UNIVERSIDADE DA BEIRA INTERIOR
Engenharia

Serious Games
Design and Development

André Filipe Santos Barbosa

Dissertation to obtain the Master degree in the specialty
Computer Science
(2^{nd} cycle studies)

Supervisor: Prof. Doutor Frutuoso Silva

Covilhã, June 2011

Acknowledgements

First and foremost, I would like to thank my supervisor Prof. Doutor Frutuoso Silva, for guiding me through the process of writing and developing this dissertation. I would also like to thank every member of the Regain lab for the company and help provided throughout the whole process. Especially, Pedro Pereira and João Dias, who were very important for the development of Rocky - The Math Cat and without whom it wouldn't have been possible.

I would also like to acknowledge the importance my family, especially my brother Tiago Barbosa, my mother Filomena Santos and my father José Barbosa, for supporting me and for always being there for me, whenever I needed help. I also want to give a very special thanks to my girlfriend Paula Freire, for being so supportive throughout the process of writing this dissertation. You made this a much more pleasant experience for me. Thank you.

Resumo

O crescimento da indústria dos jogos de vídeo, despoletou um maior interesse no estudo deste fenómeno, o que consequentemente levou ao estudo de Jogos Sérios. Jogos Sérios são normalmente considerados jogos de vídeo que são desenvolvidos para outros fins para além do entretenimento. Estes fins incluem a educação e o treino, entre outros. Ao utilizar Jogos Sérios para a educação, os docentes poderiam conseguir captar a atenção dos alunos da mesma forma que os jogos de vídeo normalmente conseguem. Desta forma o processo de aprendizagem poderia ser mais eficiente. Adicionalmente, ao explorar o potencial destes mundos virtuais, é possível experienciar situações que de outra forma seriam difíceis de experienciar na vida real, devido ao seu custo, a razões de segurança e também ao tempo dispendido para as realizar.

O estudo de Jogos Sérios é ainda bastante disperso. No entanto, hoje em dia existe já um grande número de plataformas e ferramentas disponíveis que podem ser usadas para desenvolver Jogos Sérios. Por exemplo, os web browsers podem agora fornecer acesso fácil a mundos virtuais 3D. Isto permite que os criadores de jogos de vídeo tenham acesso às ferramentas necessárias para criar ambientes ricos, que possam ser acedidos por qualquer pessoa através de uma ligacção à internet. Adicionalmente, existem outras plataformas de desenvolvimento que podem ser utilizadas para alcançar objetivos diferentes. Tecnologias desktop fornecem um maior poder de processamento e permitem alcançar melhores resultados em termos de qualidade visual, bem como em termos de criação de simulações mais precisas.

Nesta dissertação descreve-se a criação e o desenvolvimento de dois Jogos Sérios, um para PC, desenvolvido em XNA e outro outro para a web, desenvolvido em WebGL.

Palavras-chave

Jogos Sérios, Jogos de Vídeo, Educação, WebGL, XNA

Abstract

With the growth of the video game industry, interest in video game research has increased, leading to the study of Serious Games. Serious Games are generally perceived as games that use the video games' capabilities to emerge players, for other purposes besides entertainment. These purposes include education and training, among others. By using Serious Games for education, teachers could capture the students' attention in the same way that video games often do, thus the learning process could be more efficient. Additionally, by exploiting the potential of these virtual worlds, it is possible to experience situations that would otherwise be very difficult to experience in the real world, mainly due to reasons of cost, safety and time.

Serious Games research and development is still very scarse. However, nowadays there is a large number of available platforms and tools, which can be used to develop Serious Games and video games in general. For instance, web browsers can now provide easy access to realistic 3D virtual worlds. This grants video game developers the tools to create compelling and rich environments that can be accessed by anyone with an internet connection. Additionnaly, other development platforms can be used to achieve different goals. Desktop technologies provide greater processing power and achieve greater results in terms of visual quality, as well as in terms of creating more accurate simulations.

This disseration describes the design and development of two Serious Games, one for PC, developed with XNA, and another for the web, developed with WebGL.

