Universidade do Minho
Escola de Engenharia

João Miguel Pereira Alves

Knowledge Extraction from the Behaviour of Players in a Web Browser Game

Setembro de 2013

Universidade do Minho

Escola de Engenharia

Departamento de Informática

João Miguel Pereira Alves

Knowledge Extraction from the Behaviour of Players in a Web Browser Game

Dissertation

Master's Degree in Informatics Engineering

Work done under the supervision of:

José Neves, University of Minho
Martin Riedmiller, University of Freiburg

Setembro de 2013

Acknowledgments

The path travelled until the conclusion of my thesis has been hard and full of difficulties but even so it was a wonderful and rewarding path to take. Reaching the end would not have been possible without the efforts of many people to whom I am grateful.

The first acknowledgement of course needs to go to my supervisors José Neves, Martin Riedmiller and Sascha Lange. Their expertise, mentoring and patience were invaluable to the completion of this thesis.

I would like to express my gratitude to all my close friends, especially the ones I made in Freiburg during this amazing year, for the constant encouragement and advices.

Finally, I could not end without mentioning my parents Victor and Margarida Alves, my brothers Pedro and Daniel Alves, my girlfriend Eeva and the rest of my family, who were always there to support me.

Title

Knowledge Extraction from the Behaviour of Players in a Web Browser Game

Abstract

The analysis of the player's behaviour is a requirement with growing popularity in the traditional computer games segment and has been proven to aid the developers create better and more profitable games. There is now interest in trying to replicate this attainment in a less conventional genre of games known as web browser games.

The main objective of this work is to analyse and create a technique for the analysis of the behaviour of the players inside a web browser game. For this analysis a system to automatically collect, process and store the relevant data for the referred analysis was developed. The web browser game used as a case study for this work is developed by *5DLab* and is called *Wack-a-Doo.* The work developed focused on creating short-term prediction models using the information collected during the first days of playing for each player. The objectives of these models are to predict the time played or the conversion state of the players. With the study of the created models it was possible to extract results that provide potentially useful information to increase the profitability of *Wack-a-Doo.*

Título

Extração de Conhecimento Através do Comportamento de Jogadores num Jogo de Browser Web

Resumo

A análise do comportamento de jogadores é uma prática com crescente popularidade no segmento dos jogos de vídeo tradicionais. Esta técnica foi já aprovada como capaz de ajudar os criadores a desenvolver melhores e mais lucrativos jogos. Existe agora interesse em tentar replicar este sucesso num género de jogos de vídeo menos convencionais normalmente referidos como jogos de browser web.

O objetivo deste trabalho é analisar e criar uma técnica para essa análise do comportamento dos jogadores de um jogo de browser web. Para isto um sistema automático de recolha, processamento e armazenamento dos dados relacionados com o comportamento dos jogadores foi desenvolvido. O jogo de browser web usado para este estudo foi criado pela empresa *5DLab* e dá pelo nome de *Wack-a-Doo.* O trabalho desenvolvido centrou-se em fazer modelos de previsão de curto prazo usando as informações recolhidas durante os primeiros dias de jogo de cada jogador. Estes modelos têm como objetivo prever o tempo jogado e o estado de conversão do jogador. Estudando os modelos criados foi possível extrair resultados que fornecem informação potencialmente útil para melhorar a rentabilidade do *Wack-a-Doo.*

Contents

List of Figures

List of Tables

List of Equations

List of Acronyms

CSV - Comma-Separated Values

KPI - Key Performance Indicators

MMOG - Massive Multiplayer Online Game

MTS - Multivariate Time Series

NPC - Non-Player Character

RIPPER - Repeated Incremental Pruning to Produce Error Reduction

SysCoPPE - System for Collection and Processing of Player Evolution

1 Introduction

1.1 Motivation

Browser games are computer games that have the unique characteristic of being played directly on the web browser, without the need of additional software requirements. Typically games in this genre offer a multiplayer environment where users can interact with each other. Attending to the way this genre works, the current game status of the players needs to be stored by the game provider. This fact makes it easy to collect information about the players, since companies already need to store it in order for their games to work.

Nowadays not only Internet access is almost universal in a modern house but also a growing number of people carry smart phones that have web browsing capabilities [1]. With this rise in Internet accessibility and the growing popularity of social platforms there is an increasing awareness in the game developers community about browser games [2].

The web browser is responsible, not only for the player interface but also for the access to the game world. This characteristic is also one of the biggest strengths of these types of games because it reduces the setting up time to only a fast registration, which allows the player to start playing the game very quickly.

Traditionally the business model for computer games involves selling a game package to the customer that enables him to play the game. However, by definition, in browser games there is no package to be sold. To be able to gain popularity and to attract new players, due to fierce competition, these games are available free of charge. Because of this constraint developers have to resort to other sources of revenue.

The easy accessibility of browser games is what makes this genre popular and is the reason why so many people who would never buy a traditional computer game start playing [3]. Currently the popularity of browser games is rising and the future of computer games could very well be in this genre as the software industry is facing a paradigm shift towards web-based software [4]. The recent growth in the number of users of browser games is largely motivated by the popularity of online social networks, such as Facebook, that allow the easy introduction of many social features in the games of this genre [5] [6].

One of the best examples of the potential of this business model is of a very successful company called Zynga[1]. This company was founded in 2007 in San Francisco with their main focus being on developing browser games. The success of their games was so great that since only 2007 Zynga generated a revenue of more than 1500 million dollars [7]. Zynga's most famous and successful game was *Farmville*, which reached 10 million daily active users in only 6 weeks [8].

This growth in popularity of free-to-play games has created a new research perspective on behaviour analysis. Because the major revenue stream for the developers of these kind of games is selling in-game content through micro transactions, the analysis of the player's preferences and their development is truly crucial for the game's financial success [3]. The games developed under this game model require a constant analysis of the player behaviour in order to maintain their financial sustainability [9].

1.2 Scope

The purpose of this work is to perform an analysis of the behaviour from the players in a browser game in order to extract relevant knowledge to the company developing the game. In the case of

[1] www.zynga.com

this study the game used as a the case study was developed by a startup called *5DLab*[2]. One of the projects of *5DLab* is the development of a browser-based massively multiplayer online strategy game, called *Wack-a-Doo*[3]. *Wack-a-Doo* makes money when users pay for in-game premium content, this content gives them small bonuses or access to new game features.

For this analysis it was developed a System for Collection and Processing of Player Evolution (SysCoPPE) to automatically collect and process players data on a daily basis. SysCoPPE allowed the aggregation of player data related to their behaviour inside the game. As an example some of the data collected refers to the number of logins per day, the time played or in-game scores.

After this collection and processing phase the player data is then analysed in order to extract knowledge and to make predictions about the *Wack-a-Doo* players behaviour. For this analysis it was resorted to the use of different statistical and data mining techniques.

1.3 Objectives

The main objective of this work is to analyse and create a technique for the analysis of the behaviour of the players inside a web browser game. This work has also the aim to create a system to automatically collect, process and store the relevant data for the referred analysis.

With the analysis of the player's behaviour it is expected to extract useful knowledge such as interesting behavioural patterns or accurate playing predictors that help the developers make a better and more engaging game.

[2] www.5dlab.com

[3] www.wack-a-doo.com

To achieve the analysis goals mentioned in the last paragraph it was defined a set of objectives that would enable the collection, treatment and processing of the relevant player data.

- Development of a system to collect, process and store player data:
    - Way to collect directly data from *Wack-a-Doo*;
    - System to process the collected data;
    - Database for the collected data;
- Analysis of the collected data:
    - Process the collected data;
    - Use of data mining and statistical tools;
- Evaluation and interpretations of the results achieved.

1.4 Research Methodology

The work developed was made according to the Action-Research methodology. This is a methodology that is often used when trying to solve a problem that aims to obtain information leading to its resolution by the means of an iterative and recurrent process. The Action-Research methodology also assumes the existence of cooperation between all the entities involved in the problem and in its resolution [10].

According to this methodology, not only at the beginning but also during all of the duration of the work there is always the need for a continuous update and analysis of the state of the art and how similar researches have been handled.

The first task in the work developed was to explore *Wack-a-Doo's* database and discover what kind of data was stored by the game about the players.

After this raw analysis of the data there was the need to design and develop a system to easily store and compile the necessary data for this project from *Wack-a-Doo's* database. There is then made an analysis on the data processed by SysCoPPE and the evaluation and interpretation of its results.

1.5 Structure of the document

This document started by introducing some of the key concepts needed to understand the work developed. Chapter 2 begins with an introductory note about the developers of *Wack-a-Doo* and by explaining what is *Wack-a-Doo.* After those explanations it details the functioning of the *Wack-a-Doo* game by explaining its gameplay. This part of Chapter 2 also tries to make a general description of all the game mechanics and features.

Next in Chapter 2 the business model of *Wack-a-Doo* is explained and the way the developers make money with the game detailed. The way that the players can obtain the in-game currency is described along with the company's strategies for selling that currency. Chapter 2 ends with a description of the various ways that *5DLab* uses to advertise *Wack-a-Doo.*

The detailed description of the analysis made on related works is presented in Chapter 3. This chapter gives an insight about the state of the art not only in knowledge extraction in web browser games but also in other related fields such as game mining and social gaming.

