Universidad Autónoma de Madrid
Escuela Politécnica Superior

Trabajo Fin de Grado

DATA EXTRACTION METHODOLOGY TO IMPROVE THE GAMEPLAY EXPERIENCE IN VIDEO GAMES AND TO ANALYSE THE USER'S PROFILE BEHAVIOUR AND ITS EVOLUTION

Grado en Ingeniería Informática

Rafael Vindel Amor
junio 2014

DATA EXTRACTION METHODOLOGY TO IMPROVE THE GAMEPLAY EXPERIENCE IN VIDEO GAMES AND TO ANALYSE THE USER'S PROFILE BEHAVIOUR AND ITS EVOLUTION

AUTOR: Rafael Vindel Amor
TUTOR: Héctor Menéndez Benito
PONENTE: David Camacho Fernández

Applied Intelligence and Data Analysis
Dpto. de Ingeniera Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
junio 2014

Resumen

Resumen

Los videojuegos son la principal fuente de entretenimiento en nuestros días y la que más beneficios genera en la industria de los contenidos audiovisuales.

Para las compañías que desarrollan videojuegos es muy importante saber el perfil de los usuarios que adquieren sus videojuegos para, posteriormente, desarrollar contenido específico para dichos usuarios. Los usuarios, por otra parte, exigen un mayor realismo en la inteligencia artificial de los enemigos y un mayor nivel de dificultad, todo ello unido a una elevada capacidad de personalización del modo de juego. En el ámbito de la inteligencia artificial y de la personalización existen videojuegos con mecánicas dinámicas que hacen que cada partida tenga una experiencia única.

En este trabajo se pretende, mediante el diseño y programación de un videojuego, abordar dichos problemas para conseguir una metodología que sirva de ayuda en este campo. Para ello, se recopilarán estadísticas de uso del videojuego que serán analizadas para determinar las mejoras realizables dentro del propio videojuego. Con estas estadísticas, se realizará un análisis de los perfiles de los usuarios presentes en el videojuego con el objetivo de saber los distintos tipos de usuarios que hay en función de su nivel de habilidad y en función de su estilo de juego. Todo esto se realizará con el objetivo de proporcionar una experiencia más gratificante de cara al usuario. De esta forma, se podrán crear mecánicas dinámicas de juego en función de las acciones que hayan realizado cada uno de los usuarios.

Finalmente, este trabajo aprovecha esta información para aportar posibles soluciones para mejorar la jugabilidad del propio videojuego y para clasificar a los usuarios en función de la evolución de su perfil utilizando los resultados extraídos del análisis realizado. Para realizar el análisis propuesto se han empleado técnicas de Data Mining no supervisado y series temporales.

Palabras Clave

videojuego, perfil de usuario, evolución de usuario, series temporales, clustering

Abstract

Video games are the main source of entertainment these days and the most profitable industry that generates audiovisual contents.

On the one hand, video game companies consider important to understand their users profile in order to develop especific content for them. On the other hand, current users require some artificial intelligence improvements and gameplay customization to enrich the game experience. In the field of artificial intelligence and gameplay customization, there are several video games with dynamic gameplay mechanics that make each game looks like a new and unique experience.

The goal of this work, through the design and programming of a video game, is to get a methodology that helps in this field. To do this, usage video game statistics are collected to be analyzed in order to determine the achievable improvements within the video game itself. With these statistics, an analysis of the users profiles present in the video game is performed in order to know the different types of users depending on their skill level and their play style. All this is done with the aim of providing a more rewarding experience for the user. Thus, it can be created dynamic gameplay mechanics based on the actions of each user.

Finally, this work uses this information to provide possible solutions to improve the gameplay of the video game and to classify the users according to their profile evolution by using the results extracted from the previous analysis. To perform the proposed analysis, unsupervised data mining and time-series techniques have been employed.

Keywords

video game, user profile, user evolution, time series, clustering

Agradecimientos

Quería dar las gracias a mis padres, ya que han sido los que me han apoyado durante estos cuatro años y han hecho que llegue hasta donde me encuentro ahora mismo. A mi familia por animarme durante todo este tiempo. A ti, Patricia, ya que has sido un apoyo enorme en este tiempo y me has dado las fuerzas y las ganas necesarias para seguir adelante. A mis amigos por disfrutar de los buenos momentos que hemos pasado juntos. Dar las gracias también a todos aquellos que han colaborado con este proyecto jugando al videojuego. A los compañeros del departamento y; en especial, a Héctor, ya que, a parte de guiarme, ha sido una persona de la que he tenido el placer de aprender muchas cosas. Muchas gracias a todos ya que sin vosotros esto no habría sido posible.

Contents

List of Figures

1 Introduction

A video game is an electronic game in which one or more persons interact, by a controller, with a device provided with video images. This electronic device known as a platform can be: a computer, an arcade machine, a game console or a portable device. Currently, video games are the main source of entertainment among the current population. The development of artificial intelligence techniques is a very studied area by large companies to try to give their video games a more realistic experience. Similarly, the customization of the user experience is a task that has become especially important nowadays. That is why the development of techniques to help improve any of the two points mentioned above have a great impact on the video games industry.

Due to that each user typically has a different style of play compared to the rest of the users (more offensive, more defensive or a more casual style of play) but with certain common patterns in all of them, and their expectations about the video game vary greatly to the expectations of the others, the data analysis process becomes very complex due to the large number of variables presented in the system. The goal of this work is to group users according to the similarity in their styles of play and analyze the configuration of the game to improve the experience of the users.

Therefore, in this first model a video game is realized. This video game allows the extraction of the statistics needed to analyze the users profiles and to determine whether the configuration of the video game is optimal. Thus, a database with the set of actions to be analyzed of each user is created.

To analyze the data extracted from the users, it has been chosen to use Data Mining algorithms [1]. Due to it is preferred performing a blind analysis of the extracted data, it has been decided to use clustering techniques [2]. The clustering techniques allow to group the users by their similarities. In order to measure the evolution of the users, techniques of time series analysis [3] which include the evolutionary factor within the clusters of the clustering algorithms have been introduced.

There are a great number and a great variety of problems which arise in this work: the constraints under which the project has been realized have been taken into account. On the one hand, it is a very large project, so it is necessary to establish goals and to delimit it at different levels to cover it. On the other hand, it is necessary that the developed video game is entertaining enough for the users in order to extract the needed statistics for the realization of the analysis. The number of users can also be a problem, the project has a sufficient number of users, but in order to obtain more reliable results, it is recommended to increase the number of users present in the video game. If the number of users increases, it may be necessary to perform a new architecture on the server side in order to behold that number of users. This architecture requires a better configuration and a considerable improvement in the security system. Once the statistics are extracted, it is necessary to provide a set of metrics that are meaningful enough in order to obtain accurate information to build the user profiles. It is necessary to provide the correct parameters to the algorithms used to obtain reliable results. Finally, note that during the project possible extensions and improvements have been taken into account, so it would be possible to realize them within a short period of time.

The rest of this work is organized as follows. The Chapter 2 introduces the state of the art as well as work related to the project and current tools used in the industry. The Chapter 3 describes the software architecture and the operation of the model. In the Chapter 4 the tests performed and the results obtained are shown. Finally, the Chapter 5 includes the analysis of the results in the form of conclusions, as well as a section on future work.

1.1 Motivation

Video games are the main source of entertainment these days. The video games propose an environment where many users develop their skills. The development of artificial intelligence techniques allow users to evolve to adapt to the new conditions and achieve the ultimate goal of each of the video games. The possibility of obtaining a procedure to significantly improve the gaming experience of users is, without a doubt, the biggest motivation that has made this project an option to take.

Obtaining an environment where the enemies are based on an artificial intelligence that always poses a challenge for the users is the main goal of this first study. The users will also develop their skills in order to overcome the artificial intelligence of the enemies. It is also pretended to make a custom video game experience based on the skills of the users.

1.2 Objectives

The objectives of this work are:

- **Design and programming of a video game**. Development of a video game that provides an environment where a certain number of users play in order to extract their gameplay statistics.
- **Usage statistics extraction**. Extraction of the actions performed by the users within the video game.
- **Video game analysis**. Analysis of usage data extracted from the users and identification of the users behavior within the video game.
- **Users profile analysis**. Identification of the different types of users based on their playing style and their skill level proposed solving the challenges in the video game.
- **Video game improvements**. Analysis of the possible improvements to be made within the video game once performed the previous analysis. These improvements are ranged from changes in video game settings to changes in the artificial intelligence of the enemies.

- **Users clustering**. Obtention of what types of users are the most probable in all the experiments performed and the main groups of users based on their skill levels.

2
Introduction

This chapter pretends to introduce a general overview around the videogames industry, its history and the current approaches around this market. Besides, in order to cover the analytical techniques used in this work, it also introduces some methologies used in behavioural analysis, specially those focused on the gamer profile. Finally, it gives some notions about the Data Mining techniques used in this work, in this case, specially focused on time series and clustering.

2.1 Video Games Industry

Video games are one of the most profitable current business in our society [4]. This business started with the first graphic interfaces for computers which allows users to play different games in their PCs [5]. The main entertaiment concepts about video games have made them a situable environment for different genres.

The different advances in technological devices such as graphic cards, RAM memory, ROM devices, processors, and, specially, the Internet have supposed also an important improvement to the video games market. The first generations was based on offline 2D environments while the current versions are focused on multi-player and online 3D environments.

The games evolution has produced different games categories. The most studied genres are [6]:

- **Action**. The action genre consists of two subgenres: first-person and third-person games. This genre has usually a cinematic perspective where the player is represented by a character. This character has to overcome obstacles using tools or items, or fighting with different enemies.
- **Role-playing**. Role-playing games (RPG) are usually associated with fantasy. These games are usually focused on the story in a literary way.
- **Simulation**. The simulation genre includes video games that simulate sports, flying, driving, the dynamics of towns, cities, small communities, etc.
- **Strategy**. There are two main subgenres which belong to strategy games: Real Time Strategy (RTS) and Turn-Based Strategy (TBS). Usually, the former games are those

where the player has to decide its strategy during the game play and adapt it (e.g., wargames); the latter games are similar but the player decides the strategy step by step (e.g., chessgames).

One of the main problems developers face is how to create an enjoyable game [7]. This field needs to consider several factors of different sources. Next sections explain how different works are focused on the game analysis in order to improve the game-play experience [8, 9, 10]

Video games have an important social repercussion [11], too. One of the main problem in video games is the aggresive behaviour which is usually produced by violent video games in children and young adults [12]. However, the game experience can be also use to learn about different aspects of human beings such as risk taking, strategy formulation and execution, moral or ethic decisions, etc. [13]. Also, games can be used to help in different environments. These perspectives are usually called gamification [14].