Keywords

Serious Games, Video Games, Education, WebGL, XNA

Contents

List of Figures

List of Tables

Acronyms

ASD - Autism Spectrum Disorder

CSS3 - Cascading Style Sheets 3

DGBL - Digital Game-based learning

DOM - Document Object Model

EMS - Emergency Medical Services

GBL - Game-based learning

HLSL - High Level Shader Language

HTML - HyperText Markup Language

RTS - Real-time strategy

STEM - Science, Technology, Engineering and Math

UDK - Unreal Development Kit

VRML - Virtual Reality Modelling Language

Chapter 1

Introduction

There is no doubt that nowadays video games are a growing part of our culture. Several research studies express that video games are now a rather large part of society[1][2]. The digital entertainment industries are multi-million dollar industries that keep growing and reaching more people every day. The video game industry in particular, was expected to generate an astounding $48.9 billion in 2010[3]. Serious Games have a broad spectrum of application areas, of which we emphasize education, healthcare, military and administration areas. The education and military markets combined were worth $400 Million per year in 2007 in the USA; the development of the emerging company training market was valued at $600 Million a year; the same growth was determined for the healthcare market, which was valued at $600 Million a year. The estimation of the total Serious Games market is $1.5 Billion in the USA for 2008 and $2 Billion, if combined with the European market[3], thus showing the great financial potential of Serious Games.

Serious Games are still relatively new in terms of research, but are raising a lot of interest regarding education and training since video games are able to capture players' attention and concentration for long periods of time, which is one of the biggest struggles for teachers or parents in terms of education. However, some concerns arise in regards of the claimed positive and negative effects of this type of games. There are some obvious potential advantages, such as allowing users to experience situations that would be unreasonable in real life due to safety reasons, excessive cost and time constraints, among others. Nonetheless, there has been some discussion whether video games can have a truly positive effect on determined skills development, or their effectiveness for learning or training. Regardless, if teachers managed to engage students in the same way that video games often can, the learning process could be greatly enhanced. This however, is not an easy task. In order to achieve this, we need to delve in several important topics:

- How are players being engaged in game play and how can we apply this knowledge to education or training.
- Are traditional video games viable for learning activities.
- What elements of traditional video games can be used in training or teaching.

1.1 Motivation for Video Games

Evidentially, video games have the ability to motivate players to carry on playing for large periods of time, hence the interest in researching what motivates players to do so. Greenberg et al.[2] argue that motivations for playing video games vary with age and developmental stage. This particular study suggested that younger players (5th grade) choose fantasy over other reasons for motivation, while older players (8th to 11th grades) prefer competition as motivating factor. Additionally, in a similar study conducted with children from 4th to 6th grade, the participants acknowledged pride in overcoming their competition as their primary reason for game playing. Being good at video games seems to increase the players' self-esteem, especially if they fail at other popular activities[4]. However, these younger players are more likely to experience frustration if games are too challenging, since they have less coordination skills, which can lead to discouragement. The same study reveals that for young adults (undergraduate students), stress relief is considered to be the most important stimulation for gaming. These young adults seem to believe that engaging in rule-breaking or authority-defying activities during game-playing is a method for dealing with daily life frustrations. Nonetheless, video games provide more than just averting negative feelings. The same adult gamers stated that video games are often used to avoid boredom, therefore they can incite fun and entertainment, which contributes to more positive moods.

People are driven by emotions, which video games appear to trigger rather easily. Olson's[5] survey pointed out numerous emotional reasons for playing video games, for both male and female participants. These reasons range from relaxation methods, to anger management, abstraction from life problems or even coping with loneliness. But even though emotions are regarded as an important factor for game playing, this survey acknowledged fun as the overwhelming motivator for players, regardless of age or gender (See Figure 1.1).