Chapter 4 starts by stating what are the goals of this study, in here the details about what it was decided to predict and with which data are presented. After the study goals, still in Chapter 4, the process used to extract data from *Wack-a-Doo* is described and explained in detail.

In Chapter 4 a system for collection and processing of player evolution named SysCoPPE is also proposed. The description of how SysCoPPE works and how its architecture is designed is also included on this chapter. After this description, the details of all the components of this system namely the data extractor, the data processor, the database and the dataset creation interface are presented.

The description and characterization of the datasets created using SysCoPPE is present on Chapter 5. Those datasets are then the ones used for all the experiments made during the period of development of the presented work.

Chapter 6 starts by describing the methodology followed during the experiments referred in this document. Later, the details of the experiments performed are presented along with the most relevant conclusions obtained from those experiments.

The last chapter is Chapter 7, which starts with a brief summary of all the important points in each chapter. It also presents the relevant work created during the process of development of this dissertation. Chapter 7 ends with a description of some of the critical points faced and the list of contributions.

2 Understanding the Game - Wack-a-Doo

2.1 Introduction

The game used as the case study for this work is called *Wack-a-Doo* and is a web browser strategy game developed by a company called *5DLab*. This company is a startup that was founded in 24 February 2012 and is headquartered in Freiburg, Germany. Despite being quite young with around 1 and half years it has already some appreciable portfolio. It specializes in using latest technologies in fields such as Internet and mobile development, or in modern sensor processing algorithms. In their portfolio they have a diverse range of projects such as *Monsters' Traveller Guide to Reality*, *MoMPF Currency Converter*, *Cover Shuffle Player*, *PanzerWars* or even a two time world champion robot soccer team [11].

As of September 2013 *Wack-a-Doo* is still a beta product and is *5DLab's* latest project and main focus in their efforts. *Wack-a-Doo* is a Massive Multiplayer Online Game (MMOG) created originally to be played exclusively in a web browser. However since June last year it is also possible to play *Wack-a-Doo* in any IOS compatible device through an application available in Apple's iTunes store.

Similarly to what occurs in MMOGs, in *Wack-a-Doo* the players can interact in various ways with each other, for example forming alliances, collaborate towards a common objective or even engage in coordinated attacks against other players.

In browser games the playing mechanics usually involve the player making a limited set of actions, for example moving armies or founding cities, that have an associated time cost for completion. As a result the player can leave the game and return only when the time associated to these actions is over and he can perform more in-game actions. Because of these playing mechanics the typical user performs short playing sessions various times a day [12]. The mechanics of *Wack-a-Doo* while having their own original flavour follow this generic formula.

Wack-a-Doo is a strategy-based game where throughout their gameplay the players need to devise tactics and make plans in order to outplay their enemies. These enemies in the case of *Wack-a-Doo* can be other players or artificially controlled Non-Player Characters (NPCs). The game was created following a humorous Stone Age thematic. As an example of the influence of the game's theme the characters that the players control are portrayed as cavemen and the NPCs that roam throughout the game are portrayed as neanderthals.

2.2 Gameplay

The game *Wack-a-Doo* is scheduled in rounds that last at least a few months to finish. In the end of the round all the players have their game progress restarted and need to start from scratch in the beginning of the next round. The time that a round lasts is variable because it depends on certain game objectives being accomplished. In the case of *Wack-a-Doo* the players from an alliance have to collect and protect certain artefacts very hard to obtain in order to win the game and as a result end the round. There is always a very fierce competition around these artefacts by all the top ranking alliances in the game.

Alliances are the player aggregating factor in *Wack-a-Doo.* They are created by one single player who is then the alliance leader and has the task to invite new members and coordinate the alliance in game efforts and strategies. For a player being in an alliance brings many advantages such as security from attacks of other players or the experience of participating in a coordinated organization with all the social interaction between its members. Its also only in an alliance that a player has the ability to compete for the final round objectives or in other words it is an alliance that wins the round and not an individual player. The alliance of a player is identified in their game nickname and its alliance is featured directly after the character '|'.

Figure 2.1 - Wack-a-Doo map region screenshot

The game *Wack-a-Doo* takes place on an expanding map that is divided by regions, that are added to the map according the growth of the player base. In Figure 2.1 it is shown a region of the *Wack-a-Doo* map. Every region has a name that is shown in its top right corner, in the case of the region in Figure 2.1 its name is *Takka Tukka Land*. Each region has a varying maximum number of spots available for settlements, for example in the case of *Takka Tukka Land* the maximum number of settlements is 9.

In *Wack-a-Doo* there are three types of settlements where the player who owns it can construct buildings in. The first one is the capital that is by default the players first city in the game, however this can be latter changed. This settlement is not conquerable by other players and enjoys various military and economical bonuses. The region Takka Tukka Land is the land of the capital of three players, Niki-112, Kol1967 and Guenni from SMA alliance. The map symbol for the capital evolves according to its development, for example in the map it can be sees that Guenni's capital is more developed than the other two.

Other type of settlement is the camp and is also present in *Takka Tukka Land.* The owners of a camp settlement in this region are players *Ratman* and *Quokka from alliances SOA* and *SzH* respectively. Contrary to capitals a player can have more than one camp, however only one camp or capital is allowed per map region. A camp is founded using a special unit called *Little Chief* and can be conquered by enemy players. There is one little chief present in the *Takka Tukka Land* region, which corresponds to the unit that can be found more to the left of the picture. A camp is less powerful than a capital but it is nevertheless a good way for a player to expand its power.

The last of the three types of settlements is the one that can be found in the centre of every map region and it is called the fortress settlement. The fortress is a very unique kind of settlement and plays a very important role in the regional domination competition that every player must participate in order to be successful. The special thing about the fortress is that it can charge taxes up to 15% of the production of all other settlements of the region. Because of this there are constants power struggles to control the fortress or at least the tax rate of the region. It is not uncommon for players to join forces against the owner of the fortress of their region because of a high tax rate. In the case of *Takka Tukka Land* region the fortress is controlled by the player called Ratman. Apart from the tax bonus there are not more advantages in owning a fortress, for example the building space is smaller and it is frequently under the attack of other players.

In *Wack-a-Doo* every unit belongs to one of two categories, the category of the units that have an utility purpose or the category of the units that have a military purpose. In the utility category there are units such as *Little Chiefs* that have the utility of founding camps, or *Trading Carts* that enable the trade of resources between players. In the military category there is three types of units, the ranged units, monted units and melee units. Each type has its own weekness and strengthness and even the units inside the types can have distinctive perks and attributes. All these kind of units make *Wack-a-Doo* a complex and very strategic intensive game to play, however it is this variety that makes the game fun to play where each player can have its own unique strategy.

A player can create a limited number of armies that varies depending on their state of development in the game. An army is constituted by military units, that can be added or removed according to the player desires. It should be noted that this kind of army edition is only allowed in the settlement where the units were created. The strenght of an army depends not only on the ammount and on which units it has but also the experience that it accumulates in battles. An army has a maximum number of actions allowed that are replenished with time, but because of this maximum the carefull planning of the movement and attacks of the armies is crucial for a player not be caught off guard.

The movement of armies between the map points is not instantanious and for an army to travel to another map regions it needs to be in the fortress map point. It is possible for armies from diferent players to be present in the same map point. A map point is all the places where a settlement exists or can be built. In the case of *Takka Tukka Land* there is an army in the *Niki-112* capital settlement map point.

The other army type present on *Wack-a-Doo* are the NPCs *neanderthals* armies that are scattered around the map and randomly disturb the players objectives. Although they can be very inconvenient, the *neanderthals* can also be used for the players advantage. One example where this is done is when players attack the *neanderthals* armies just for the battle experience they gain from these encounters.

An army can fight battles against other armies or settlements and the result from the battles or in other words the ammount of units and buildings lost varies depending on the strength from each side of the battle. It should be noted that armies that are defending settlements get a huge boost that depends on the state of development and the type of settlement being defended.

The types of resources that a player can produce in *Wack-a-Doo* are stone, wood and fur. These resources can be used to construct everything from new buildings, armies or even to pay another player in order to avoid an attack

Figure 2.2 - Inside a Wack-a-Doo capital settlement screenshot

The interface that represents the state of a capital settlement from some player is exemplified in Figure 2.2. The resources available to the player are presented on the top of the image, and it can be seen that the player has 26184 of stone and is creating 1172 stone per hour, has 22178 wood and is creating 1171 wood per hour, and at last has 2274 fur and it is creating 634 fur per hour. There is an infinite quantity of resources that a player can store, in the case of the player in Figure 2.2 he can only store 30440 units of each resource.

Bellow the resource information there is a small pane, the one with one arrow pointing to the left and another arrow pointing to the right, where the information about the specific settlement selected is presented. On the top left of this pane the name of the settlement is visible, in the case of Figure 2.2 the name chosen by the player is *Bear Land*.

