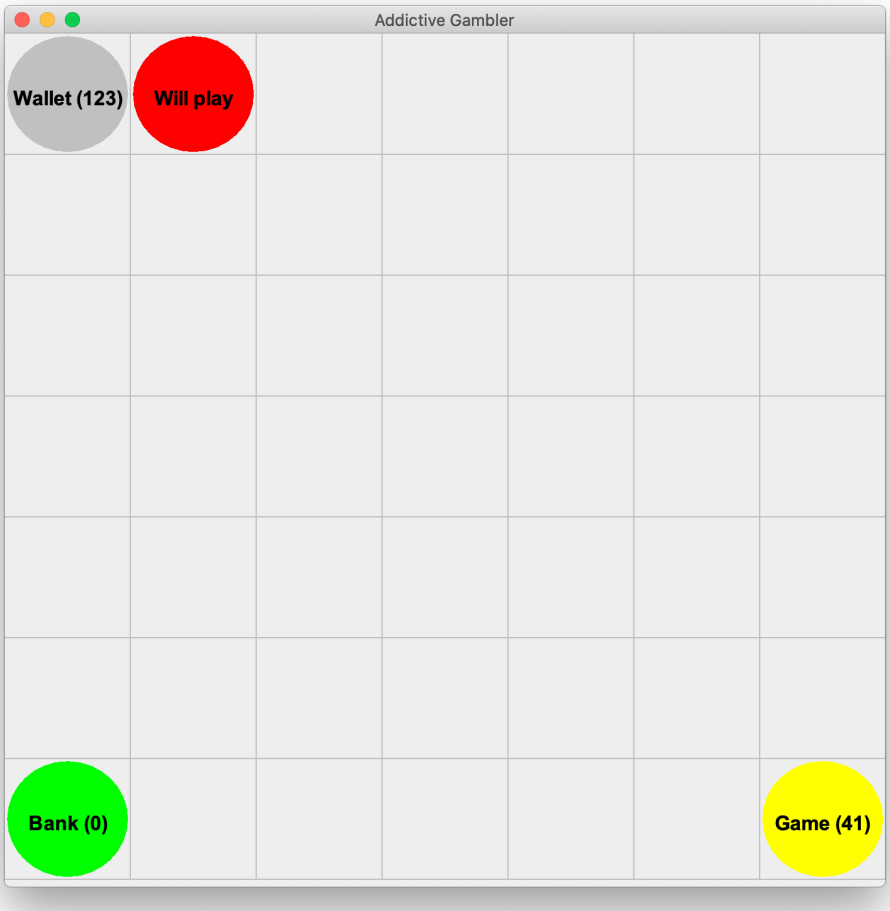

Figure B.5: AG uncontrolled

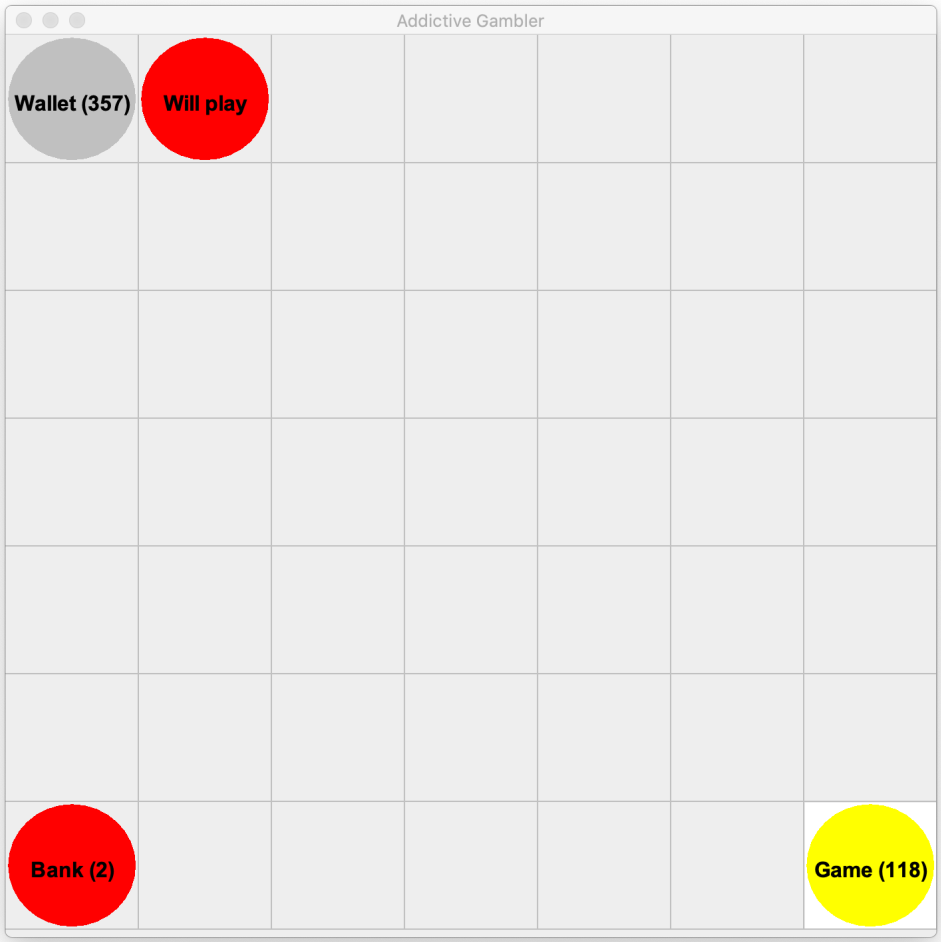

Figure B.6: AG used their two opportunities to use the bank

References

[1] João Emílio Almeida. Serious Games as a Behaviour Elicitation Tool : Applications to Evacuation Scenarios. 2016.