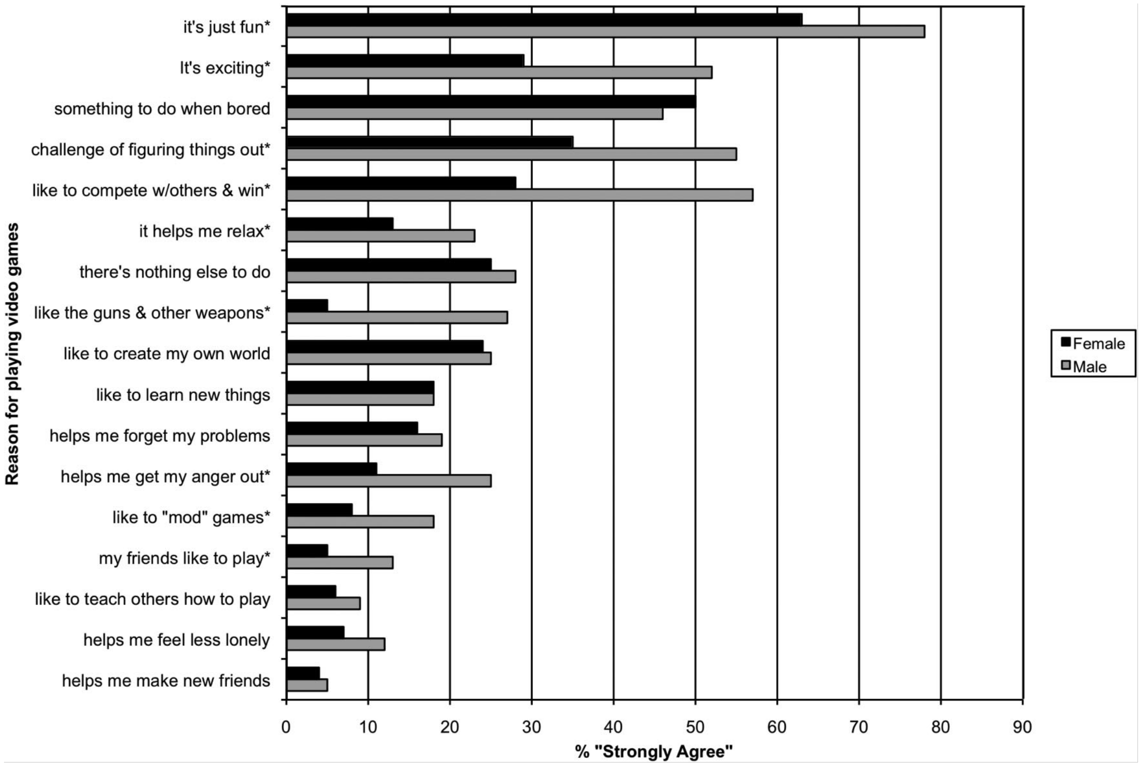

Figure 1.1: Reasons for playing video games for boys and girls. Extracted from [5].

Researchers present challenge, competition, a sense of achievement and ultimately the state of flow, as reasons for having fun with video games. Their results show that easy games that don't require much focus to beat, are not as fun as challenging ones[6]. Also, games that promote competition seem to particularly ensure that boys enjoy themselves and generate fun, while playing against each other[2]. In another survey conducted by Yee[7], the sense of achievement caused by the successful progression in video games, or by figuring out games' flaws is praised by players. This sense of achievement is also triggered when challenging another player and successfully defeating him in competition. Interestingly, one of the most valued ways to ascertain fun, is by determining the players' immersion in these games. This immersion is often called the state of flow, in which players get so engaged that they become oblivious to distractions[8].

1.1.1 Flow

The state of flow has been defined by Csikszentmihalyi[9], as a state in which people become so focused on an activity and experience such enjoyment, that they abstract themselves from everything else, whatever it may be. In this study, the author identified various characteristics that are required to enter the flow state:

- Feeling the activity can be successfully completed.
- The player can concentrate fully on the activity.
- The activity has clear goals.
- The activity provides fast feedback.
- The player is deeply involved in the activity.
- A sense of control over the actions is necessary to perform the activity.

During this state, a person's sense of self-awareness is lowered and the sense of time also seems to be altered. Consequently, players are so absorbed in the activities that they lose track of time or awareness of their surrounding environment. Flow can be achieved by adjusting the level of challenge present in the game, to the player's level of skill, ultimately keeping the player's emotions in between boredom and frustration. Namely, not making the game neither too easy (to avoid boredom), nor too difficult (to avoid frustration).

With the exception of the flow state, there is still not a general consensus on why people play video games, with various reasons being chosen by different people. However, this fact is not very surprising since video games themselves vary immensely, even within each genre. There are genres and sub-genres that range from graphics style (2D or 3D, cartoon or realistic), to complexity (simulation or arcade) or even narrative, thus providing different experiences. Nonetheless, different genres or sub-genres may have different potentials for applying learning aspects.

To further comprehend video games and in what ways they can help the learning process, it may be important to categorize the main game genres.