In the settlement information pane there are two lines that give all the general information needed. The first line indicates for example that the combat bonus in *Bear Land* is 900%, that the building construction speed is accelerated 402%, that the melee unit production is accelerated 125% or that there are 2 armies out of a maximum of 3 associated with this settlement. The second line informs the player that this settlement has a total of 28 building out of a maximum of 40, that the population is 3092 and informs about the resource production in this particular settlement. In this last line there is also a new type of resource that was not mentioned previously called *golden frogs*. This resource can be acquired directly in the *Wack-a-Doo* shop and will carefully described in the next section.

The rest of the interface in Figure 2.2 is a scheme about how the settlement is organized and how are the building distributed. A settlement has a limited number of places where buildings can be constructed, in the case of *Bear Land* the maximum is 40. It should be noted that not all places on a settlement are equal, some allow for more upgrades on the building on its place to be built or allow the construction of some special buildings. To create a building the player selects and queues what to construct and after some time, depending on the building selected, the queued buildings will be done constructing one by one according to their ordering. By default the maximum number of buildings that can be on the construction queue of a settlement without the use of any bonus is two.

The buildings in *Wack-a-Doo* can be divided in four categories, the unit producing buildings category, the resource mining category, the bonus category and finally the utility category. In the unit producing category there are the buildings that produce military units, for example the *Training Grounds* or the *Firing Range* building, or utility units, such as the *Little Chiefs* produced in the *Campfire* building. The fur, wood and stone mining buildings comprise all of the members of the resource mining category. As an example of a building beloging to the bonus category is the *Windproof Hut* that accelerates the construction of buildings or the *Training Cave* that gives experience to the settlement's garrisoned army.

The most obvious example of an utility building in *Wack-a-Doo* is the *Chiefthan's Hut* building, this building is the central and most important structure in a capital settlement. For example it enables the progression in the game technological tree, it raises the resource maximum capacity in the capital or it makes possible the creation of a bigger number of armies.

A feature recently added was the adition of resource bubles that randomly appear floating arround the settlements map that give a bonus amount of a certain resource. To gain these bonus resources the player needs simply to click on the bubble and pop the resource out. In Figure 2.2 the floating bubbles appear hovering above seven buildings.

Wack-a-Doo provides two diferent ways for players to comunicate with each other, one of them is a mail type message system and the other is an in-game instant messaging client. These messaging systems are not the only ones that allow the players to interact with each other directly. It was also implemented a like/dislike approval system very similiar to the one present in facebook. With this approval system the *Wack-a-Doo* can express their opinions about other armies, other settlements and even other players. For example if a certain player scams another in a trade agreement the deceived player can *dislike* him on *Wack-a-Doo* and make the other players that interact with the scammer more catious. All of these social systems highlight the ammount of effort that *Wack-a-Doo* developers have invested in having a thriving and social comunity.

There is also available in *Wack-a-Doo* a ranking type system that allows the players to compare themselves to the others. The players are ranked by population number, resource score, comunity approval, defeated units, number of victories and army experience.

2.3 Business Model

The business model in *Wack-a-Doo* consists in selling to the players in-game features or bonuses. An example of feature that is currently being sold is the ability to change the name of a settlement more than once, the ability to further customize the player's profile or the warning by mail anytime one of the players settlements or armies is under attack. The bonuses available for the players to buy can be divided in two categories, the military bonuses and the economic bonuses.

The military bonuses provide a certain game advantage to the player who uses them regarding the military aspect of the game. Examples of military bonuses can be faster movement of armies around the *Wack-a-Doo's* map or quicker training of a limited number of military units. The economic bonuses allow the players to temporarily increase the strength of their economy and to accelerate their technological progression. Examples of economic bonuses can be boost in resource mining, increased size of the construction queue or even faster construction of buildings.

As it was said previously in this section, these bonus cost money and has such the players to use them need to spend the in-game currency resource, which is golden frogs. This resource is not minable directly through buildings however players can win some golden frogs as a reward for completing certain tasks in the game. However these tasks are not a reliable way to gain golden frogs because the rewards are very small and because they become increasingly difficult as time passes. The best and most reliable way to gain golden frogs is to buy them in the store available inside the game. In this store the players can acquire golden frogs in exchange for *5DLab* credits.

The company *5DLab,* is responsible not only for the development of *Wack-a-Doo* but also for other browser and mobile games or applications. In order to raise company loyalty they decided to adopt a company wide system of credits. This means that players who buy credits from 5DLab can use them in any of the *5DLab* products.

There are packages containing a number of *5DLab* credits that can range from a small package with only 13 credits to a big one 4000 credits. The price police of *5DLab* regarding their credits incentivises players to buy the packages with the bigger amounts of credits by diminishing the price per credit as the number of credits in the package gets bigger. For example in the smallest package with 13 credits the price per credit is 0.0076€, in a median package with 250 credits the price per credit is 0.0040€ and in the biggest package with 4000 credits the price per credit is 0.025€. These credits can be bought directly from a *5DLab* payment processor partner and accept various payments methods such as the normal Visa or MasterCard but also SMS texting.

One of the most critical aspects for *5DLab* regarding *Wack-a-Doo* is trying to maintain or even increase a sustainable player base that keeps the game alive and fun to play. Without a decently sized player base a game so oriented to player interaction, as is *Wack-a-Doo,* can't be successful. In order to achieve this, the developers need in a first step to raise the player base to a considerable level, and then in a second step to maintain it and try to compensate for the inevitable number of players that stop playing. The way to do this is by raising the game visibility through advertising [3].

In order to increase the visibility of *Wack-a-Doo, 5DLab* runs advertising campaigns using Google's *AdWords* services and Facebook's ad services. These campaigns are created and defined cautiously in order to target the group of people that most likely would like to play *Wack-a-Doo.*

Besides using Facebook's ad services *5DLab* also maintains a page inside Facebook promoting *Wack-a-Doo* where they give updates about the development of the game, the upcoming features or even publish trailer videos of the game.

A very good way that *5DLab* also uses to raise the visibility of *Wack-a-Doo* is to be present in internet forums dedicated to browser games and participate in polls or rankings for games in this genre. Recently a team of developers from *5Dlab* also went to Gamescom which is the largest gaming event in Europe and second in the world to advertise their game [13].

A clever way to raise the visibility of the game is to put the players themselves advertising the game. In order to incentivise this behaviour the players integrated into the game a referral system that benefits not only the player inviting is friends into the *Wack-a-Doo* world but also brings advantages to the newcomer. A player to access the referral system needs to control a region through a fortress, when that happens he can use the fortress interface menu to invite his friends using their email. When a friend eventually accepts an invite his first settlement is created in the same region where the fortress from the inviting player is situated. This brings advantages for the new player because he knows that at least he will have an ally to protect him in his home region and also brings advantages to the inviting player because he can have a region with more players in and as a result collect resource taxes from more players.

3 Related Works

3.1 Browser Games

The growth in popularity of web browser games caused an increased interest in the developer community to this game genre. This lead to a rise in competition where the developers are always trying to innovate and figure ways to create not only more engaging games but also more profitable games [3]. Studying the behavior of their player base is starting to be increasingly common in the industry due to its success in increasing the game popularity and profitability [9] [14].

The dependency that games have on obtaining knowledge from the player behaviour lead to the creation of many fairly recent companies that provide solutions which enable the game developers to perform all kinds of in-depth analysis of the data collected by the game. An example of companies that provide those services is Honeytracks[1] or GameAnalytics[2].

Unfortunately the amount of studies publicly known about knowledge extraction in MMOGs is limited, mainly due to confidentiality issues. This confidentiality is explained because for a MMOG developer company such publication will deprive them from a slightly advantage factor very

[1] www.honeytracks.com

[2] www.gameanalytics.com

important in the competitive market that it actuates on. The available knowledge for free-to-play games is more comprehensive, however it generally comes from articles or blog posts [15].

One of the techniques used for analyzing player data is calculating metrics that serve as Key Performance Indicators (KPIs); examples of KPIs in MMOGs could be the session times, the churn rate or if applicable even tutorial completion information [16] [17]. Because of the more mature state of the field of web analytics there are some techniques that were adapted from this field and used in the context of MMOGs. Examples of this adaptation could be conversion rates analysis, user acquisition cost analysis or cohort analysis [18].

3.2 Traditional Computer Games

There are also advanced data mining techniques used on more traditional computer games that have objectives such as behaviour prediction [19] [20] [21], classification of user behaviour [22] [23], and retention modeling [18] that are interesting and can potentially provide very useful information to the MMOGs developers [15].

There are also a rising number of studies that are dedicated to study the profile and identifying the traits and motivations of the persons who play online games [24] [25] [26]. Some of the motivating and engaging factors identified on these kinds of studies are sense of advancement in the game, competition, relationship, teamwork or socializing.

There are studies that also analyse how the gender and age of the players influences their motivation, for example it has been observed that the relationship component of the game is more important for women than for men [27]. All these insights can effectively help the developers shape their game according to the target audience of their choice.