The main idea around gamification is to create different environments, based on games, oriented to learning activities or cooperation between users. It usually obtains better results than tradicional methodologies and has been successfully applied in different fields such as:

- **Learning**. Lecturers generate a ludification of the learning process creating cooperative or competitive activities for the students [15].
- **Marketing**. Several research lines follows how gamification can improve the costumer interest in different business and products [16].
- **Military**. Usually games are used to train soldiers in different aspects, such as strategies or missions, simulating a real world [17].
- **Health**. Different methodologies of gamification are focused on health such as exercise games [18].

Finally, it is also important to understand how the user adapts himself to the video game during the game-play experience. This information provides a general profile of the gamer which allows the developers to adjust the game to a concrete user profile, for example, according to their age, gender, play time, etc. [19]. Moreover, the video game can also be used to define the player behaviour, extracting data from different movements and decissions chosen during the game-play [20].

2.2 Gamers and their interactions

Games are usually focused on players. Several games provide a virtual representation of the gamer inside the environment created. This representation can be totally or partially physical (e.g., a thrid-person [6] or first person [6] game) or unphysical (e.g., building games or puzzles [6]). The former provide a whole individual inside the game which is controled by the gamer, the latter provides a set of tools which the gamer uses in order to complete the game, such as, for example, move the different pieces of a puzzle.

Depending on the game, the decissions made by the user are focused on solving a deterministic pattern (such as a fixed Artificial Intelligence or puzzles) or they have a random component which makes the gamers change their strategy (e.g. RTS games).

Gamers usually learns within the game through a reinforcement learning methodology [21]. Usually, they receive different rewards (such as coins, level improvement, new abilities, new puzzles, etc.) or punishments (life reduction, less money, reduce abilities, etc) depending on their behaviour during the gameplay.

Gamers are also the main focus for the industry. From the marketing point of view, they are the clients and it is important to satisfy their spectatives. However, it is difficult to measure how the game experience is attractive or not to a determine user or group. Next section explains how several works have tried to face this problem analysing the user behaviour inside the game and generating user profiles in order to categorize the users [22, 23, 24].

2.3 Perspectives for Game and Gamer Analysis

As was mentioned above, it is important to understand both: how the game is working and how the user is involved in the game environment. Here, we present several points of view and motivations for game and gamer analysis.

2.3.1 Motivations for Game Analysis

Game analysis is usually based on the analysis of the game itself or its effects in the users from different perspectives such as psycology or marketing. Here, we collect some works which explains different motivations for Game Analysis.

First, it is important to understand how the game interacts with the devices. This information is extreamly important specially for online games where the company usually provides a server and the number of user is not limited. Henderson [8] provides a good example in this field. He studies the hardware implication in online games. In his work, he provides a server for Half-life and studies the server behaviour specially focused on the payload and how it affects to the users behaviour in the game play experience. They discover that players are usually toletant to delay.

Other of the main analysis perspective is Marketing analysis in order to generate a profitable market. The work of Chung and Grimes [25] is a good example of how the information has to be considered. In their work, they analyse how online games affect to children from a marketing perspective. Authors study different websites exploring the privacy issues in order to discover that the information provided by the users is usually ignored even when this information is really valuable for marketing.

In this field it is also important to evaluate the enjoyable experience. For example, Vorderer et al. [26] study the factors which produces an enjoyment experience in video games. They focused their work, specially, in collaborative and competitive games, discovering that competitive environments obtain better results entertaining players.

From a different point of view it is also important to understant how the user is attracted by the game. For example. Yee [10] studies the player motivation of different users to play online games. Author tries to understand, using the game as a starting point, those relationships that are important between the user and the game. Finally, he discriminates three main motivations: achievement, social and inmersion.

Following this previous work, he also study an important psycological phenomena of current video games. In [9], Yee analyses how current games has become as a second work for several users and how these games have been designed, in order to achieve this goal. The work is focused on describing patterns in those platforms designed to train gamers to become better game workers. This shows social trends between play and work.

Finally, games also migth produce some psychological effects in the gamers. The most studies effects are addiction and aggresive behaviors derivate from games. Wood [27] focuses its analysis on the former (game addiction). In order to evaluate previous methodologies to evaluate game addiction, he proposes four case studies demonstrating how old techniques where not totally accurate and concluding that the main factors behind game addiction are: ineffective time management skills and a response to other problems users are escaping from, instead of addictive properties of the game itself. Griffiths [28] analyses the literature around the latter: aggresive behavior produced by violence in video games. Several studies blame video games for producing aggresive behaviour in children, however, author proves that previous studies have methodological problems and they only measure aggresive consecuences in short-term.

2.3.2 Motivations for Gamer Analysis

Previous section shows the importance of analysing the games and its influence in the users. This section tries to motivate the importance of analysing the gamer, its effects on the game and also the data that can be extracted about their profile using information about the game-play.

The first relevant step in the player analysis is how to choose the relevant data which is going to be analysed. In order to choose the most relevant factors it is important to define the goal of the analysis. Moura et al. [24], for example, propose a methodology to extract data from games and analyse it in order to understant player behaviour within the game. Due to the high amount of data, they propose different visualization techniques to identify patterns and player behaviour. Other good approach is to focus the analysis on the player interaction. Marsh et al. [23] propose a methodology to deal with this kind of analysis in game environments. They create a tool called ISIS to query and indentify data of interest in order to identify actions, activities, design problems, navigation problem, dificult tasks, etc. Moreover, it is also important to keep information about the player tracking. In this context, Coulton et al. [29] focused their analysis on spacio-temporal techniques to provide a 3D visualization of players movements. This methodology is applied to real users in a pacman based real game where the players are characters of this game and the system tracks their movements.

The information of the data extracted from the user is helpful to give some new features to the game. For example, Weber and Mateas [30] present an approach to opponent modeling in strategy games. They try to predict the opponent strategy before it is executed. They also try to incorporate their approach to a full game playing agent. It also can be useful to develop characters in game. This perspective has been studied by Hoorn et al. [31] who studied how to create NPC AI for video games based on players behaviour. They propose a methodology which combines record players and traditional progress functions. They apply this idea to a modern racing game.

Finally, the information extracted from the player can be applied to personalize some features to the game in order to make it more enjoyable. Hunicke and Chapman [22] faces this problem focusing their analysis on the game play experience from the difficulty point of view. They discuss how easy or frustrating games are usually worse for gamers and how static methods for difficulty selection are not enough in order to personalize the game level. They also propose a dynamic difficulty adjustment system.

2.4 Gamer behaviour Analysis using Data Mining

Data mining is defined as a toolset of knowledge extraction or pattern identification techniques applied to information sources. The most famous current techniques are supervised and unsupervised learning techniques. The former needs information about the quality of the proccess in order to identify predictive patterns in the data, while the latter identify patterns by similarity blindly.

On the one hand, supervised [1] approaches are based on the class information of the data or an special attribute which is the objective of the analysis. The algorithms are usually designed to consider this information as a feedback in order to generate a good predective model adapted to the data. There are two main approaches in these methodologies [1]: regression (which tries to estimate the relationship between the variables using statistical models) and classification (which tries to create a model to predict a target class variable). The are several algorithm categories for both fields, however, the most classical categories are Linear and Logistic for regression algorithms; and Decision Trees, Naive Bayes, Neural Networks and Support Vector Machines for classification.

On the other hand, unsupervised [1] techniques are based on a blind search within the data. The most classical methodologies are Hidden Markov models, blind signal separation and Clustering. Clustering techniques group similar data points into clusters according to a cost or objective function, which is usually minimized or maximized, making this clusters different from each other at the same time. There is a large number of clustering approaches, similar to classification, depending of the goal the algorithm should achieve. This work has been focused on partitional cluster [32]. This techinque separate the instances producing that each instances belongs to a single cluster. Depending on the clustering algorithm, there are several models focus on the solutions. These models are divided in parametric [33] or non-parametric model [34]. The former has an statistical estimator that is adapted to the data while the latter separates the data using different topologies or techniques.

The most classical partitional algorithms are K-means [33] and EM [35]. Both K-means and EM are parametrical clustering algorithms, which usually try to optimize an estimator parameters. Other classical techiques for non-parametric clustering are those based on medoids. These algorithms try to find the solution within the data [36]. They do not need the features of the search space in order to find a solution (they can deal with the information extracted from the data distances). This special property makes medoid-based algorithms a good choice for problems where the search space is not well defined, such as time series clustering. Also a medoid is a representative instance for the cluster formed, which means that the analyzer only needs to analyse the set of instance corresponding to the medoids, providing an easier human interpretation of the cluster than parametrical approaches, where the estimator has to be analyzed.

Over the last decade, clustering techniques have exploited forecasting fields, and have been specially focused on time series analysis [3]. Measuring the evolution of several time series, which are group by the clustering algorithms, these techniques can be used to predict a new series trend.

There has been used different approaches, based on Data Mining, to model the player behaviour. These approaches try to identify and group different players according to the game statistics. They have been specially applied to Robossocer simulations [37]. Due to this extraction process is initially blind, it is interesting to consider unsupervised Data Mining techniques, such as, clustering, in order to face this problem.

2.5 Video Game Development Tools

In this section we presents wome relevant tools which are useful for game developers:

- **Unity 3D**[1]. Unity is a game development framework created to develop games for different Operative Systems and devices. This environment uses different tools in order to help developers to create interactive 3D and 2D content. The main features of this engine is its facility to export the content to different platforms, the automatic integration of Artificial Intelligence and the complete market which provides different resources for the developers called Asset Store supported by a knowledge-sharing community.
- **Blender**[2]. Blender is a 3D modeling system which is usually used rigging, animation, simulation, rendering, compositing and motion tracking, even video editing and game creation. This system is free and open source. It also provides several tools to deal with different programming languages such as PythonThis suite is usually used by individuals and small studios.
- **Maya**[3]. Maya is a 3D modelation system used for animation, modeling, simulation, rendering, and compositing software. It offers a tools set which is intuitive for their users and extensible to several platforms. IT is also focused on increasing the productivity for modeling, texturing, and shader creation tasks.
- **Unreal Engine**[4]. Unreal Engine is a set of professional tools focuses on creating games for different platforms. This engine works with a rendering architecture which enables developers to achieve improve visual effects easily.
- **CryEngine**[5]. CryEngine is focused on big projects related to graphical solutions. It has been applied to the creation of games, movies, simulations, and interactive applications. It has been also focused to special devices such as Xbox 360, PlayStation 3 and PC games providing a good set of tools to develop games in these platforms.
- **Ogre3D**[6]. OGRE (Object-Oriented Graphics Rendering Engine) is focused on the creation of 3D scenes. It has been written in C++ and tries to provide an intuitive framework for developers in order to produce applications through 3D graphics accelation. It provides access to system libraries like Direct3D and OpenGL and provides an interface based on world objects and other classes.
- **Source Engine**[7]. Source Engine is focused on character animation, AI, physics, shader-based rendering, and an extensible development environment to develop games. The engine tries to optimize the hardware resources of the different devices scaling smoothly on older systems, enabling developers to reach a diverse range of gamers.
- **Anvil Engine**[8]. Anvil (also called Scimitar) is a game engine created by Ubisoft for Microsoft Windows, PlayStation 3 and Xbox 360. Prince of Persia and Assassin's Creed series have been developed using this engine.
- **SmartFoxServer**[9]. SmartFoxServer is a platform focused on multi-user applications and games with Adobe Flash/Flex/Air, Unity, HTML5, iOS, Windows Phone 8, Android, Java, Windows 8, C++, among others. SmartFoxServer provides several features and manuals with several example to explain how to use its administration tools for these kinds of games.