1.2 Video Game Categorization

The video game industry evolves with every new generation of hardware, thus increasing video games' complexity, not only in terms of graphics, but also in terms of interaction and design. Therefore, it is only natural that new types of games are created, generating more genre diversity or even combining genres. This fact makes it increasingly more difficult to label video games based on categories. Also, there are no standard categorization guidelines, resulting in different entities labeling games in different ways. Still, video games are often labeled within the following genres:

- **Action Games**: Reaction-based games that require fast hand-eye coordination. Include other sub-genres, such as first person shooters or platform games.
- **Adventure Games**: Unlike action games, these games don't require fast reflexes. Instead, they normally require the player to solve various logical puzzles without time constraints.
- **Fighting Games**: Generally, involve fast paced combat between the player and one or more opponents.
- **Puzzle Games**: As the name suggests, provide various puzzles to be solved by the player. The puzzles include ciphers, visual-spatial clues and word or symbol patterns, among others.
- **Role-Playing Games**: Usually story driven games, where the player takes the role of the hero and can adjust his abilities or skill set, as he progresses in the game.
- **Simulations**: Allow players to simulate aspects of a real or fictional reality in recreated environments.
- **Sports Games**: Try to emulate real sports activities.
- **Strategy Games**: Focus on game play mechanics, which require careful and skillful thinking and planning, such as commanding armies.

The above genres, however, cannot accurately incorporate every single game available or made. It is common for new games to not fit in any of the pre-existent categories.

Additionally, various games go beyond a single definition and mix elements from several genres. For instance, almost every modern first-person shooter includes some kind of role-playing element, such as weapon enhancements.

There is not a lot of literature in terms of inquiring on which genres are fit for applying learning concepts. Frazer et al.[10] conducted a study in order to compare first-person shooters, role-playing games, strategy games and puzzle games, to determine which genre is more suitable for learning. This study's conclusions are summarized in Table 1.1.

Game Genres	Strong Features	Poor Features
First-person Shooter	Affording conversation Displaying new knowledge Encouraging exploration Immersing the player Rewarding success	Uniting learned resources Balancing difficulty Too fast-paced
Role-Playing	Provoking curiosity Displaying new knowledge Uniting learned resources Encouraging exploration Immersing the player Rewarding success	Information context Blend learning information World creation Balancing difficulty
Strategy	Displaying new knowledge Uniting learned resources Expressing information Blend learning information	Affording conversation World creation Provoking curiosity Immersing the player
Puzzle	Clear goals Information context Immersing the player	Affording conversation Encouraging exploration Provoking curiosity

Table 1.1: Comparison between different video game genres in regards of learning application.

So, first-person shooters seem to provide strong attributes for good learning games, but their fast-paced action does not contribute to an effective learning environment, since learners have different learning paces. This genre can, however, supply exploration of learning environments with a first-person view, which in combination with realistic settings can offer good simulation experiences. Therefore, this genre may be more appropriate for training than education. On the other hand, role-playing games have innate properties suited for learning, such as effective allocation of learned information and instant display of new information. It is usual for these kinds of games to have some sort of implementation of a log system, where players can check previously introduced game play hints, or additional game rules. Puzzle games lack most of the beneficial attributes present in the previous two game genres, but excel in terms of clarifying goals, thereby being perfectly adequate for teaching simple, but important concepts. For instance, board games have clear goals that players learn rather easily, whereas other game genres have a much bigger learning curve, due to their complexity. The strategy genre has a nice pace to it and provides information clearly, which makes it appealing for adaption into a learning context.

Although current research is yet to provide overwhelming evidence regarding the correlation between video game genres and their potential for teaching, we can infer that the game genre must be chosen considering the theme or subject that we are trying to teach, due to their different characteristics.

1.3 Objectives

The main objective of this dissertation is to research video games, in order to infer the potential of this medium's usage on activities other than entertainment. It is important to determine what makes video games so successful in capturing the player's attention, in order to be able to use these elements for non-entertainment purposes, such as education or training. Additionally, it is necessary to survey what application domains are viable for the employment of Serious Games and what benefits can come from using these solutions, for each of these domains.

Furthermore, in order to choose development technologies and deployment platforms, it is required to survey the most powerful tools for designing and developing video games available today. This includes researching game engines and target platforms in regards of not only visual and feature prowess, but also in terms of user accessibility. After this survey and choosing the development technologies, the goal is to design and develop Serious Games for education, in order to use them as research experiment and try to understand the benefits of using these types of games for learning, in comparison to traditional methods.