3.3 Social Games

The rise of social networks gave a very important incentive in the investigation of the social gaming field [5] [28]. In this genre of games the design needs to be embedded as a subsystem of a larger system that is the social network it is hosted on. This symbiosis is beneficial for the developers of social games because the social aspect gives a positive impact on the usage of the game over time by easily raising the number of friends that are also involved in the game [29].

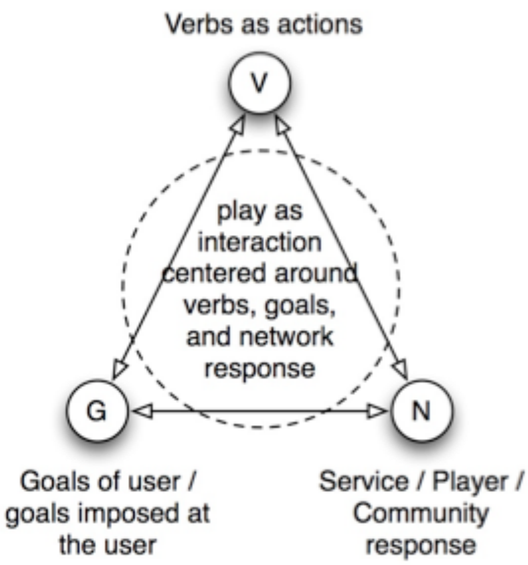

Figure 3.1 - Verbs - Goals - Network play model. Source [28]

The social network environment makes the game play revolve around a feedback loop of player actions that try to accomplish goals and are given feedback through the same network. The triangle in Figure 3.1 describes the Verbs-Goals-Network play model for social games and gives a sense of the dynamic between the fundamental elements to be taken into account when designing a social network game [28]. In the Verbs-Goals-Network play model the goals of the user are dynamic and can be created as a result from community response. When the goals are accomplished there is also a feedback going to the social community.

Although *Wack-a-Doo* is not integrated into a social network it has already a community platform that enables the developers of *5DLab* to use many of the conclusions from the studies in social gaming to their advantage.

4 System for Collection and Processing of Player Evolution

4.1 Study Goal

The sources of revenue in the game used for this study are the selling of extra features to the players and the selling of bonus that give a temporary game advantage to the players who buy them. There is then, of course, a huge incentive for the developers to create a game that incentivises the consumption of those products. To improve their game experience it was decided to analyse the player's behaviour.

The company behind *Wack-a-Doo* divides the players in categories according to a hierarchy of states and for those categories makes an analogy to an inverted pyramid. This classification is made in such a way that when a player advances to the next conversion state he can never posteriorly be classified at a lower position in this hierarchy or in other words return to a higher level in the pyramid. The objective of the developers is to have as many players as possible in the lower parts of the pyramid.

The inverted pyramid in Figure 4.1 denotes the hierarchy of all the conversion states found in this study. The players are classified and advance in this inverted pyramid according to well-defined objectives. These states are self-explanatory and intuitive to understand. Another characteristic of this classification is that a player can fit the requisites for many states but they are always classified within the lowest possible level of the pyramid.

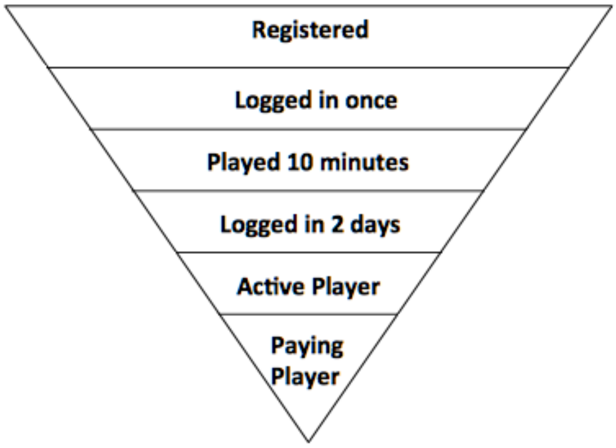

Figure 4.1 - Hierarchic scheme of Wack-a-Doo players conversion state

Conversion state analysis is, to the *Wack-a-Doo* developers, a very important way to measure almost directly the performance of their game and its financial stability. However there are other indicators that they suggest as important for those objectives such as time played or of course the very common churn ratio. The objective in this study is to analyse the player behaviour and extract knowledge that can help developers improve or understand these indicators. The indicators used in this study will be primarily the time played and the conversion state.

There could be multiple ways to make the analysis of the player behaviour, the approach chosen for this study was to extract as much knowledge using as source only the first playing days of each player. For this study it was decided to use the information from the first 3 days of each player and make a short-term prediction. It was also used the 10th day as the prediction day of the chosen game indicators.

Given the long term characteristic of the games in this genre [3], the study focus on only some days can be confusing. The fact is that the analysis is not only on some days of the game history, what happens is that not all days of the player's game history are used but only the first three days of its history. This short period is relevant due to three factors. The first relies on the fact that it is in the first days of playing that information about all the users experimenting the game for the first time can be found. The second point is that the first days of each player are very important in defining its behaviour for the rest of the game. For example, in *Wack-a-Doo* after the 10th day and not counting inactive users 81% of the players maintain their conversion status. The last and third factor is because *Wack-a-Doo* has a high short-term churn ratio and that makes it harder to create a significant dataset where there would be information about the players behaviour for a longer period of time.

This short-term analysis can also be very useful to help the companies quickly decide on the effectiveness of a marketing campaign. For example, they can study the players that registered from a certain marketing campaign in their first few days, and predict whether or not there is a good percentage of them that are going to bring any added value to the game.

4.2 Data Extraction

For this study to be relevant the data needs be extracted automatically from the deployed Wack-a-Doo system. As such, the first step in this study was to analyse how *Wack-a-Doo* was developed and decide what should be the best and least intrusive way to obtain access to the necessary data. For various reasons such as the need to support lots of concurrent playing session *5DLab* developed quite a sophisticated infrastructure behind the *Wack-a-Doo* game.

For the *Wack-a-Doo* browser game *5Dlab* has the need to maintain two databases:

1. A database responsible for storing game related information such as the position and units of each player's armies or the messages and trades exchanged between players;
2. A more sensitive database responsible for storing personal information about the players such as their login information.

For the purposes of this study only the data presented in the first database is needed because it is the one that effectively stores the player's behaviour. The second step was then to analyse what information did *Wack-a-Doo* store about the player's behaviour in the selected database by not only studying the game very carefully but also by inspecting all the data tables one by one. Fortunately the developers had a very well designed database and stored various player attributes useful for this study.

The approach taken while selecting the attributes to collect from the database was a comprehensive one. This means that if there was the slightest impression that a certain attribute helped describe the player's behaviour it was collected. The final list contained 33 attributes for each player that varied from the gameplay related such as the number of armies or the number of settlements founded, to the more social oriented such as number of messages received/sent or number of likes received/sent, to the obligatory more financial oriented such as amount of money spent or kind of bonus used.

One of the characteristics of the *Wack-a-Doo* database is that it does not have a time related attribute associated with its data. In practice only the current state of the player is registered and there is no information about when certain changes in the state of the player occurred. Because of this it would not be possible to study the evolution of the player using *Wack-a-Doo's* database alone.

To surpass this shortcoming a system for collection and preprocessing of player status named SysCoPPE was developed. This system is capable of providing the necessary knowledge about evolution of the *Wack-a-Doo* players. With SysCoPPE it is easy to collect all of the selected attributes from the *Wack-a-Doo* database at regular intervals. For the reasons of this study it was decided to opt for an interval of 24 hours between each data collection.

4.3 Description

The main programing language used for *Wack-a-Doo's* development is the dynamic and object-oriented Ruby programing language. SysCoPPE was then also developed in ruby in order to enable a better synergy with *Wack-a-Doo.*

The developers created also for the various entities and events in the game objects that represented them. Using those classes was deemed the easiest way to access the data needed for this study for three reasons, first because they provide an easy interface to access the games data, second because they give access to some methods that allowed to do some useful operations on the data and third because they provide a safeguard from updates that change the way the data from the players is stored. However because those classes were not able to provide some desired information and because of the lack of documentation on those classes there were various cases where SQL queries were made directly to the database.

Testing the stability and correctness of this system is very important for two reasons, first because *Wack-a-Doo* is a live game that is constantly running and second because it is important to ensure the correctness of the collected data. For this the test servers of *Wack-a-Doo* were used, these test servers are replicas from the online servers but without any players actually playing the game and only containing some data made purposely for testing.