[1] `https://unity3d.com/`
[2] `http://www.blender.org/`
[3] `http://www.autodesk.es/products/autodesk-maya/overview`
[4] `https://www.unrealengine.com/`
[5] `http://cryengine.com/`
[6] `http://www.ogre3d.org/`
[7] `http://source.valvesoftware.com/`
[8] `http://www.desura.com/engines/scimitar`
[9] `http://smartfoxserver.com/`

3
Architecture

This chapter describes the software architecture and the data model used in this work. The main goal is to define user profiles based on gameplay data extracted from a video game. With this information, the user profile is elaborated by calculating certain metrics from the data. Thus, it is possible to determine the user features which describe its profile. Besides, the evolution of each user is analyzed using time series and users are classificated according to their evolution. Therefore, it is possible to classify the users in groups with similar learning abilities.

This analysis is useful to perform the game experience by improving; for example, the artificial intelligence of the enemies present in the video game or the mechanics and configuration of the video game. This analysis is also useful to apply matchmaking techniques through the learning abilities of each user.

The architecture is divided into several modules (see Figure 3.1). The modules of the architecture are the following:

- **Game module (1)**. 3D environment where the user controls the main character and overcomes a certain number of challenges. The user actions are stored for future retrieval.
- **Data extraction and storage module (2)**. It is responsible for connecting the video game with the database where user actions are stored. It extracts and stores in the database the actions performed by the users in the video game.
- **Data representation module (3)**. It plots the data stored in the database. This module helps developers to represent user actions and decisions during the gameplay.
- **Basic user profile module (4)**. It uses the stored data to calculate certain metrics in order to quantify the characteristics and abilities of the users. With these metrics, it defines the user profile. The user profile could be represented by spider graphs.
- **Profile evolution module (5)**. It analizes the user profile evolution during a certain number of games. To do this, it will use time series and clustering techniques to group users with similar learning abilities.

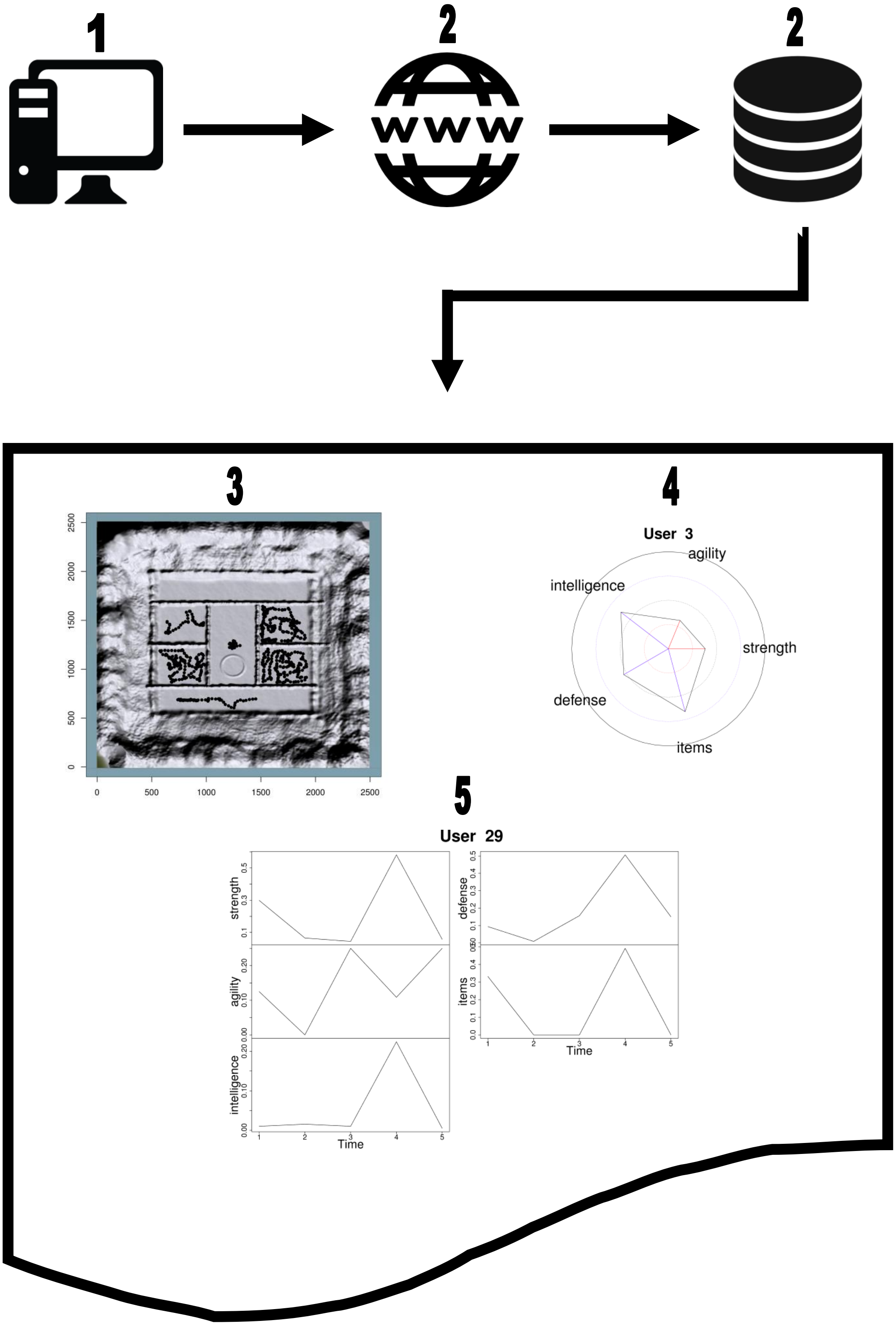


Figure 3.1: Architecture schema

Figure 3.2: Video game environment.

3.1 Game Module

Dream is a 3D computer video game corresponding to the Action-RPG[][1] genre. This type of video game incorporates elements of traditional RPGs, such as statistical character development; attached to real-time combat, characteristic of the Hack and Slash[][2] video games. This video game has a very similar gameplay dynamic of games like: Skyrim[3], Dark Souls 2[4] or The Witcher 3[5].

Dream has been developed using Unity3D[6] and C# as programming language. Unity3D is a video game engine platform that incorporates a development environment which enables to program the functionality of the video game. For 3D video game elements, 3D modeling and animation programs were used (such as: Autodesk 3ds Max[7], Autodesk Maya[8] or Blender[9]).

In Dream, the user controls the main character and its aim is to overcome a certain number of quests that consist of defeating a certain number of key enemies. In order to achieve this goal, the user has a set of skills that, combined with its basic attack, will face enemies that will stand in his way.

The character has a series of control bars that will show its status during the game (see Figure 3.2). The control bars are the following:

- **Health bar**. It indicates the amount of life that the character has. When life downs to zero, the character is defeated and the player has to start from the begining. The health is regenerated steadily over time.
- **Mana bar**. It indicates the amount of magical energy that the character has for its skills. The skills consumes a certain amount of mana. If the character does not have that amount

[1]http://en.wikipedia.org/wiki/Action_role-playing_game
[2]http://en.wikipedia.org/wiki/Hack_and_slash
[3]http://www.elderscrolls.com/skyrim
[4]http://www.darksoulsii.com/es/
[5]http://thewitcher.com/witcher/
[6]http://unity3d.com/
[7]http://www.autodesk.es/products/autodesk-3ds-max-design/overview
[8]http://www.autodesk.es/products/autodesk-maya/overview
[9]http://www.blender.org/

of mana, the character is not able to use that skill. The mana is regenerated steadily over time.

- **Stamina bar**. It indicates the number of attacks that the character is able to do. Each weapon consumes a certain amount of energy every time that the character attacks. When it has not enough energy, the character is not able to attack. The energy is regenerated steadily over time.
- **Experience bar**. It indicates the total experience of the character. The experience is obtained by defeating the present enemies in the video game. Upon reaching the maximum experience, the character will level up and will earn points to increase his attributes. After going up a level, the experience bar will be empty again.

As mentioned above, the user is able to increase certain attributes of his character when he reaches a new level. These attributes improve certain abilities of the character. The available attributes are the following: strength, agility, magic, physical resist, magic resist, vitality, mana and luck.

The character also has an inventory where he stores the items that he collects during the game. Items may be from potions that instantly increase the health, the mana, or the energy of the character; up to pieces of armor and weapons. The pieces of armor and the weapons that the character collects during the game may be equipped. The character's equipment will consist of the following items: helmet, chest armor, pants, boots and a weapon.

The character has four skills, each one with different effects. The first skill inflicts high damage to a single enemy. The second skill inflicts moderate damage to enemies within a certain distance of the character. The third skill changes mana points for health points. The fourth skill changes health points for mana points.

The enemies, like the character, have a number of attributes and control bars according to its difficulty. They also have a type of artificial intelligence. In the game, an enemy has one of the following artificial intelligences:

- **Haunter**. This type of artificial intelligence is the most basic of all the enemies present in the video game. The enemy stands in its original location until it is attacked by the character or until the character enters into its action range. In the event of any of the above situations, the enemy begins to chase the character without stopping, across all the map. When the enemy is in the proper range, he attacks the character.
- **Shout**. This type of intelligence is an extension of the functionality of the previous artificial intelligence. The enemy, when his life downs below 25% of its maximum value, emits a shout that alerts the enemies within a certain range. These enemies chases and attacks the character. The shout can only be activated once per enemy.
- **Group**. This type of intelligence is an extension of the functionality of the previous artificial intelligence. The enemies form groups. When an enemy is attacked by the character or the character enters in its range of action, all the enemies in that group chase and attack the player.