1.4 Dissertation Organization

Chapter 2 comprises of an analysis of the state of the art for Serious Games. This chapter attempts to define the term itself, present research about its origin, introduce some of the reported benefits discovered in recent research publications and finally typify some application domains on which Serious Games are being applied successfully.

Chapter 3 presents the development technologies researched, from games engines, to target development platforms and technologies. This chapter also presents comparisons between these development technologies' features, as well as reasons to why some of them were chosen for the development of the work on the context of this dissertation.

On chapter 4, the design and development stages involved for the creation of the work developed for this dissertation are described. This chapter also contains information on the contributions made through the developed work and also some results obtained throughout the process.

Finally, chapter 5 consists of the conclusions to this dissertation and also presents the future work planned to further deepen our research on Serious Games and their effectiveness.

Chapter 2

Serious Games

2.1 Definition

Following the growth of the video game industry, the Serious Games theme has seen an increased interest by many research groups that delve in areas as diverse as computer science, psychology, pedagogy, health, among many others. Serious Games have the ability to expand the horizons of video games to include scientific simulation and visualization, industrial and military training, medical and health training, education, amongst other areas less studied, such as cultural training or public awareness[11]. The growing integration of Serious Games research and video games research in general, into research groups, is leading researchers to believe that a big gaming revolution is imminent [12]. In the coming years, video games should expand and spread as distinct media forms, in comparison to those available today. For instance, in the next couple of years, 3D web pages may be very common with the standardization of 3D web technologies like WebGL. Such media forms should provide virtual experiences and simulations, to allow users to experience different beneficial situations (health and professional skill improvement, among other benefits) thus becoming a vital part of society.

However, the concept of Serious Games still lacks a single definition. It usually refers to video games used for purposes other than entertainment, namely training, advertising, simulation or even education. Still, many authors define Serious Games as more than mere video games used for non-entertainment purposes[13][14]. They are referred to as games that engage the user and simultaneously contribute to the achievement of a certain objective other than just entertainment, whether the user is aware of that fact or not. There has also been some discussing about the term *Serious Game*, since usually the terms *Serious* and *Game* aren't associated and are actually perceived as opposite terms. Most traditional video games are developed with the fun factor as one of their main goals. Apart from providing an engaging storyline, being visually appealing to the user or even thematically interesting, a game's core is about the fun it can bring to the user. This fun factor can be experienced by the players through a wide range of the game's core features, one of them being the fact that a game must provide a challenge and yet this challenge must be surmountable. This is one of the biggest issues with learning through Serious Games. Learning is usually regarded as a serious activity and set apart from fun. However, Serious Gaming research has shown that learning through video games has positive effects, since the users can abstract themselves from the learning process itself and learn while playing. Furthermore, fun has also been described as a side effect of learning something new[15].

To further highlight the importance of the fun factor in learning, it is noted that commercial entertainment games are already being used for other purposes than entertainment. Simulation games such as Flight Simulator are being used as training tools for flight learning[16]. These games, while being entertaining, have underlying elements of strategic skill training, which provide a perfect tool for training under certain situations. Through the use of these games, it is possible to simulate realistic situations that provide valuable experiences and also support discovery and exploration while saving money and lives.

Zyda[17] has argued that the addition of pedagogy[1] is what differentiates a Serious Game, from a traditional one. Nonetheless, pedagogy should not completely overshadow the entertainment factor when developing Serious Games. Video games can be effective for pedagogy simply because the players can abstract themselves from the learning process and learn while they are playing. Therefore, if games aren't entertaining, users will become bored and the learning curve will be decreased. Fun is often regarded as the main factor for entertainment. But, it is not the only form of entertainment, nor is it the only way to engage users in a video game. There are a lot more elements responsible for engaging users in a game besides fun, such as intense or passionate involvement through emotional output, goal driven gaming or structuring rules[18]. These engaging elements provide motivation to players and are

[1] Pedagogy - Activities that educate or instruct, thereby inducting knowledge or skill.

the reason why research on game-based training was initiated in the first place[19]. Using Serious Games to achieve a more effective education, may lead people to believe that children don't enjoy learning. However, this is not the case. Some researchers claim that enjoyment is a consequence of learners sensing that they are progressing and also when the learning process is done within an adequate context[20]. Fun and enjoyment are also side effects from being in the state of Flow[9]. Therefore, maybe the focus of Serious Games should be on achieving this state, instead of focusing only on entertainment. In this condition, users can learn and enhance their skills more efficiently [21]. However, effectively promoting Flow isn't a simple task. Malone [22] described in his research that in order to achieve the Flow state, the game must be able to take into account the learners' skills and knowledge and accommodate the proper challenges. Otherwise, the learners won't enter the Flow state, but instead will either be bored or feel anxiety. Figure 2.1 shows a graphic that represents the flow model in regards of the challenge.