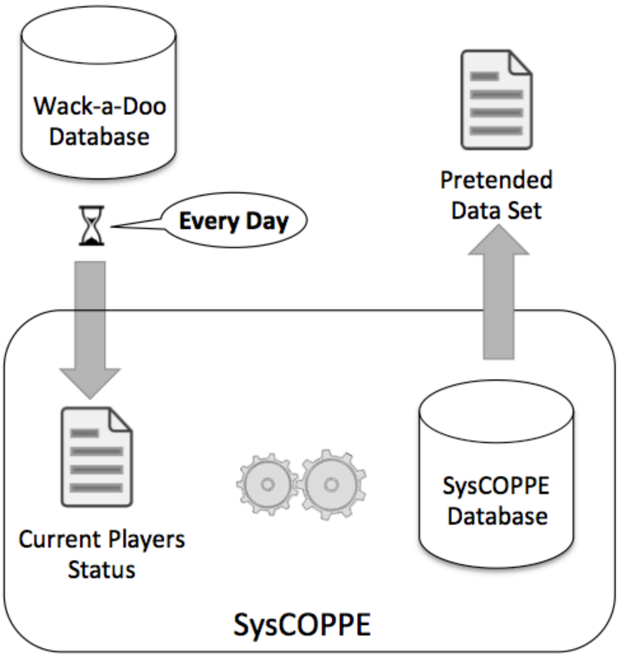

Figure 4.2 - SysCoPPE architecture

The scheme in Figure 4.2 highlights how does SysCoPPE work and helps understand the way SysCoPPE is structured. As it can be seen in the figure SysCoPPE runs every day in order to take a snapshot of the current status of the player's data and record in a file easy to process and read by SysCoPPE. In the collected data there is then the need to run some pre-processing and cleaning tasks before the data being inserted in the SysCoPPE database.

Another of the features of SysCoPPE is the ability to aid the generation of datasets according to user specified requirements. The user can specify for example, the time interval, the filtering of inactive users or of course the target feature of the pretended dataset.

The SysCoPPE system is divided in four different components, the *Wack-a-Doo* data extractor, the extracted data processor, the database and finally the dataset creation interface.

4.4 Components

4.4.1 Wack-a-Doo Data Extractor

The *Wack-a-Doo* data extractor needs to run every day at the same time in order to keep the time distance between all of the snapshots collected constant. After collecting the data from all of the players the information is stored in a single text Comma-Separated Values (CSV) file. This file contains then the data from the players regarding the collected day.

The CSV format was adopted because it was easy to read, which facilitates the discovery of errors or incoherencies and easy to process using ruby's CSV library. This file format might not be indicated when there are a very large number of players because this would make the CSV file grow too big. In the case of *Wack-a-Doo* the collection of the daily behaviour of all its players never amounted to a CSV file bigger than one megabyte, giving a still big margin to grow in case the player base started increasing.

The data about the player's behaviour is scattered throughout the *Wack-a-Doo's* database and as stated before to retrieve it there are two methods employed. The first is by using the already made ruby classes, for example to select all the non-inactive players it is used the method *non_deleted* from the *Character* class that represents the player. The second is by using directly SQL queries, for example to know which players are leader of an alliance. This last case happens because information about who is the alliance leader is not present in the *Character* class.

In some cases, there was also information that was not directly stored by *Wack-a-Doo* such as which paid extras has the player purchased. Because of the way *Wack-a-Doo* was developed, it is not possible to know if the player has bought a certain paid feature or if it was acquired using bonuses that the game gives to the players. Another problem is that it is not possible to know what the player bought in the store but only how much credits he spent. However except for the stone, wood and fur resource bonus all the other paid extras had different prices. Using this difference in prices and the credit spent history it was possible to record what paid extras the players had effectively bought.

4.4.2 Extracted Data Processor

The data processor is needed because when the data comes from *Wack-a-Doo* into SysCoPPE it still has some incoherencies and problems that need to be treated and also some processing that needs to be done before going into the SysCoPPE database.

During this phase lots of checks are made in order to ensure that all the data is without any problems. When one problem is identified the system tries to fix it. For example, sometimes players use stolen credit cards to make purchases and when that happens their registered money gross will revert back but the conversion state will not. In this case SysCoPPE will update the conversion state to the value it was before. Another example more related to the developer's *Wack-a-Doo* implementation could be the negative values for time played when the player has registered but never logged in, when this happens SysCoPPE changes the value to zero.

During this phase there are also some attributes that are calculated using the data collected from *Wack-a-Doo*. One example could be the calculation of the number of days that the player has been playing.

4.4.3 SysCoPPE Database

The DBMS chosen to use in SysCoPPE was SQLite3 version 3.7.14.1 because of its versatility, ease of use and the amount of data involved in this analysis. SQLite is a small C program with a size less than 400KB which is a zero configuration ACID compliant DBMS and implements most of the SQL standard [30].

The database architecture contains all the collected information from the players. The primary keys of the main table is a composite key made by the player identification number attribute and the number of days that has passed since the round started when that information was collected.

There were also two extra indexes created using only the player identification number attribute and the day when the sample was taken attribute.

4.4.4 Dataset Creation Interface

One of the possibilities to generate a dataset could be using the database directly and manually creating the SQL commands, however this is inadequate when the queries get too long and complex to write manually each time there is the need for a different kind of dataset or just a new updated one.

To solve this problem and make dataset creation a smoother process it was created an interface in SysCoPPE that makes this process much simpler. This interface although very simple and only accessible using the Terminal allows the user to customize the way they want their dataset created. For example it allows the user to choose the period of days that he wants to use as source for the study, what attribute he wants be the target for the prediction or what day he wants to be his prediction day.

5 Wack-a-Doo Datasets Characterization

5.1 Description

The data used for this study was collected during a period of 108 days or more or less 3 months and a half. The first time data was collected from *Wack-a-Doo* was on 23 of March and the last was on 9 of July. Because the game is still in beta this long period of time was needed to gather a sufficient number of data in order for this study to be relevant.

During this time the behavior of more than 5000 players was recorded, however there is information about the 10th play day of only around 85% of these players. As such this study is made using the information collected from 4212 different player accounts. Because a player can register with a different email or even make a second account to gain some advantage inside the game there is the possibility that some of the accounts belong to the same person.

Table 5.1 - Dataset characterization

Dataset Characteristics	*Multivariate, Time-Series*
Attribute Characteristics	*Categorical, Integer, Real*
Associated Tasks	*Classification, Regression*
Number of Instances	*4212*
Number of Attributes	*158*
Missing Values?	*No*

The data presented in Table 5.1 is a general characterization of the created datasets and the ones that are going to be used in the experiments performed in this work. It should be noted that there are two different datasets, one used when the prediction target is the conversion state on the 10th day and the other used when the prediction target is the time played on the 10th day. These two datasets differ only in their target attribute.

The datasets created do not contain any missing values and the attributes are either categorical such as the conversion state, integer such as the number of armies or real such as the time played.

5.2 Attributes

For each player in each collection there are 33 attributes that are collected or calculated which are stored in the database. Following, the different attributes about the player's behaviour are presented:

- Account:
  - Number of logins;
  - Time played;
  - Conversion state;
  - Account creation date;
  - Days since account creation and round star;
- Gameplay:
  - Number of settlements;
  - Number of fortresses;
  - Number of outposts;
  - Number of victories;
  - Number of defeats;
  - Number of armies;
  - Kills;
  - Victory ratio;

  - If has completed the tutorial.
- Social:
  - Is member of an alliance;
  - Is leader of an alliance;
  - Messages exchanged;
  - Likes sent and received;
  - Dislikes sent and received.
- Rankings:
  - Alliance score;
  - Player overall score;
  - Player resource score;
  - Player power score.
- Monetary:
  - If has bought platinum;
  - If has bought resource bonuses;
  - Credit bought and spent;
  - Current Credits;
  - Money spent;
  - Number of money transactions.

These attributes can be divided on five categories namely, account category, gameplay category, social category, rankings category and monetary category.

The account category attributes are measures that are attributed to the behavior of the player on the website, for example the number of logins in the game or the time spent playing *Wack-a-Doo.*

The gameplay category attributes are related to in-game indicators of the player performance, for example the number of controlled fortresses or the victory ratio of the players' army.

The social category attributes indicate how the player is interacting with others and how is that interaction being made, for example the number of likes received and sent or if the player is leader of an alliance.

The rankings category attributes measure the performance and progression of the players in the game relative to its community, attributes in this category can be the player power score or the player resource score.

The monetary category attributes are the ones that tell if a player is spending money on the game and how is that money being spent, the number of credits bought or credits spent is a good example of an attribute in this category.

The first step to prepare for the experiments proposed by the study goals was to create the two datasets with the 33 collected features during the first 3 played days with one having as target the conversion state and the other the time played. Because the datasets comprise of data collected during three days plus the prediction target value on the tenth, the number of features for each recorded player was not 33 plus the prediction target but 33 for each day. For example for each player the number of logins on day 1, on day 2 and on day3 were recorded.

The number of attributes collected are already considerable however, because the datasets consisted only of absolute values they did not revel directly the player evolution. To overcome this problem it was decided to do some data transformation adding new attributes.

One of the measured features where it was made some data transformation was the number of Armies (where it was calculated the difference of the number of armies between each of the 3 days). This gives a more direct measure of how fast the player is creating armies. Another example could be the information that it is stored each day telling if the player has joined or not an alliance under the form of a Boolean and it was added a new attribute saying specifically in which day he joined an alliance.