The game map is divided into six distinct areas. These areas can be seen graphically in Figure 3.3. A brief description of the features of each one of these areas is shown below:

- **Tutorial**. In this area the character is guided along a path and the basic controls of the video game are explained. This part of the game may be skipped and the character will go directly to the next area, called rest area.

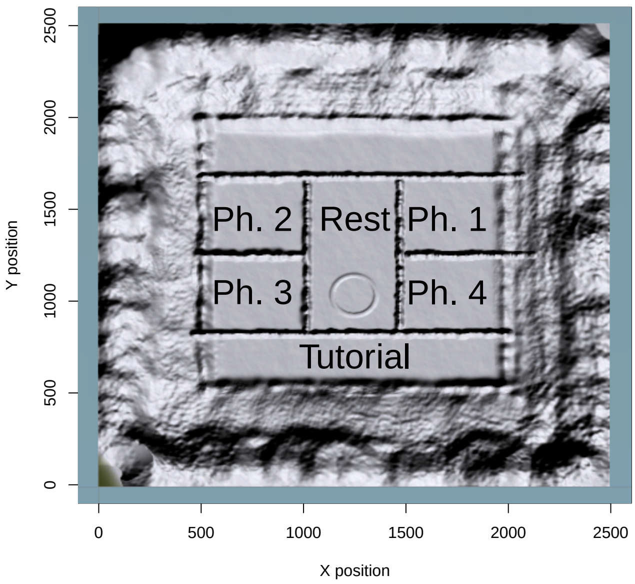


Figure 3.3: Video game areas.

- **Rest area**. This is the area where the character begins if he decides to skip the tutorial. The area consists of four doors, connecting each one to the four areas of the video game. Each one of these areas contains different types of enemies with different types of artificial intelligence. The enemies also have different attributes. Also, each zone contains a boss. To move between each zone, the character must kill the final boss. When the final boss is defeated, the next area is unlocked.
- **Phase one**. This area contains the weakest enemies in the video game. The artificial intelligence of these enemies is based on the haunter type. The levels of the enemies in this area varies between 1 and 3.
- **Phase two**. In this area, the enemies with levels between 2 and 4 are found. The artificial intelligence of these enemies is based on the haunter type, except for level 4 enemies, which are based on shout artificial intelligence.
- **Phase three**. In this area, the enemies with levels between 4 and 6 are found. The artificial intelligence of these enemies is based on the shout type.
- **Phase four**. In this area, the enemies with levels between 6 and 7 are found. The artificial intelligence of these enemies is based on the group type. There will also be a number of enemies who have an artificial intelligence based on the haunter type. When the final boss of this area is defeated, the video game is completed.

3.2 Data Extraction and Storage Module

This module is responsible for extracting usage statistics of each character and user, storing these statistics in the video game database. This module is presented in both the game and the server (where the database is located). Therefore, it is the communication bridge between these two services.

The module is divided into two parts: the collection of statistics within the video game and the transmission of the data for storage in the video game database. These two parts are detailed below:

- **Data extraction**. This process is produced during the gameplay and is transparent to the user. It means that the user knows nothing about the process of data extraction. Due to there are different types of statistics that are extracted from the video game, not all of them are collected in the same way. The movement of the character, for example, is obtained every 2 seconds. However, the increase of the character attribute is only obtained when the character increments that attribute. Therefore, one can distinguish between synchronous and asynchronous statistics. Some of the statistics that are collected are: tracking of the character, tracking of the enemies, attacks realized by the character, attacks realized by enemies and initial positions of the objects and enemies, among others. Statistics collected are sent directly to the server, they are not stored in any temporary files, so an Internet connection is required for proper operation of the game.
- **Data storage**. This process is carried out when the communication between the video game and the web server where the database is located is established. The game establishes the communication with the web server by using HTTP protocol. It sends the character statistics using POST parameters. In this way, they are sent to the server where the statistics are processed and stored in the database. The statistics are sent after being collected. The database that is used to store the statistics is a MySQL database. This database has a number of tables equal to the number of statistics collected by the video game.

The extraction and storage methodology that has been raised is used by the most popular current video games that collect usage information from its users. Some examples os those video games are: World of Warcraft[10] and Call of Duty[11].

3.3 Data Representation Module

This module is responsible for representing the statistics stored in the video game database. To do this, R[12] is used. A series of queries will be realized on the video game database to obtain the statistics that can be represented. After obtaining the statistics, they are processed and plotted using R over the video game map. Thus, the statistics may be viewed within the video game environment itself, providing a full view of the user behaviour. Additionally, it is also useful to do graphs where can be observed certain relevant statistics about the functionality of the video game.

It is possible to represent the movement made by the players and enemies into the total number of games played. With this information, it can be analyzed the behavior of players and which parts of the areas are the most explored. Similarly, it can be analyzed the movement made by the enemies which is useful to think about variations in their movement covering the spaces which are the most explored.

It can also represent the position of the items present in the video game (such as potions and equipment items) compared to the users movement. With that representation, it can be analyzed if the users movement is based on obtaining the most part of the items available on the map. It is expected that the users will move around the items instead of avoiding them.

[10] `http://eu.battle.net/wow/es/`
[11] `http://www.callofduty.com/`
[12] `http://www.r-project.org/`

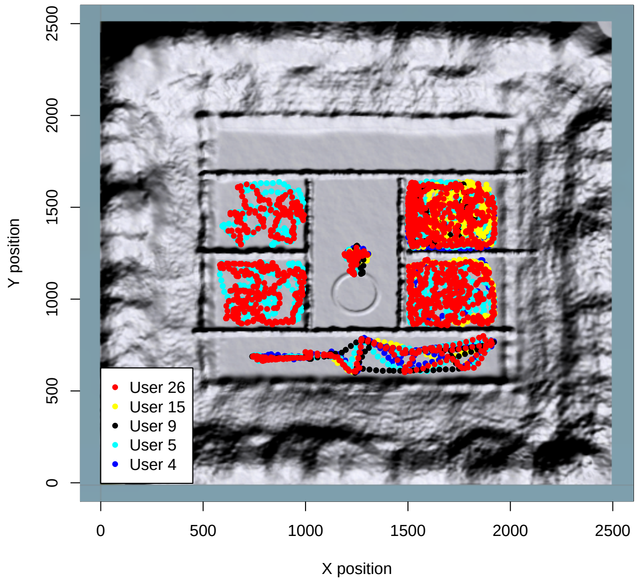


Figure 3.4: Basic representation example.

Likewise, it can represent the movement made by the players compared to the starting position of the enemies present in the area. It is expected that, contrary to the case of items, players will avoid fighting with the most part of the enemies.

After obtaining the position where the players where killed, it knows if the difficulty level in each one of the video game areas responds properly. In lower areas is expected that the number of deaths is less than in the higher areas due to that, in higher areas the enemies are stronger than in the lower areas.

Together with the above, the usage of potions is an example of the difficulty of fighting against the enemies. If a player uses a greater number of potions in the lower areas than in the higher areas, it may indicate that the difficulty does not work according to the expected. That is why the representation of the positions where the players use potions will be an important fact to consider. The type of potion used is also important.

The increase of the players attributes is a statistic to consider.If there is an attribute that predominates over the others, it is possible that may have some additional advantage and, consequently, it may affect the evolution of the video game. That is why it will be analyzed the number of attributes increased over the total number of attributes.

The damage done by the players attacks compared to the maximun enemies life indicates if the game responds well to higher level characters or, on the other hand, the higher level characters are stronger than the higher enemies

In Figure 3.4, it can be seen an example of one possible representation of the data stored in the video game database.

3.4 Basic User Profile Module

This module is responsible for calculating and representing profiles of each one of the players present in the video game. To do this, it is employed usage statistics drawn by the previous module, which are stored in the video game database. The profile consists of a certain number of metrics. Each one of these metrics quantifies a certain characteristic of the character in order to classify them into groups with similar profiles. All these metrics are normalized and take a value between [0,1], with 0 being the lowest possible value and with 1 being the highest value for a particular metric.

The metrics which are used for the construction of the profiles are the following:

- **Strength**. This metric indicates whether a player **(pl)** has an aggressive profile. A player with a high value will be the one that prefers physical attacks **(PA)** instead of magical attacks. This kind of user usually increases attributes like strength **(SA)** and agility **(AA)** instead of the other attributes **(tot(A))**. Also, the metric considers the number of enemies that the player has killed **(EK)** over the total of enemies present in the game **(ET)**. Finally, the level reached by the player **(LP)** also influences the final value achieved in that metric. The value is defined by the following formula:

$$st_{pl} = \frac{1}{4}\left(\frac{EK_{pl}}{ET} + \frac{SA_{pl} + AA_{pl}}{tot(A)_{pl}} + \frac{max(PA_{pl})}{max(PA_{all(pl)})} + \frac{max(LP_{pl})}{max(LP_{all(pl)})}\right) \tag{3.1}$$

- **Agility**. This metric indicates whether the player **(pl)** solves the puzzles easily and if he has chosen to kill only the minimum number of enemies to overcome each of the phases. A player with a high value will be the one who solves each different phase in the shortest possible time **(PT(i))**, compared against the times of the rest of players **(all(pl))**. Moreover, the metric also considers the number of enemies killed **(EK)** to complete each phase. The lower the number of enemies killed, the higher the value of this metric. The value is defined by the following formula:

$$ag_{pl} = \frac{1}{2}\left(\sum_{i=1}^{4} \frac{PT(i)_{pl}}{max(PT(i)_{all(pl)})} + \frac{1}{EK_{pl}}\right) \tag{3.2}$$

- **Items**. This metric indicates whether the player **(pl)** behavior is based on the acquisition **(ac())** and use **(use())** of objects that can be found in the video game or, on the contrary, he dispenses with the items to complete the game. A player with a high value is the one who collects as many objects scattered around the map and use those objects for his own benefit. The use and acquisitions of potions **(PO)** and equipment **(EQ)** increases the value of this metric. The value is defined by the following formula:

$$it_{pl} = \frac{1}{4}\left(\frac{ac(PO)_{pl}}{tot(PO)} + \frac{ac(EQ)_{pl}}{tot(EQ)} + \frac{use(PO)_{pl}}{ac(PO)_{pl}} + \frac{use(EQ)_{pl}}{ac(EQ)_{pl}}\right) \tag{3.3}$$

- **Defense**. This metric indicates whether the player **(pl)** chooses to avoid damage and confrontations. A player with a high value increments attributes that increase resistance **(RA)** over other attributes **(tot(A))**. The minimum hit **(min(H))** by an enemy is also considered. Finally, the total number of deaths that the player suffers **(DE)** and the enemies killed **(EK)** are part of the metric. The value is defined by the following formula:

$$df_{pl} = \frac{1}{4}\left(\frac{1}{EK_{pl}} + \frac{RA_{pl}}{tot(A)_{pl}} + \frac{1}{min(H)_{pl}} + \frac{1}{1 + DE_{pl}}\right) \tag{3.4}$$

- **Intelligence**. This metric indicates whether the player **(pl)** bases its offensive strategy in the use of magic skills **(MS)**, and whether he increases his level **(PL)** killing the minimun possible number of enemies **(EK/PL)**. A player with a high value uses offensive skills

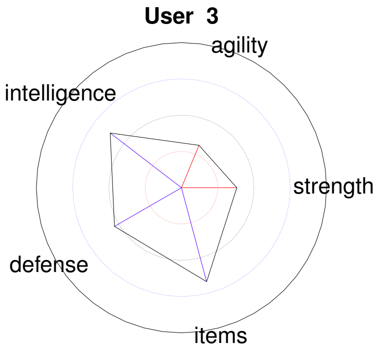


Figure 3.5: Basic profile example.

compared to other players **(all(pl))**. He also increases magical abilities **MA** over other attributes **(tot(A))**. The value is defined by the following formula:

$$in_{pl} = \frac{1}{3}\left(\frac{MS_{pl}}{max(MS_{all(pl)})} + \frac{MA_{pl}}{tot(A)_{pl}} + \frac{(EK/LP)_{pl}}{max(EK/LP)_{all(pl)}}\right) \tag{3.5}$$

Figure 3.5 shows an example of the representation of the metrics calculated of one of the video game players.