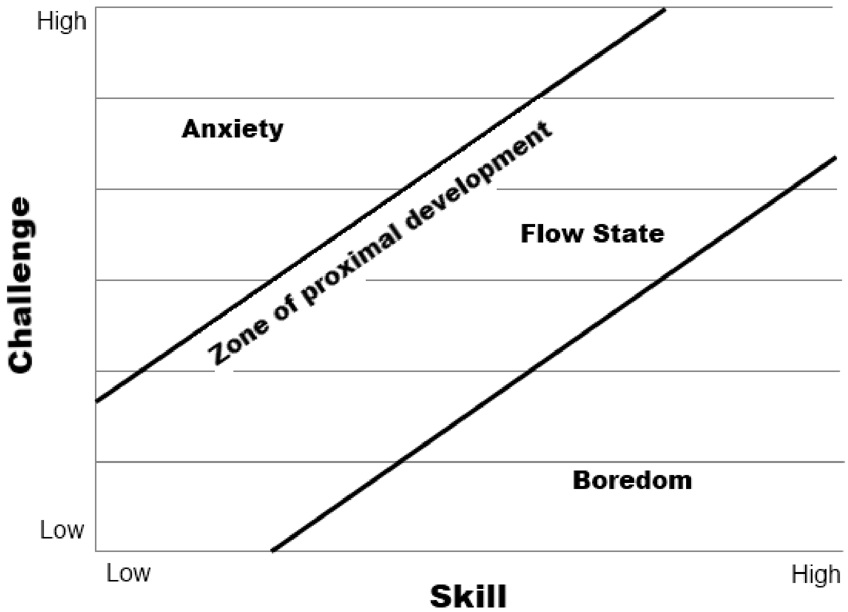

Figure 2.1: Flow Model. Adapted from [21].

So, rather than trying to provide just fun or entertaining experiences, we should aim to understand what elements contribute to Flow, since this state induces great enjoyment on players. Then, build the core of the game based on these engaging elements, while having the learning factor as a side effect, or as an unconscious achievement to the user.

Serious Games and traditional video games have different purposes and thus are developed with different goals in mind. Serious Games focus on problem solving, while traditional games often ensure richer experiences, with bigger and better effects. Also, while both aim to provide fun experiences, Serious Games need to include elements of learning or training. In terms of simulations, although many high-end entertainment games render realistic graphics, their simulations are usually simplified in order to avoid tedious processes. For instance, racing games simplify the process of driving with manual transmission[15]. Serious Games however, shouldn't always simplify these processes. Also, researchers acknowledge the importance of 3D for Serious Games, since it enhances the experience, making it a much more realistic simulated situation. Furthermore, a quality 3D simulation application gives users the immediate notion of value to replace certain real world activities[17]. These studies seem to indicate that a compelling 3D environment is in fact a necessity, rather than a good feature. But interestingly, most available Serious Games are not very technologically advanced. This fact results from the desire to develop games that will be widely available, on a variety of hardware and operating systems. Most authors believe that, when developing Serious Games, the technical proficiency or experience fullness is not as important as the simulation used to solve a problem.

Even with the amount of research done on Serious Games, there is still some skepticism about their real benefits, in part due to the lack of unquestionable and concrete proof of the usefulness of these games as tools and also because of the general sense that a game is basically a toy. Accordingly, the use of video games in schools remains very uncommon. Several reasons are usually given on why video games have yet to succeed in schools[20]:

- Due to time constraints, teachers claim that it is hard for them to identify in a timely manner, how a specific game can be relevant to the subject they are trying to teach. Also, they find it difficult to verify the game's thematic accuracy and content quality.

- It is difficult to persuade education representatives to use video games, despite their apparent educational benefits, since it represents a drastic change to the traditional teaching methods.