Table 5.2 - Attributes created using the number of armies for player with the id 1000

Attribute Name	***Value***
Number of Armies Day 1	0
Number of Armies Day 2	2
Number of Armies Day 3	2
Number of Armies Growth between Day 1 to 2	2
Number of Armies Growth between Day 2 to 3	0
Number of Armies Growth between Day 1 to 3	2
Difference between the Growth from Day 2 to 3 and Day 1 to 2	-2

In Table 5.2, the various attributes created using the information about the number of armies from the first three play days of a random *Wack-a-Doo* player are listed. In this case for example the indication that the growth of the number of armies is declining might indicate a certain loss of interest for the game.

It was also added some attributes that are not game related but could prove to be relevant such as the day of the week when he registered. With this new approach there were 59 new attributes added, having the new version of the two datasets a total of 158 attributes. These additions should increase the possibility for more interesting results during the knowledge extraction phase. An example could be the knowledge that a player that registers on a Friday is more probable to be playing after 10 days then a player that registers in other day of the week.

6 Experiments and Results

6.1 Methodology

According to the definition, a time series is a series of timely sequential observations:

$$X_i(t);\ [i\ =\ 2, \dots ,N;\ t\ =\ 1, \dots ,T]$$

Equation 1 - Time Series

In Equation 1 the letter *T* denotes each time in point and *i* denotes [14]the number of variables in the time series. In the case of this study the number of observations by each point in time is equal to the number of players and the variables registered are the features that are recorded for each player. Because the number of observations per point in time is bigger than one it can be said that the data registered on the SysCOPPE database allows the construction of a Multivariate Time Series (MTS).

However when creating this time series, care should be taken when choosing what values to use for the time axis. It is possible to make a time series using the collection data as the time value, but from an analysis point of view this would be highly debatable. What makes sense is to position the time axis in relation to the days since the player registered in the game. For example, with this change, when the time position number 1 is selected, only the players in identical game development stages will be seen. In the case of this study the time series would needed to be shortened to only three points in time. However for a MTS so small the traditional MTS mining techniques are inadequate [31]. Because of this constraint imposed by the study goals that were defined and to avoid confusions the collected data will not be treated as if it were a time series.

During the experiments performed it was used the Repeated Incremental Pruning to Produce Error Reduction (RIPPER) algorithm [32] for rule learning and the C4.5 algorithm [33] for decision tree generation. The particular decision to use those algorithms is justified by the fact that the objective is not to create the best prediction model possible but to extract knowledge from the data. Decision tree and rules are one of the most direct ways to accomplish this objective [34] [35].

The RIPPER algorithm is an inductive rule-based learner that aims to build a set of prepositional rules that identify classes while keeping the amount of error to a minimum. The value of the error is defined by the number of training examples that are classified incorrectly. Historically the goal of RIPPER was to handle efficiently large noisy datasets and at the same time achieve a good generalization performance [32].

The C4.5 is a decision tree algorithm where a series of if-then rules are represented under the form of a tree structure. Every node in the tree corresponds to a test of the intensity at a particular data point of the spectrum. Until a leaf node is reached it is the result of a test at one node that determines which node should be checked next. Every leaf indicates the class to be returned if it is reached [36] [33].

$$f_p(A) : A = \bigcup_{i=1}^{d} A_i$$

Equation 2 - Prediction model function

All the prediction models built in this study can be represented by the function presented in Equation 2. In this equation the letter *p* denotes the player where the prediction is being made, the letter *A* denotes the set of data from a certain collection day and *d* denotes the amount of days used as the source for the prediction. In this specific case considering the established study goals *d* will always have the value of 3.

One of the problems in the collected data is that it is extremely imbalanced. For example the percentage of players that used real money to buy anything inside the game is under 5% of the total players. However this is expected from a dataset from a web browser game [3]. This imbalance also makes somewhat harder the task of the classification algorithms.

6.2 Conversion State

As it was explained previously the conversion states are organized hierarchically. Because of this property it was decided to add new attributes to the standard dataset that translated each of the conversion states to a numeric value. The lowest value was attributed to the players with the registered state and the highest value was attributed to the paying state, those states have respectively the value 1 and 6 associated. Each step towards the top of the hierarchy adds one unit to the state value. This feature was added in order to more accurately register in the dataset the relation between each of the conversion states.

After this transformation the rule learning algorithm RIPPER and the C4.5 decision tree were used to create a prediction model. For this classification problem the decision tree model had a general accuracy of 90.2%, really close to the 91.0% general accuracy obtained with the model created by RIPPER.

Table 6.1 - RIPPER confusion matrix for conversion state classification

	Pred. Ten minutes	***Pred. Logged once***	***Pred. Active***	***Pred. Logged 2 days***	***Pred. Registered***	***Pred. Paying***
Ten minutes	1317	0	4	0	0	0
Logged once	0	911	0	0	0	0
Active	14	1	292	46	0	2
Logged 2 days	174	26	72	382	31	0
Registered	1	0	0	0	896	0
Paying	0	0	7	0	0	36

The confusion matrix associated to the model created by the RIPPER algorithm shows that there is some difficulty distinguishing between the *active player* and the *logged 2 days* conversion states. Actually, this observation also holds when analyzing the model resulting from the C4.5. To the RIPPER model the precision of the *active player* conversion state and the *logged 2 days* is respectively 77.9% and 89.25%, being the former the worst class precision of both models. The recall for these two states is, in the same order, 82.25% and 55.77%.

These results show that models created have a very big difficulty predicting the *logged 2 days* state, and the class most affected by this is the *active player* state. In fact this is one of the reasons why it can be said that the investment in adding new attributes to the dataset was worth it. When comparing the datasets without the new attributes and with the new attributes it is observed that the confusion between those two classes decreases around 20% in both of the algorithms used it is used the dataset version with the new attributes. It was interesting to verify also that the attributes that made most difference in decreasing this confusion were the ones directly measuring player evolution for example, the difference between the time played in one day and the time played in the next one.

This difficulty to distinguish between active players and players who logged in twice in a row can probably be explained by the nature of the proposed study goal. If only the information about the first 3 days is used the difference of a player who has logged in twice in a row and an active player can't be very noticeable. However the active players are crucial to the prosperity of *Wack-a-Doo* and this is why it was decided to analyse how the models behaved in distinguishing these two classes.

In the example contained in Snippet 1, *diff_time_d3d2* refers to the difference of the cumulative value of time played between day 3 and day 2. The *b_tutorial_completed_day3* and b_has_alliance_day1 are Boolean values indicating respectively if the player completed the tutorial on day 3 or not and if the player is in an alliance on day 1 or not. Another attribute referred on this example is the *round_day_on_creation*, this is a numeric attribute that indicates the number of days that passed since the last version of the game started when the player registered.

Snippet 1 - Excerpt of a tree used to predict the player's conversion state

```
conversion_state_day3 = logged_in_two_days
| num_log_ins_day3 <= 5
| | num_likes_received_day1 <= 0
| | | diff_time_d3d2 <= 1456: logged_in_two_days (267.0/6.0)
| | | diff_time_d3d2 > 1456
| | | | b_has_alliance_day1 = FALSE
| | | | | num_armies_day2 <= 1: logged_in_two_days (2.0)
| | | | | num_armies_day2 > 1
| | | | | | num_log_ins_day2 <= 1: logged_in_two_days (2.0)
| | | | | | num_log_ins_day2 > 1
| | | | | | | b_tutorial_completed_day3 = FALSE: active (4.0)
| | | | | | | b_tutorial_completed_day3 = TRUE
| | | | | | | | time_played_day1 <= 2849: logged_in_two_days
(2.0)
| | | | | | | | time_played_day1 > 2849: active (3.0)
| | | | b_has_alliance_day1 = TRUE: logged_in_two_days (4.0)
| | num_likes_received_day1 > 0
| | | round_day_on_creation <= 73: active (5.0/1.0)
| | | round_day_on_creation > 73: logged_in_two_days (6.0)
| num_log_ins_day3 > 5
| | victory_ratio_day3 <= 0.5: logged_in_two_days (8.0)
```

Tree created with the C4.5 algorithm

In Snippet 1 it is possible to see how the created decision tree tries to distinguish when a player with a *logged 2 days* state changes to the *active player* state or not. The leaf that affects the highest number of instances is the one that states that on day 3 the player played less than 1456 minutes, this is a very useful value that can be used by the developers as a reference to improve. A very interesting fact is to verify that some change in the like system made around the day number 73 of the round modified the way this system is used, the developers would be probably interested in verifying what change was made and reflect about what happened. It is also curious to observe again that the victory ratio helps classify players who will keep playing or not.

For this analysis it was decided to create a simple baseline model. This model classifies all the players according to their conversion state on their third day. When tested the baseline model has a good general accuracy with 81.0% of the players correctly classified. This surprisingly high number reveals that the conversion state of the players is relatively stable during its first days of playing.

In the RIPPER model there was only one rule that was used to classify the players belonging to the paying state. This rule can be interpreted as stating that a customer that was a paying customer in the first three days is classified as a paying customer in the 10th day. This is a trivial rule however what surprises is that it is able to classify almost 84% of all the paying customers. This high percentage supports again the idea that the game developers are doing a good job on encouraging the new players to spend money in their very first days.