3.5 Profile Evolution Module

This module is responsible for making the necessary operations to group the users into different groups according to the similarity in the evolution of their profiles along each one of the games played. It will also be responsible for selecting a representative user for each one of the groups obtained.

This allows the analyser to consider not only the global information of each player, but also his learning abilities. Thus, the gameplay experience may be customized for each user by using the previous analysis.

The analysis steps are as follow:

- The first thing to be done is to calculate the value of each metric for each game played by all users present in the system. Thus, the module has the resulting profile of each user in each game that the user has played. This way, the module has the neccesary data in order to realized temporal series clustering and compare the users profile similarity.

- Once the user profiles are calculated, the time series statistics of the players are generated. The statistics represent the evolution of each metric of the user profile during all the games. The temporal statistics are generated by round, this allows to compare the player evolution during different rounds.

- With the time series data for each user, the time series data of the players per metric are clusterized. By using this information, the model generates a matrix with the metric cluster and the player associated. Thus, the evolution of each metric is analized individually and compared to the others. The matrix has the following configuration:

$$\begin{pmatrix} \mathcal{C}_1^{(st)} & \mathcal{C}_3^{(st)} & \mathcal{C}_4^{(st)} & \dots & \mathcal{C}_1^{(st)} \\ \mathcal{C}_2^{(in)} & \mathcal{C}_3^{(in)} & \mathcal{C}_2^{(in)} & \dots & \mathcal{C}_4^{(in)} \\ \mathcal{C}_1^{(it)} & \mathcal{C}_2^{(it)} & \mathcal{C}_1^{(it)} & \dots & \mathcal{C}_3^{(it)} \\ \mathcal{C}_3^{(df)} & \mathcal{C}_2^{(df)} & \mathcal{C}_3^{(df)} & \dots & \mathcal{C}_3^{(df)} \\ \mathcal{C}_4^{(ag)} & \mathcal{C}_3^{(ag)} & \mathcal{C}_1^{(ag)} & \dots & \mathcal{C}_4^{(ag)} \end{pmatrix} \tag{3.6}$$

In this matrix, each row represents a metric and each column represents a player. The elements of the matrix $\mathcal{C}_i^{(metric)}$ represents the assignation of each user to a determined cluster during the time-series clustering process. Thus, it can analize which one of the users are more similar according to the cluster value of each one of their metrics.

- Using the previous matrix, it generates a dissimilarity matrix around the players. This dissimilarity matrix is used to clusterized the players using a medoid based clustering algorithm.

  To calculate the dissimilarity matrix, the model calculates the dissimilarity between each user. The dissimilarity measure applied for players is the following:

$$diss(p_i, p_j) = 1 - \frac{\sum_{C_q} \delta_{C_q}^i \cdot \delta_{C_q}^j}{M} \tag{3.7}$$

  Where M is the number of metrics considered, p_i, p_j are the players to be compared, C_q represents the posible clusters per metric, and $\delta_{C_q}^i$ is the Dirichlet delta defined by:

$$\delta_{C_q}^i = \begin{cases} 1 & \text{if } p_i \in C_q \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

  Where the value is 1 if the cluster is the same in both users or 0 if is different.

- The medoid based clustering algorithm is applied with the dissimilarity matrix. It clusters the user into different groups according to their evolution similarity. It also determines the most relevant medoids of the datasets which corresponds with the most representative players.

  The medoid based clustering algorithm is chosen because a representative user of each cluster is obtained and the features of the search space are unknown.

4
Experiments

In order to evaluate the previous modules defined in Chapter 3, this chapter shows how Dream has been tested from different points of view.

In the first part, the tests are focused on the video game itself. These tests are realized before the video game is provided to the users to start playing and, consecutively, before beginning to extract the statistics of usage. The existing problems and the possible solutions to these problems are presented.

In the second part, the experimental setup of the analysis is described. This includes the total number of players and items that have been used for the study and the parameters that have been used in the algorithms. It also provides the methodology of time series clustering and user profiles clustering that have been used during all the experiment.

In the third part, an analysis of the data extracted from the video game is realized. These data indicate: the movement of the users, the movement of the enemies, the most used objects, the number of the attacks realized by the users and the enemies, the most used attribute, etc.. The analysis of these data provides certain conclusions that will help significantly to improve the gameplay experience.

Finally, the last part details the profile analysis of the groups of users obtained and the representative user of each group. A description of each extracted profile is performed. These profiles are obtained as a result of applying time-series clustering algorithms as was explained in Section 3.5.

With all this, a complete view of the main objective of this work is displayed. Additionally, it will pave the way for improvements that arise from the analysis, allowing to increase the functionality and the gameplay experience of the video game and adapt it to each user.

4.1 Testing Phase

In this first phase, it proceeds to perform a set of tests on the video game and on the architecture before it will be provided to the users, in order to begin to extract the statistics that will be used on the experiments.

The device that has been chosen to play the video game is a computer. This computer must have a minimum requirements for the video game to work correctly. These requirements extend both hardware and software level.

4.1.1 Hardware Requirements

The necessary hardware requirements are explained below. The computer must have, at least, the following components:

- A processor with, at least, two processing cores. The optimal is a quad core proccesor.
- A graphics card with 512MB of memory. The optimal is a graphics card with 1GB of memory.
- A total of 512MB of free RAM memory. The optimal free space is 1GB of RAM.
- A total of 50MB of free space in the hard drive.
- A normal internet connection with 10Mb is enough to the video game to work correctly.
- A keyboard and a mouse are required to play.

4.1.2 Software Requirements

The necessary software requirements are explained below. The computer must have, at least, the following software installed:

- A computer with a Windows XP operating system or higher is required. The game is not supported on computers with other operating systems. There is no implementation for UNIX and UNIX-derived systems.
- Due to the video game may be played in an internet browser, a plugin is required. This plugin may be downloaded from Unity[1].

4.1.3 Starting Guide

Once the user has a computer capable of supporting the implementation of the game, all he needs is to run the video game itself. The video game is an executable file that will launch all the 3D environment. Thus, no installation process is required.

Just begin, the user is asked to identify himself within the video game to be able to monitor his actions. The identification process needs a valid username with a valid password. The username is chosen by the respective user and the password is randomly generated. Such identification performs a check against the database of the video game. That is why the video game requires an internet connection. Without the identification process, the user will not be able to start playing.

Once the user has logged in, he can start playing normally.

[1] `http://unity3d.com/`

4.1.4 Server Characteristics

Here, the components required by the video game to provide a good service are detailed below. Alongside this, the possible problems that may arise in extreme conditions and the possible solutions are also detailed.

The video game must have a server where the database and the web-services are running. The former contains the player statistics while the latter is the bridge between the video game and the database. In this case, a Linux server, specifically a Debian, is chosen. The database that is used is a MySQL database. A Linux server and a MySQL database have been chosen due to both are free software.

Therefore, the system must arrange to keep active the server during the extraction of statistics because, if the server is not available, the statistics may not be stored in the database and the users will not be able to access to the video game.

The server security is also one of the most important requirements. In this case, the directions for accessing the web-services and the database are managed by different users with limited permissions.

The database must be able to allow a certain number of users to connect at the same time. These connections are concurrent. The maximum expected number of concurrent connections is 30. The MySQL databases have, by default, a number of 100 concurrent connections. This value can be changed in the configuration file. This may be a problem in the case where the number of concurrent user connections increases and the specified value in the configuration file does not cover all the users that are attempting to connect to the video game database. If our system needs more than 100 concurrent connections, one possible solution is to increase the RAM capacity of the system and other solution is to change the threading library used by the database. These libraries are system dependent and may reach 4.000 concurrent connections. If the system still needs a higher number of concurrent connections, the best solution is build an architecture with different databases and an user load balancer.

The delivery of the usage statistics, as mentioned above, depends crucially on the user internet connection and the server availability. If a user loses the internet connection while is already playing the video game, the video game will not be able to report the usage statistics and his profile analysis will be incomplete. To manage this issue, the development of temporary files containing the statistics that have not been possible to send is proposed. Thus, when the system reclaims the connection, it sends these statistics, making the data consistent and providing a complete study of the user profile.

Regarding potential bottlenecks in the system, apart from the concurrent connections to the video game database, it can be noted the number of HTTP requests to the server. Due to that the web-services are based on HTTP requests, the number of simultaneous HTTP request that are sent to the server may be a bottleneck. As shown in the case of the video game database, it may be solved with a load balancer that redirects the requests to a different server for their properly treatment. It is important that these load balancers always redirect the user requests to the same server, in order to avoid the fragmentation of the user data.

4.2 Experimental Setup

In this section, the environment on which the study is developed is properly detailed. The environment consists of the number of examples that have been treated and the procedures used to perform the user profile analysis. Among the procedures, it is also explained the specifications employed in each one, such as metric used in each algorithm or configuration values.

4.2.1 Data Collected

Regarding the number of examples used, the study is realized on a total of 36 users and a total of 107 games. This indicates that, on average, each user has made 3 games although, as will be shown later, the study is only performed on those users who have a total of 5 or more games. Thus, their profile evolution is more accurate.

The number of users may seem small but it is a fairly high number for the conditions of the study. These users have provided a fairly high number of examples which largely compensated having a small number of users.