- Serious Games have a variety of functionalities or contents that are essential to engage the player, but are irrelevant for the learning process, thus being often considered as time wasting processes in limited timed lessons.

In summary, the concept of the term Serious Game still lacks a truly consensual definition. The one that is usually agreed upon is that Serious Games use gaming technology and resemble a lot to video games in terms of core attributes, but have a purpose other than mere entertainment. These purposes range from education, to training, to health, among other areas. While some authors use the fun factor as a drawing line between a Serious and a traditional video game, others argue that it is a key component to both Serious and traditional video games.

The next section provides an insight on some related concepts and also on the advantages of Serious Games.

2.2 Related Concepts

The use of technology in education and training is not exclusive to Serious Games. Notwithstanding, Serious Games is a subject which is related to a lot of different domains. A lot of these domains overlap each other, such as e-learning, edutainment, game-based learning or even digital game-based learning. All of these concepts share certain attributes, but are considered different approaches.

2.2.1 E-Learning

E-Learning refers to computer enhanced learning, or technologically interactive learning, being usually connected to distance learning [23]. It does not, however, involve video games. With E-learning, students can have easy access to subject materials online, flexible schedules and flexible study places. In other words, E-learning is very similar to traditional learning, but with technology improvements[24]. In terms of using video games for education, there are general themes common to each domain, such as taking advantage of games' motivational capabilities, in order to make the learning process more appealing. However, domains like edutainment and game-based learning fail to achieve these goals by ignoring some key elements that make playing video games an enjoyable experience.

2.2.2 Edutainment

Edutainment is defined as education through entertainment. In general, it refers to any type of education that also entertains. However it is generally linked to video games with educational aims. Edutainment's main target group is children, hence the emphasis on basic intellectual skill learning, such as reading, mathematics and science. The problem with edutainment usually resides on the absence of efficient learning games. They are often described as being boring, because they involve only memorization drills and fact teaching, which are ineffective since they fail to engage the player. With the increasing technological evolution, edutainment evolved to Serious Gaming. This evolution was triggered by America's Army[25] in 2002. This game was released by the U.S. Army and was the first Serious Game to be introduced to the market. Later that year, Woodrow Wilson Center for International Scholar in Washington, founded the Serious Games Initiative, spreading the term Serious Games worldwide. Serious Games share edutainment's objectives, but broaden them far beyond memorization or teaching facts, instead including video game elements that result in engaging experiences. They also expand the application domains to all areas of education, like training, teaching and information, for all ages[15].

2.2.3 Game-based learning (GBL) and Digital game-based learning (DGBL)

Game-based learning (GBL) and Digital game-based learning (DGBL) are other concepts that are often related to Serious Games. They are usually described as Serious Games that deal with applications that have well defined learning outcomes[26]. The obvious difference between GBL and DGBL is that the second uses video games, while the first uses only traditional games. The DGBL approach makes more sense nowadays since today's students are fluent in the digital media language, so to speak. Some authors account DGBL and Serious Games as basically the same thing. Corti[19] claims that DGBL has the potential to improve training activities because of its ability to engage and motivate, through role playing or repeatability, in the sense that the user can always try a different approach every time he/she plays. The same can be said about Serious Games.

2.2.4 Exergames

As the name sugests, Exergames are games that motivate players to do exercise routines. They usually require players to perform movements, in order to complete goals. They can also be considered Serious Games, since they use video games to achieve other goals than solely entertainment. Exergames have two distinct goals: encouraging players to exercise and facilitating behavioral change, as in modifying the players attitude towards exercise[27]. These goals are usually achieved through game play focused on using the players' movements to interact with the game. In order to do so, exergames often use special peripherals (e.g. motion capture through camera or accelerometers) or smarthphones (e.g. accelerometers). These games can have a wide range of applications. For instance, children that suffer from Autism Spectrum Disorder (ASD), have shown to benefit from regular exercise routines. But it is hard to motivate them to exercise. However, they are very attracted to video games and technology in general[28]. By using exergames, these children are motivated to exercise and consequently improve their health and well being[29].

In the next section, advantages like the one stated above are approached, in order to provide information on how video games can be beneficial to people's lives for more than just entertainment.