6.3 Time Played

Every time a player is playing the game, *Wack-a-Doo* keeps track of the time they spend. When their session is over it updates the amount of registered playtime for the specific player. It can be said that *Wack-a-Doo* keeps a cumulative counter of each player's playtime. The metric unit used to register the time played is minutes. Because the time played is stored in a continuous scale there was the need to discretize it, in order to make RIPPER and C4.5 algorithms support a time played prediction.

The first approach to discretization tried was the creation of bins of the same range size. However there was a problem when this approach was tried, because of the big unbalance observer between the number of users in each of the bins created. The most common bin has the range $(-\infty, 63385]$ and contains 98.5% of players. This happens for two reasons the first is that it is expected for the cluster with low values to contain more players because of the users who register and never play or just try the game for one unique short session. The second is that there are some few players who spend a big amount of time playing, for example there are 18 players that are distributed between 126769 minutes and 316923 of playtime. And because of this distribution of players the range size for the bins needs to be big in order for each bin to have at least one player and to accommodate all players. Because of all this the upper boundary of the first bin is really high which again contributes to unbalance the player distribution across the bins.

The results from creating bins with the same range size were not satisfying because of the already mentioned very high difference in frequency between each of the bins. This difference makes creating relevant classification models much harder, for example a trivial model which classifies all players as belonging to the first bin would have 98.5% accuracy. A possible explanation on why this type of discretization failed for *Wack-a-Doo* players is that dividing the playtime in ranges of the same size is not a natural manner to represent the way the players behave. For example the kind of player who only comes to experiment the game and does not like and will only play a small amount of time is not represented in this type of discretization.

With this last failure to apply an automatic discretization algorithm it was decided that it would make more sense to try and create the bins manually according to an analysis of the *Wack-a-Doo* data. After this analysis it was identified 5 profiles of players according to time playing patterns found in the *Wack-a-Doo* data. It should be noted that while the specific times used in the ranges of each profile are very specific to the *Wack-a-Doo* game the theoretical notion behind each of these profiles is probably useful in every game of this genre.

Table 6.2 - Profiles for players according to their time played.

Name	***Time Range***	***Frequency***
Not Playing	[0,1]	552
Experimenting Players	(1,100]	330
Casual Players	(100,10000]	1489
Interested Players	(10000,60000]	1773
Hardcore Players	(60000,+∞)	68

The frequency of each of the classes still varies greatly from a minimum of 68 players to a maximum of 1773 however the class imbalance is not as huge as the one observed in the previous discretization attempt. But more important than that the clusters created make sense from a game point of view and should allow the extraction of much more relevant knowledge from the *Wack-a-Doo* data.

After this preprocessing of the dataset the rule learning algorithm RIPPER and the C4.5 decision tree generator algorithm were used. Both of these algorithms performed very well with the decision tree having a general accuracy with 96.84% and the rules a general accuracy of 95.75%.

Table 6.3 - C4.5 confusion matrix for time classification

	Pred. Not Playing	**Pred. Experimenting**	**Pred. Casual**	**Pred. Interested**	**Pred. Hardcore**
Not Playing	552	0	0	0	0
Experimenting	8	319	2	1	0
Casual	3	3	1482	1	0
Interested	5	2	59	1687	20
Hardcore	0	0	0	29	39

Because of the slight imbalance created by the proposed discretization, general accuracy can be a very misleading measure. However after studying both of the confusion matrixes it can be concluded that the models created are relevant. In both of the created models the worst precision results are obtained for the casual and hardcore players classes. In the particular case of the decision tree the class precision of casual players is 96.05% and of hardcore players 66.10% with a respective recall of 99.53% and 57.35%. With this analysis it can also be concluded that this model does not perform well when differentiating between hardcore players and interested players when they are hardcore players.

According to the generated tree model an experimenting player is one that plays on the first day but not on the second. This is actually the behaviour that is attributed to a player experimenting the game. This is a positive sign that can be interpreted as indicating that the discretization of the time played was well done.

In the model created by the C4.5 algorithm, for the distinction between the interested and the hardcore players who play between 997 minutes and 21307 minutes on the third day only the time played is used. This means that for relatively low values of time played the major deciding factor is the time played probably because the low playtime does not allow much of the other features to have any relevant information.

The *completed tutorial* attribute in the decision tree model is also a factor distinguishing between the interested and hardcore players. This could mean that a more appealing game tutorial could bring more players to classes related to a higher playtime. Other important factor distinguishing between these two classes is the success in battle. For the players who played more than 21307 minutes, the ones who had a number of defeats bigger than zero in the first day are classified as interested players and the ones who had more victories than defeats are classified as hardcore players. This last finding points to the effect that succeeding in the battlefield has to the players, one measure to exploit this could be the inclusion of non-playing characters relatively easy to defeat.

Snippet 2 - Example of the positive effect of the social features

```
(diff_time_d3d1 >= 10302) and (time_played_day1 >= 6680)
and (overlal_score_day3 >= 315) and (diff_ovrscore_d2d1 >= 75)
and (day_join_alliance <= 1) => result=hardcore players (15/0)
```

Rule created with the RIPPER algorithm

In the above example *diff_time_d3d1* and *diff_ovrscore_d2d1* represent respectively the difference of the cumulative values of time played and overall score between days 3 and 2 and between days 2 and 1. If the player has joined an alliance the *day_join_alliance* attribute has a numerical value indicating in which of the three days that happened.

With the creation of these two models it was also found that the social aspect of the game has a positive effect on the time played. The players who have completed the game tutorial, sent messages or likes to other players or joined an alliance are always classified as one of the classes who spent more time playing. This points to the conclusion that efforts related to making the social features of the game more appealing and to incentivise the players to invite their friends to the game have a relevant positive effect.

7 Discussion

A brief introduction containing the motivation, the scope, the objectives and the resource methodology was presented in Chapter 1. All this information is important because it gives the reader an overview of the work developed.

The contextualization and explanation of the game that is used as the target for this study is contained in Chapter 2. This chapter gives an insight of how the game works and what kind of data can be recorded about the player's behaviour.

The presentation of the state of the art in the fields where this work is most related to are presented on Chapter 3. This chapter gives an idea of what has been tried before and the limitations in the related published works.

The description of the proposed system to collect and process the evolution of player's is presented in Chapter 4 and gives a technical overview of how SysCoPPE is designed. This overview can be a useful inspiration to other researchers looking to do work in the field of web browser game mining.

The characteristics and the description of the dataset created using SysCoPPE according to the goal of this work are fully presented in Chapter 5. The description of the dataset was needed to enable a better understanding of the results of the performed experiments.

The methodology followed during the experiments performed were carefully described in the beginning of Chapter 6. The details about the experiments and results presented also in Chapter 6 highlight the kind of knowledge that can be extracted in a web browser game.

7.1 Relevant Work

Part of the work presented in this dissertation was disseminated in the following publication presented in the 10th International Conference DCAI - International Symposium on Distributed Computing and Artificial Intelligence:

> Alves J., Rebelo A., Catalão F., Marreiros G., Analide C., Novais P., Neves J., Simulating the behavior of teams of Affective Agents using Robocode, Distributed Computing and Artificial Intelligence, 10th International Conference, Sigeru Omatu, José Neves, Juan M. Corchado, Juan F. De Paz Santana, Sara Rodríguez González (eds), Springer - Series Advances in Intelligent and Soft Computing, Vol 217, pp 79-86, 2013.

Another paper which focus on the work developed for this thesis ha been submitted and accepted in the 8th International Conference on Knowledge, Information and Creativity Support Systems:

> Alves J., Neves J.; Lange S., Riedmiller M., Knowledge Extraction from the Behaviour of Players in a Web Browser Game, Knowledge, Information and Creativity Support Systems, 8th International Conference, Springer Series: Advances in Intelligent Systems and Computing, 2013.

7.2 Conclusions

With the rise of smart phones with web access and the growing popularity of social platforms the future of browser games looks very promising. This creates an increased awareness in the game developers community, which in turns raises the competition in the browser games market. The usage of data mining techniques to extract knowledge about the player's behaviour can be the deciding factor to achieve financial success.

In this work it was described SysCoPPE, a system that was created to collect and preprocess data from a browser game. This type of solution is very versatile and can be easily adapted to a browser game created by a different developer. Future work on SysCoPPE should pass by adding the capability to make an analysis of predefined features; this analysis would be updated every time the SysCoPPE is also updated.

In addition to SysCoPPE it was presented a case study using the data from a browser game currently in the market. In the experiments section it was suggested five profiles according to playtime patterns that were identified in the *Wack-a-Doo* data. These profiles can be a useful contribution for browser game developers that may help them classify players from other games of this genre.

The work developed had its focus on making very short-term predictions of features that the developers considered relevant. It is expected that the results obtained will make a contribution to comprehend, which characteristics of the player's performance inside the game are important to be taken in consideration, and which parts of the game should be object of a more careful elaboration.