Rearding the number of examples of each usage statistics retrieved, there is a fairly large number of them to make an intensive study. In relation to the movement made by the players, there is a total of 18.500 examples. In the case of the movement made by the enemies, there is a total of 16.700 examples. In general, as mentioned above, the number of examples is enough to perform a deep study. This study will show how these statistics can be used to improve the gameplay experience.

After specifying the number of examples of both users and data that the system will handle during the study; the methodology that is used for the user profile analysis is detailed.

4.2.2 Algorithms Parameters

As mentioned above, the experiments have been carried out with around 30 users. These users have been playing to Dream during several rounds. Each round has been a complete game which has finished when the user has complete all the phases or when he has been defeated by an enemy. All the data per round has been colected in order to extract the user profile. Also, the evolution of the metrics during each round has been used to measure the user profile evolution.

The algorithms which have been used for the analysis are a combination of time-series clustering and Partition Around Medoids[2] (PAM) clustering. For time-series clustering, TSClust[3] package has been used. TSClust is a R package which contains certain functions that allows working with time-series.

The time-series clustering process has been carried out in the following steps:

1. The times-series have been set in the search space.

   With the user data extracted, a time-serie is generated for each user and for each user metric. After doing this, all the users have, for all their metrics, time-series values that are used to clusterize their evolutions.

2. The time-series dissimilarity is calculated to generate a dissimilarity matrix.

   In this case, the metric used is the dissimilarity metric called Autocorrelation-based Dissimilarity. This measure performs the weighted Euclidean distance between the simple autocorrelation coefficients. It is defined as:

[2] `http://cran.r-project.org/web/packages/cluster/cluster.pdf`
[3] `http://cran.r-project.org/web/packages/TSclust/TSclust.pdf`

$$d(x, y) = \{(\rho_x - \rho_y)^t \Omega (\rho_x - \rho_y)\}^{\frac{1}{2}} \tag{4.1}$$

where ρ_x, ρ_y represent the autocorrelation vectors, and Ω is a inner product which depends on a geometric weights decaying factor p, as follows:

$$\Omega = \begin{pmatrix} p(1-p)^1 & 0 & \dots & 0 \\ 0 & p(1-p)^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & p(1-p)^n \end{pmatrix} \tag{4.2}$$

PAM needs the number of clusters that are use. In this case, the number of clusters that is selected is 3, because it provides the most stable solutions, according to the Within-Sum quality metric. That value has been selected because the study presents 3 groups of users according to their learning abilities: low, middle and high.

The last value that PAM needs is the decaying factor p. The p value which provides more stable solutions in this analysis is 0.075.

3. PAM is applied to group the time-series by their similarities.

   Once the time series are clusterized, the similarity matrix among the users is generated for the user profile evolution phase (see Section 3.5). The final clustering process is also carried out using PAM. This algorithm uses a dissimilarity matrix as a search space and chooses the most relevant instances in this dissimilarity matrix. All the data instances are candidate solutions for the algorithm, and the final solutions are composed by the most representative data instances, called medoids. In this case, these medoids correspond with the most representative user of each cluster.

4.3 Game Representation Analysis

This section shall carry out an analysis of the representations made with the data stored in the database. As was seen in Section 3.3, the users behavior based on the representation of their actions on the map of the video game is analyzed. Additionally, also shall carry on an quantitative data analysis.

As shown in Figure 4.1, the movevements made by the characters all over the map of the video game can be represented. It can be seen that, usually, the users prefer to move around the sides of the areas instead of venture inside of the areas. This indicates that the sides of the areas are safer because of the low number of enemies or because there are items placed in that sides. On the one hand, it can also be seen that the more explored areas are the first and second, where the characters explore practically all the area. On the other hand, the third and fourth are the less explored areas, also due to the small number of players who have been able to overcome the first and the second area.

As shown in Figure 4.2, the initial position of the enemies and the movements that they have can also be represented. It can be seen that the most of the enemies are distributed in the interior of the areas, leaving the side of these areas free. This explains the behavior of the users as discused above. In relation to the movement of these enemies, it can be seen that the first area has been the one that had the most activity. The entire area has been covered. The movement made by the enemies decreases in higher areas, due to the lower activity of players in those areas of the video game.

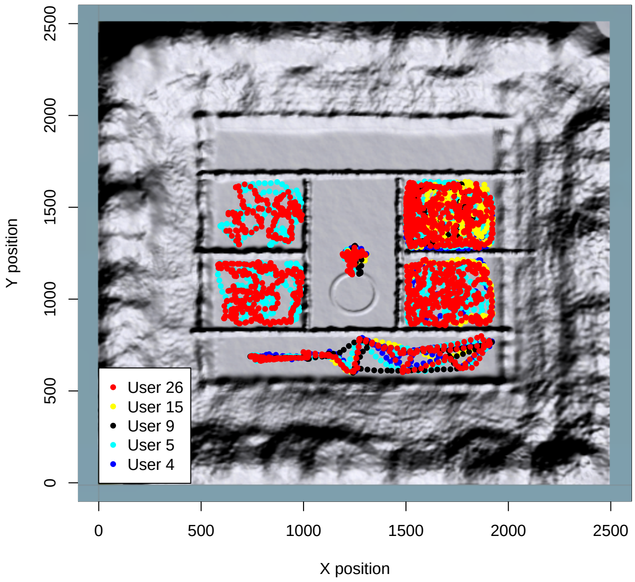


Figure 4.1: User movements.

As shown in Figure 4.3, the initial position of the objects is represented along with the movements realized by the characters all over the map of the video game. It can be observed that the most of the objects are positioned in the side of the areas, leaving the low part of the objects in the interior of these areas. It can also be observed that the players movement is aimed at obtaining such items, being clearly reflected in the movement performed by the players in the tutorial area.

Considering the three previous representations, it can be concluded that the players movement is aimed to avoid combats and to obtain all the items that they are able to. It can also be concluded that the players movement is realized in the sides of the areas, instead of in the center. In order to improve the gameplay experience, it is possible to make that the enemies patrol around the items. That will force the players to combat them if they want to obtain a determine item. With this video game version, the enemies stand into their spawning position till they are activated. It is also possible to make the enemies move randomly around certain area. This will force the users to change their rute to avoid the combats.

As shown in Figure 4.4, the position of the attacks realized by the characters can be represented along the position of the attacks realized by the total number of enemies. Both plots are very similar, indicating that the combats between the characters and enemies have been static. Once the character is facing the enemy, the combat occurs without any movement of the two opponents. It can be deduced that the combats do not offer too much choice to the user. Thus, the combats are monotonous.

As shown in Figure 4.5, the position where the enemies were eliminated and the position where the characters were eliminated can be show. The largest number of character deletions have occurred in the first zone, followed by the area of the tutorial. As for the enemies, most of the enemies in the first area have been eliminated, prompted in part by the need of the characters to gain experience to level up. The number of enemies killed by the characters descends in higher areas. This indicates that the level of the video game difficulty descends in higher areas. The difficulty should increase in higher areas so the parameters of the enemies must be changed in order to improve the gameplay experience.

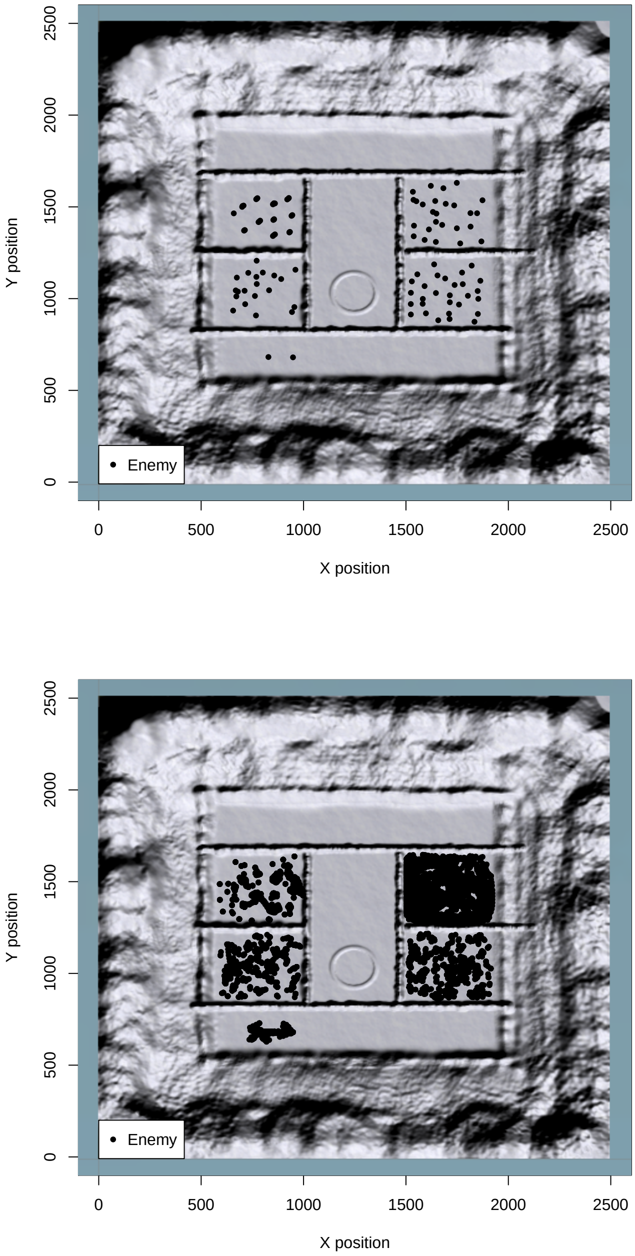
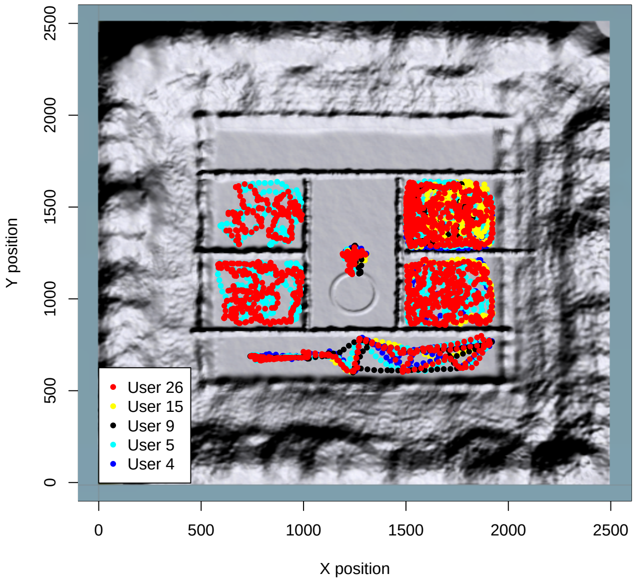


Figure 4.2: Enemies movements and initial positions.