[2] João E. Almeida, Rosaldo J. F. Rossetti, and António Leça Coelho. Crowd simulation modeling applied to emergency and evacuation simulations using multi-agent systems, 2013.

[3] James Bucanek. *Model-View-Controller Pattern. In: Learn Objective-C for Java Developers*. Apress, 2009.

[4] G. Calleja. *In-Game: From Immersion to Incorporation.* The MIT Press, 2011.

[5] Adele Cutler and Leo Breiman. Archetypal analysis. *Technometrics*, 36(4):338–347, 1994.

[6] Catherine Dawson and Catherine Dawson. *Game analytics*. 2019.

[7] Vahid Garousi and Frank Elberzhager. Test Automation: Not Just for Test Execution. *IEEE Software*, 34(2):90–96, 2017.

[8] Oyku Isik, Mc Jones, and Anna Sidorova. Psychological contracts and job satisfaction: Clustering analysis using evidential C-means and comparison with other techniques. *Intelligent systems in accounting, finance and management*, 176(January):161–176, 2011.

[9] Eric Zimmerman Katie Salen. *Rules of Play: Game Design Fundamentals*. MIT Press, 1 edition edition, 2003.

[10] A. Kerr. *The Business and Culture of Digital Games*. SAGE Publications, 1 edition edition, 2006.

[11] John Kirriemuir and Angela Mcfarlane. Literature Review in Games and Learning, 2004. A NESTA Futurelab Research report - report 8.

[12] Arijit Laha. Building contextual classifiers by integrating fuzzy rule based classification technique and k-nn method for credit scoring. *Advanced Engineering Informatics*, 21(3):281 – 291, 2007. Applications Eligible for Data Mining.

[13] Rajesh Mathur, Scott Miles, and Miao Du. Adaptive Automation: Leveraging Machine Learning to Support Uninterrupted Automated Testing of Software Applications. 2015.

[14] J. McMillen. *Gambling Cultures: Studies in History and Interpretation*. Culture: Policy and Politics. Taylor & Francis, 2005.

[15] Michelle McPartland and Marcus Gallagher. Creating a multi-purpose first person shooter bot with reinforcement learning. *2008 IEEE Symposium on Computational Intelligence and Games, CIG 2008*, pages 143–150, 2008.

[16] Frederico S Miranda and Paulo C Stadzisz. Jogo Digital: definição do termo. *SBGames*, (May):296–299, 2017.

[17] I. Mosca. *Social Ontology of Digital Games*. In M. C. Angelides and H. Agius, handbook ofdigital games, chapter 23, page 767. ieee press, 1 edition edition, 2014.

[18] W. James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences of the United States of America*, 116(44):22071–22080, 2019.

[19] Ushma Kesha Patel, Purvag Patel, Henry Hexmoor, and Norman Carver. Improving behavior of computer game bots using fictitious play. *International Journal of Automation and Computing*, 9(2):122–134, 2012.

[20] Lior Rokach and Oded Maimon. *Decision Trees*, pages 165–192. Springer US, Boston, MA, 2005.

[21] Ariel Rosenfeld and Orel Zang. Automation of Android Applications Functional Testing Using Machine Learning Activities Classification. pages 122–132, 2018.

[22] National Security. Codebook for Using cross-game behavioral markers for early identification of high-risk internet gamblers. 2019.

[23] Guoqiang Shu and David Lee. Testing Security Properties of Protocol Implementations – a Machine Learning Based Approach. 2007.

[24] J. F. Silva, J. E. Almeida, R. J. F. Rossetti, and A. L. Coelho. A serious game for evacuation training. In *2013 IEEE 2nd International Conference on Serious Games and Applications for Health (SeGAH)*, pages 1–6, 2013.

[25] Jouni Smed and Harri Hakonen. Synthetic players 7 a quest for artificial intelligence in computer games. *Human IT*, 7(3):57–77, 2005.

[26] Jouni Smed, Timo Kaukoranta, Harri Hakonen, Oy L M Ericsson Ab, R Telecom, and Fin Turku. AIsHockey-A Platform for Studying Synthetic Players AIsHockey — A Platform for Studying Synthetic Players. (February 2003), 2014.

[27] Sun-Chong Wang. *Artificial Neural Network*, pages 81–100. Springer US, Boston, MA, 2003.

[28] Erik Wendel. Cheating in Online Games. (July), 2012.

[29] M. J. P. Wolf. *The Video Game Explosion : A History from PONG to PlayStation and Beyond.* Greenwood Press, 1 edition edition, 2007.

[30] Georgios N. Yannakakis, Pieter Spronck, Daniele Loiacono, and Elisabeth André. Player Modeling. *Dagstuhl Follow-Ups*, 6:59, 2013.