An example of an important player characteristic with impact on the player's behaviour is the *victory ratio*. Another example could be the discovery that social related game features, such as alliances or a like system, have also a positive effect inside the game. These are all considerations that a developer of web browser games will find useful when creating one game in this genre.

Although it has been concluded that there was useful knowledge extracted using the current approach based only on the player behaviour in their first days, it should be noted that to identify complex behaviour patterns it is necessary to use longer time spans. As such for future work it may be interesting and insightful to use a longer time span and using it to identify complex patterns that can only be observed in a more prolonged analysis.

The major source of difficulties that emerged during the work developed for this series came from the fact that this work was made in cooperation with a company. The difference from work development in a more controlled and academically friendly environment to the work developed in a real business company was significant.

Naturally because *Wack-a-Doo* was a live web site and because it was very important for *5DLab* to maintain its stability there was the need to test carefully and explain everything that needed to have access to *Wack-a-Doo.* This is of course a very reasonable and prudent measure that does not occur in a more academic environment.

One of the first challenges faced was trying to understand how was the *Wack-a-Doo* game backend designed and where was the best place to get the data that was needed and how should it be gotten. It was also needed a long time just to try to understand and gain a general sense of what data was stored in a database as complex as the one in *Wack-a-Doo.*

The documentation available in the *Wack-a-Doo* was somewhat scarce and made it harder for an outsider to grasp some of the aspects of this game development. There were lots of features and solutions that were discovered purely on a try and error basis.

In order for this study to be successful there was the need to study and understand in a more profound way how was the gameplay in *Wack-a-Doo*, what ways did a player have to interact with each other, what extras could be purchased in the in-game shop and how could they be useful, etc. However, because the game was still in its first stages of development when this study started the translation of *Wack-a-Doo* from German to English was still a work in progress with a long way to go.

Researching for related works in the area of web browser games was also very time consuming due to the scarcity of public studies made in this area. It was only when contact was taken with the first book focused exclusively on game metrics and game analytics published in 2013 that the reason for this difficulty was found. There is a thriving business behind game design consulting and in a market so fiercely competitive as the web browser game one is, a more insightful knowledge on how to develop this kind of games can be truly crucial for the success of a company.

The companies that perform web browser game consulting are hired to study a web browser game and to identify changes and improvements that could be made in order to raise the profitability and the sustainability of the game. With some improvements and changes the work developed during the creation of this thesis could be used as the basis to form a spinoff that would work in this web browser game consulting market.

8 References

[1] Cisco Systems Inc., Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2010–2015, February 1, 2011.

[2] B. Kirman, S. Lawson, and C. Linehan, "Gaming On and Off the Social Graph: The Social Structure of Facebook Games," in *Computational Science and Engineering, 2009 International Conference on.*, 2009, vol. 4, pp. 627-632.

[3] J. Vanhatupa, "Business model of long-term browser-based games - Income without game packages," in *Next Generation Web Services Practices, 7th International Conference on*, 2011, pp. 369-372.

[4] A. Taivalsaari, T. Mikkonen, M. Anttonen, and A. Salminen, "he Death of Binary Software: End User Software Moves to the Web," in *Creating, Connecting and Collaborating through Computing (C5), 2011 Ninth International Conference on*, 2011, pp. 17-23.

[5] Rossi Luca, "Playing your network : gaming in social network sites," *Culture*, 2009.

[6] Järvinen Aki, "Workshop : Game Design for Social Networks," *Design*, pp. 224-225, 2009.

[7] Dean Takahashi. (2011, December) Venture Beat. [Online]. http://venturebeat.com/2011/12/12/zynga-history/ Last consulted in September 2013

[8] Tracey Lien. (2012, September) Polygon. [Online]. http://www.polygon.com/gaming/2012/9/5/3290747/farmville-2 Last consulted in

September 2013

[9] L. Mellon, Applying metrics driven development to MMO costs and risks, 2009.

[10] Bridget Somekh, *Action Research: a methodology for change and development*, 1st ed.: Open University Press, 2005.

[11] 5DLab. 5DLab - Augmented Worlds. [Online]. www.5dlab.com Last consulted in September 2013

[12] C. Klimmt, H. Schmid, and J. Orthmann, "Exploring the enjoyment of playing browser games," in *Cyberpsychology and Behavior.*, 2009, vol. 12, pp. 231-234.

[13] Lange Sascha. (2013, August) Wack-a-Doo Blog. [Online]. https://wack-a-doo.de/en/announcements/57 Last consulted in September 2013

[14] J. Alves, J. Neves, S. Lange, and M. Riedmiller, "Knowledge Extraction from the Behaviour of Players in a Web Browser Game," in *Knowledge, Information and Creativity Support Systems, 8th International Conference, Springer Series: Advances in Intelligent Systems and Computing*, Krakow, 2013.

[15] Anders Drachen, Christian Thurau, Julian Togelius, Georgios Yannakakis, and Christian Bauckhage, "Game Data Mining," in *Game Analytics.*: Springer London, 2013, pp. 205-253.

[16] T. V. Fields, "Game Industry Metrics Terminology and Analytics Case Study," in *Game Analytics*, Springer London, Ed., 2013, pp. 53-71.

[17] J. Kawale, A. Pal, and J. Srivastava, "Churn Prediction in MMORPGs: A Social Influence Based Approach," in *2009 International Conference on Computational Science and Engineering*, vol. 4, 2009, pp. 423-428.

[18] T. Fields and B. Cotton, *Social game design: Monetization methods and mechanics.*:

Waltham: Morgan Kauffman Publishers, 2011.

[19] T. Mahlmann, A. Drachen, J. Togelius, A. Canossa, and G.N. Yannakakis, "Predicting player behavior in Tomb Raider: Underworld," in *Computational Intelligence and Games, 2010 IEEE Symposium on*, 2010, pp. 178-185.

[20] K.J.S.K.J., Srivastava, J. Shim, "Sequence Alignment Based Analysis of Player Behavior in Massively Multiplayer Online Role-Playing Games (MMORPGs)," in *ata Mining Workshops ICDMW 2010 IEEE International Conference*, 2010, pp. 997-1004.

[21] B.G. Weber, M. John, and M. Mateas,., 2011, pp. 1701-1706.

[22] K. Anagnostou and M. Maragoudakis, "Data Mining for Player Modeling in Videogames," in *2009 13th Panhellenic Conference on Informatics*, 2009, pp. 30-34.

[23] A. Drachen, R. Sifa, C. Bauckhage, and C. Thurau, "Guns, swords and data: Clustering of player behavior in computer games in the wild," in *Computational Intelligence and Games, 2012 IEEE Conference on*, 2012, pp. 163-170.

[24] Jieun Lee, Mira Lee, and In Choi, "Social Network Games Uncovered: Motivations and Their Attitudinal and Behavioral Outcomes," *Cyberpsychology behavior and social networking*, 2012.

[25] Tom Hainey, Thomas Connolly, Mark Stansfield, and Elizabeth Boyle, "The differences in motivations of online game players and offline game players: A combined analysis of three studies at higher education level," *Computers & Education*, vol. 57, pp. 2197-2211, 2011.

[26] Francesca Dauriat et al., "Motivations to play specifically predict excessive involvement in massively multiplayer online role-playing games: evidence from an online survey," *European Addiction Research}*, vol. 17, pp. 185-189, 2011.

[27] Nick Yee, "Motivations of Play in Online Games Nick Yee, Department of Communication,"

CyberPsychology & Behavior, vol. 9, no. 6, pp. 1-10, January 2007.

[28] Aki Järvinen, "Game design for social networks: interaction design for playful dispositions," in *Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games Sandbox 09*, Bernard Perron and Mark Wolf, Eds.: ACM Press, 2009, pp. 95-102.

[29] Nazir Atif, Raza Saqib, and Chuah Chen-nee, "Unveiling facebook: a measurement study of social network based applications," *Methodology*, pp. 43-56, 2008.

[30] (2012, June) SQLite. [Online]. http://www.sqlite.org/different.html Last consulted in September 2013

[31] S. Swift, J. Kok, and X. Liu, "Learning short multivariate time series models through evolutionary and sparse matrix computation," in *Natural Computing.*, 2009, vol. 5, pp. 387-426.

[32] W.W. Cohen, "Fast Effective Rule Induction," in *Proceedings of the Twelfth International Conference on Machine Learning*, vol. 3, 1995, pp. 115-123.

[33] Q. Ross, *C4.5: Programs for Machine Learning*. San Mateo, California: Morgan Kaufmann Publishers, 1993.

[34] P. Langley and H.A. Simon, "Applications of machine learning and rule induction," *Communications of the ACM*, vol. 38, pp. 54-64, 1995.

[35] Sukontip Wongpun and Anongnart Srivihok, "Comparison of Attribute Selection Techniques and Algorithms in Classifying Bad Behaviors of Vocational Education Students," in *2nd IEEE International Conference on Digital Ecosystems and Technologies*, 2008, pp. 526-531.

[36] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 21, no. 3, pp. 660-674, 1991.