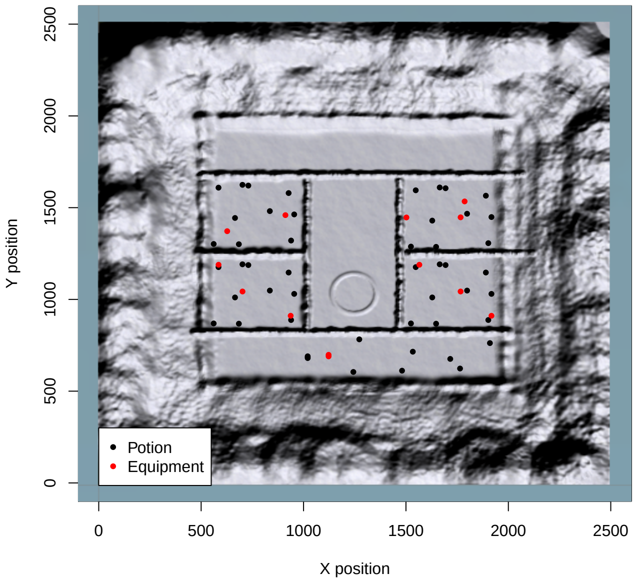


Figure 4.3: User movements and items positions.

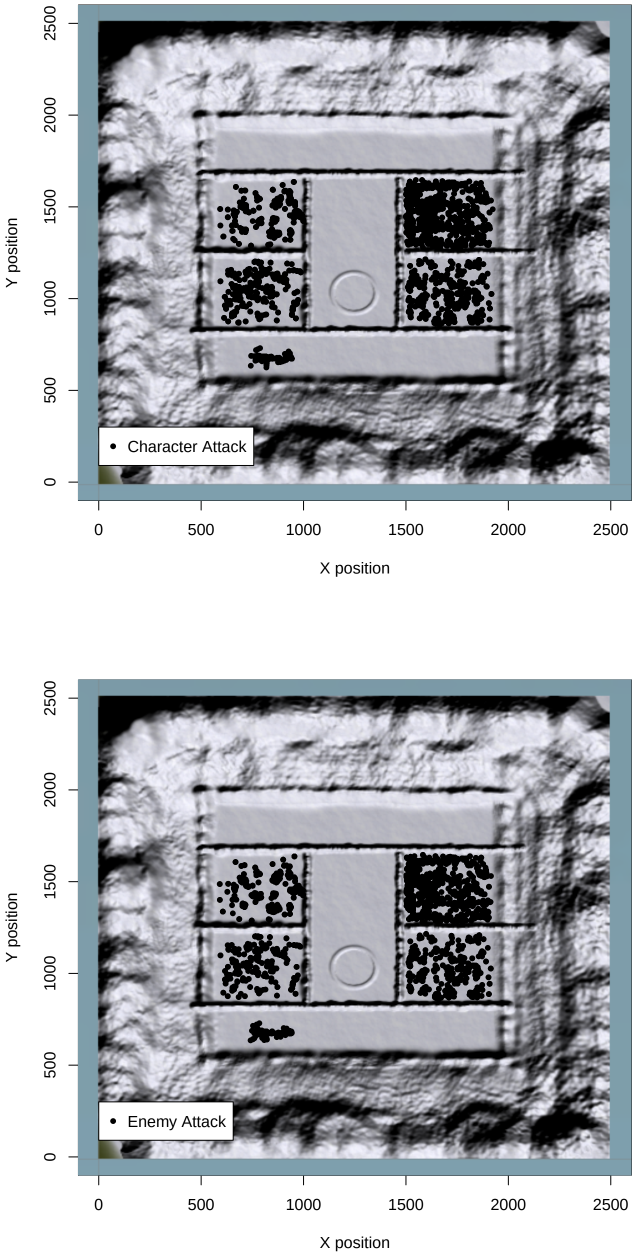


Figure 4.4: User during attacks positions and enemies during attacks positions

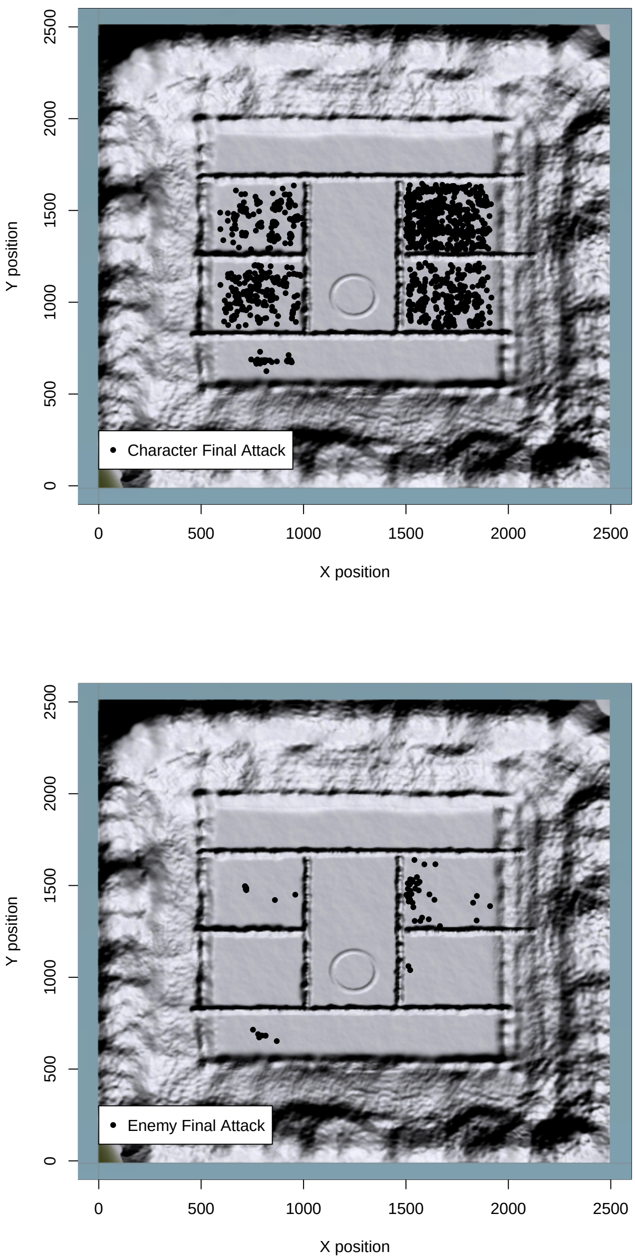


Figure 4.5: User final attack positions and enemies final attack positions

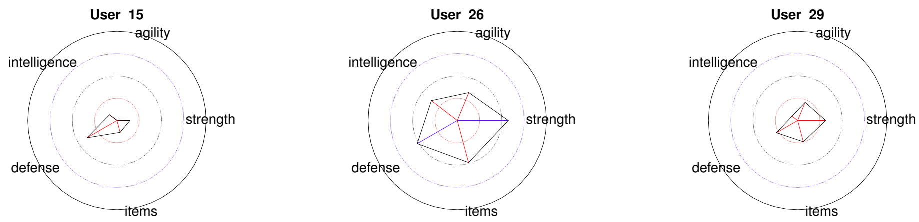


Figure 4.6: User global profile extraction based on the five metrics. These results represent the most relevant users of the profile extraction process.

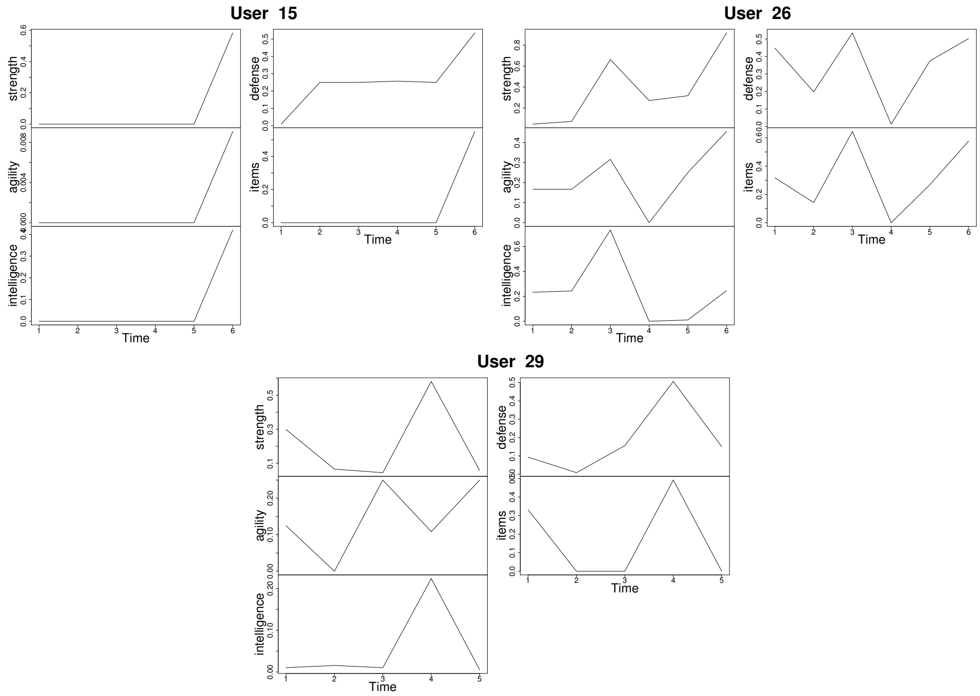


Figure 4.7: Evolution of one user in the different games that he has played. This results are used for the time-series clustering process.

4.4 User Profile Analysis

The application of the time series clustering techniques have some three relevant profiles as the chosen medoids for the final analysis. Figures 4.6 and 4.7 show the global profiles and the evolutionary profiles, respectively, of the chosen users. These results discriminate three different user behaviours and evolutions:

- **Inexperience Gamer Profile**: This profile is associated with User 15 and covers the 60% of the total users. Analysing the general profile of this kind of player (see Fig. 4.6) we can discover that the values related to strength, items and intelligent are low, the agility is the lowest and his strategy is only focused on the defense. The evolution of the player (see Fig. 4.7) shows that this kind of gamer has usually been defeated several times and he only starts to learn after several rounds. In this case, the learning improvement is remarkable for all values but the agility. It means that this kind of gamer does not deeply adapt to the game.
- **Average Gamer Profile**: This profile is associated with User 29 and covers the 20% of the total users. The average profile, in general terms (see Fig. 4.6), has balanced and

low results according to all the metrics except for the intelligence. These users tries to profound in the game-play and interact more with the environment. According to their evolution (see Fig. 4.7), it is clear that their learning process is fuzzy but there are some trends which indicates that they are trying to improve a metric per match, specially the defense, agility and items used.

- **Hardcore Gamer Profile**: This profile is associated with User 26 and covers the 20% of the total users. The hardcore gamer shows good general statistics (see Fig. 4.6) according to all metrics (the best for the three representative users). This means that these users quickly adapt to the game enviroment and have a deep game-play experience. Their evolution is a little fuzzy (see Fig. 4.7), which is usual with this kind of user because the real adaptation becames during the first or second round. These users specially focused their evolution on the strength and agility trying to optimize their game-play decisions. They modify their behaviour according to satisfy this goal, using a deep knowledge of the game, as it is shown in the high values of each metric.

With the information of the users profile, we are able to choose those users which could be more interested on the different extensions and propose different options to each profile, for example, helpful objects and equipments for average users, new habilities which makes the game easier for inexperienced gamers and more complex phases for hardcore users (which can provide and interesting challenge for them). Also, given a new user, we are able to identy the most accurated profile according to his features and evolution.

5

Conclusions and Future Work

5.1 Conclusions

In this work it has been developed a methodology that, through the analysis of usage statistics of a particular video game, it allows concluding improvements of the video game and grouping users in terms of their evolution in their skill level. To perform this analysis, it has been developed a video game that serves as an environment where users perform actions that are extracted and stored in a database for later analysis. Once stored, they can be represented and used to see, in graphic form, the evolution of the users actions on the video game map or to obtain usage data. Besides being represented, these statistics are used to analyze the profiles of the users present in that video game. To do this, time series clustering techniques and clustering algorithms based on medoids are used, such as PAM. With this algorithm, it can be also analyzed the most representative user of each of the groups obtained, giving a more particular view of each user profiles.

After performing the experiment, it can be found different constraints that can be highlighted:

- In case that the system has a very high number of users, the proposed architecture would have to be changed to one that supports that number of users.
- It could have benefited from a greater number of users that allows to obtain more representative samples.

The results demonstrate that:

1. The model allows to draw conclusions on the optimal configuration of the variables of the video game to give the video game a better experience for the user.
2. The model used allows to know the users profiles to allow make better decisions when making modifications to the video game. With the users profiles, the most used playing style is known, so it is easy to conclude what changes cause a greater impact among those users.

5.2 Future Work

There are also some issues that could be developed and studied in the future:

- With the results obtained from the analysis of the representation of the game, it could make changes to the configuration of the video game in order to balance or improve the aspects that have not been satisfactory. Thus, a second version of the game that would improve the gameplay issues would be launched.
- Improvements could be applied to artificial intelligence algorithms of the enemies to implement the proposed behaviors. This would increase the dynamism of the video game making the gemeplay experience for the user more satisfying.
- Improvements could be applied when sending statistics to the server in order to deal with the potential problems of the connection lost. A system that stored the statistics in a file and, if the connection is lost at some point, sent the missing statistics in order to not cause partition in the information could be developed.
- MatchMaking techniques could be applied matching users with the same kind of evolution of its profile. To do this, the video game should be improved to add the online component.
- It could be improved the video game to allow interaction between users. To do this, the tool called SmartFoxServer provides an environment integration with Unity for managing the configuration of online video game effectively.
- It could be improved the performance of the video game allowing a decrease in the minimum requirements to play. Likewise, it could be developed the video game for UNIX platforms.
- In case of having a large number of users, it would proceed to make a new server-side architecture that should manage the new user connections. To do this, SmartFoxServer has configuration options to accomplish this task.
- It could be improved the analysis techniques and extend them to contemplate more data analyzed in this work. As already mentioned, this work is a model that can be extended to consider a greater number of features when performing the analysis.

Glossary

- **FPS**: First Person Shooter
- **TPS**: Third Person Shooter
- **RTS**: Real Time Strategy
- **RPG**: Role-Playing-Game
- **PAM**: Partitioning Around Medoids
- **TBS**: Turn-Based Strategy
- **RAM**: Random-Access Memory
- **ROM**: Read-Only memory
- **NPC**: Non-Player Character
- **MMOG**: Massively Multiplayer Online Game

Bibliography

[1] Daniel T. Larose. *Discovering Knowledge in Data*. John Wiley & Sons, 2005.

[2] Rui Xu and II Wunsch, D. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645 –678, may 2005.

[3] T. Warren Liao. Clustering of time series data—a survey. *Pattern Recognition*, 38(11):1857 – 1874, 2005.

[4] Robert W Crandall and J Gregory Sidak. Video games: Serious business for america's economy. *Entertainment Software Association*, 2006.

[5] Ralph H Baer. *Videogames: in the beginning*. Rolenta Press, 2005.

[6] Thomas H. Apperley. Genre and game studies: Toward a critical approach to video game genres. *Simulation and Gaming*, 37(1):6–23, 2006.

[7] James Paul Gee. What video games have to teach us about learning and literacy. *Comput. Entertain.*, 1(1):20–20, October 2003.

[8] Tristan Henderson. Latency and user behaviour on a multiplayer game server. In Jon Crowcroft and Markus Hofmann, editors, *Networked Group Communication*, volume 2233 of *Lecture Notes in Computer Science*, pages 1–13. Springer Berlin Heidelberg, 2001.

[9] Nick Yee. The labor of fun: How video games blur the boundaries of work and play. *Games and Culture*, 1(1):68–71, 2006.

[10] Nick Yee. Motivations for play in online games. *CyberPsychology & behavior*, 9(6):772–775, 2006.

[11] Mark D. Griffiths. Amusement machine playing in childhood and adolescence: A comparative analysis of video games and fruit machines. *Journal of Adolescence*, 14(1):53 – 73, 1991.

[12] Craig A. Anderson and Brad J. Bushman. Effects of violent video games on aggressive behavior, aggressive cognition, aggressive affect, physiological arousal, and prosocial behavior: A meta-analytic review of the scientific literature. *Psychological Science*, 12(5):353–359, 2001.

[13] Marc Prensky. *Don't Bother Me, Mom, I'm Learning!: How Computer and Video Games are Preparing Your Kids for 21st Century Success and how You Can Help!* Paragon House New York, 2006.

[14] Sebastian Deterding, Miguel Sicart, Lennart Nacke, Kenton O'Hara, and Dan Dixon. Gamification. using game-design elements in non-gaming contexts. In *PART 2———– Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems*, pages 2425–2428. ACM, 2011.

[15] Karl M Kapp. *The gamification of learning and instruction: game-based methods and strategies for training and education.* John Wiley & Sons, 2012.

[16] Kai Huotari and Juho Hamari. Defining gamification: A service marketing perspective. In *Proceeding of the 16th International Academic MindTrek Conference*, MindTrek '12, pages 17–22, New York, NY, USA, 2012. ACM.

[17] Roger Smith. The long history of gaming in military training. *Simulation and Gaming*, 41(1):6–19, 2010.

[18] Simon McCallum. Gamification and serious games for personalized health. *Studies in health technology and informatics*, 177:85–96, 2012.

[19] Dmitri Williams, Nick Yee, and Scott E Caplan. Who plays, how much, and why? debunking the stereotypical gamer profile. *Journal of Computer-Mediated Communication*, 13(4):993–1018, 2008.

[20] Guillermo Jiménez-Dıaz, Héctor D Menéndez, David Camacho, and Pedro A González-Calero. Predicting performance in team games. *II for systems, C. Technologies of Information, and Communication, editors, ICAART*, pages 401–406, 2011.

[21] István Szita. Reinforcement learning in games. In Marco Wiering and Martijn Otterlo, editors, *Reinforcement Learning*, volume 12 of *Adaptation, Learning, and Optimization*, pages 539–577. Springer Berlin Heidelberg, 2012.

[22] Robin Hunicke and Vernell Chapman. Ai for dynamic difficulty adjustment in games. In *Challenges in Game Artificial Intelligence AAAI Workshop*, pages 91–96, 2004.

[23] Tim Marsh, Shamus P Smith, Kiyoung Yang, and Cyrus Shahabi. Continuous and unobtrusive capture of user-player behavior and experience to assess and inform game design and development. In *In Proceedings of the 1 st World Conference on Fun'n Games.* Citeseer, 2006.

[24] Dinara Moura, Magy Seif el Nasr, and Christopher D. Shaw. Visualizing and understanding players' behavior in video games: Discovering patterns and supporting aggregation and comparison. In *ACM SIGGRAPH 2011 Game Papers*, SIGGRAPH '11, pages 2:1–2:6, New York, NY, USA, 2011. ACM.

[25] Grace Chung and Sara M Grimes. Data mining the kids: Surveillance and market research strategies in children's online games. *Canadian Journal of Communication*, 30(4), 2005.

[26] Peter Vorderer, Tilo Hartmann, and Christoph Klimmt. Explaining the enjoyment of playing video games: The role of competition. In *Proceedings of the Second International Conference on Entertainment Computing*, ICEC '03, pages 1–9, Pittsburgh, PA, USA, 2003. Carnegie Mellon University.

[27] Richard TA Wood. Problems with the concept of video game "addiction": Some case study examples. *International Journal of Mental Health and Addiction*, 6(2):169–178, 2008.

[28] Mark Griffiths. Violent video games and aggression: A review of the literature. *Aggression and Violent Behavior*, 4(2):203 – 212, 1999.

[29] Paul Coulton, Will Bamford, Keith Cheverst, and Omer Rashid. 3d space-time visualization of player behaviour in pervasive location-based games. *Int. J. Comput. Games Technol.*, 2008:2:1–2:5, January 2008.

[30] B.G. Weber and M. Mateas. A data mining approach to strategy prediction. In *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on*, pages 140–147, Sept 2009.

[31] N. van Hoorn, J. Togelius, D. Wierstra, and J. Schmidhuber. Robust player imitation using multiobjective evolution. In *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, pages 652–659, May 2009.

[32] E.R. Hruschka, R.J.G.B. Campello, A.A. Freitas, and A.C.P.L.F. de Carvalho. A survey of evolutionary algorithms for clustering. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 39(2):133 –155, march 2009.

[33] J. B. Macqueen. Some methods of classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[34] Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

[35] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[36] L. Kaufman and P. Rousseeuw. *Clustering by Means of Medoids*. Reports of the Faculty of Mathematics and Informatics. Faculty of Mathematics and Informatics, 1987.

[37] Ricardo Aler, Jose M Valls, David Camacho, and Alberto Lopez. Programming robosoccer agents by modeling human behavior. *Expert Systems with Applications*, 36(2):1850–1859, 2009.