2.3 Benefits of Serious Games

Researchers have proved that video games can teach lower-level intellectual skills and also improve physical skills, but that is just a glimpse of what video games are believed to achieve. The principle of Situated Cognition states that learning within a meaningful and relevant context is more effective than learning outside of those contexts[30]. Researchers have demonstrated its effectiveness in many studies over the years[26], specifically, that learning that occurs in meaningful and relevant contexts has increased effectiveness, whereas learning outside those contexts is less effective. Therefore, if developers achieve the creation of compelling contextual environments, Serious Games can provide more effective learning experiences. Also, recent studies conducted about STEM (Science, Technology, Engineering and Math) education have demonstrated a 7%-40% increase in learning effectiveness among students who used Serious Game-based education over traditional education programs, while teaching the same subject matters[26].

Playing has also been appointed by researchers as a primary socialization and learning mechanism to all human cultures, as well as many animal species[26]. For instance, lions do not learn to hunt through direct instructions, instead they learn through play and modeling. Serious Games and video games in general, definitely use play as an instructional strategy. It is known that video games have well-established rules that players must learn in order to succeed, so learning is inherently present in playing a video game. Thus, taking these elements and providing an educational context, video games become effective learning tools because the learning process occurs within a purposeful context (to the game). If the learning is relevant when applied and practiced within a certain context, then what players learn is correlated to the environment in which they play[30].

There are also other theories that contribute to the belief that games bring cognitive benefits to their users. Jean Piaget's theory of cognitive development[31] claims that intellectual advancement happens due to the fact that humans seek cognitive equilibrium, a state of mental balance. When new information is presented to a person, he/she experiences cognitive disequilibrium, a state in which the person has to adjust himself/herself to include the newly acquired information by assimilation or accommodation.

- **Assimilation** - is the process of interpreting new information by referencing preexisting cognitive structures. For instance, if a player who is aware of a set of game rules, is presented with a new rule within the game, he assimilates the new rule in accordance with the previous basic rules.

- **Accommodation** - is the process of restructuring old information to accommodate new one. For example, if during a game a so-called game-changing event occurs, where the game play changes drastically, the player has to accommodate these new rules and restructure his preexistent knowledge about the game and its rule set.

Piaget claims that both of these processes are used throughout life, as the person increasingly adapts to the environment in a more complex manner. He also believed that Cognitive Disequilibrium is the key to the intellectual maturation of an individual, since intellectual growth is achieved through acquiring new ideas and experiencing new things.

The process of cognitive disequilibrium is engrained in games, providing motivation and engagement. The extent to which video games surpass expectations, through creating Cognitive Disequilibrium, without exceeding the user's capacity to succeed, seems to be one of the main elements responsible for whether games fail to engage or not. Playing video games requires a constant cycle of hypothesis formulation, testing and revision, which happens fast and results in immediate feedback. Video games that offer few challenges, are not engaging and will not succeed as teaching tools. Video games as teaching tools are efficient when they create a continuous cycle of cognitive disequilibrium and resolution (through assimilation or accommodation), while also allowing the player to be successful.

Positive impacts are being reported regularly, like the positive relation between experience in video games and performance in training medical students[32][33]. This particular positive effect is assigned to the 3D perception experience that comes from playing 3D video games. Furthermore, Serious Games have been found to increase players' abilities in spatial modeling, form creation or even mental rotation capacity[34][35]. On a different note, through the use of realistic environments, Serious Gaming technology can teach major incident triage with improved accuracy[36]. Also, software developed for attention training, has shown that even non-systematic experiences with video games improve the attention of children[37]. Furthermore, video games have shown to have potential benefits that include improved self-monitoring, problem recognition and problem solving, decision making, short-term and long-term memory, and increased collaboration skills[38]. For instance, on-line community games help the creation of collaborative knowledge, as well as the development of information-seeking habits[39]. Another advantage for gamers resides on the fact that their thinking strategies are more analytical, instead of a trial-and-error thinking approach[40] and also that increasing the difficulty of video games or having competitive scoring is useful in competitive corporate training[41]. Also, experiences made in a traffic school, showed that students with high experience in video games managed to get superior driving ranks, in comparison to students with lower experience[42]. Accordingly, through multiplayer online games, children can learn socialization and negotiation techniques by participating in collecting or trading game items[43]. Multiplayer games may also be helpful in testing the limits of what is acceptable behavior and what is not, since players are bound to interact with each other. For instance, if a player is reprimanded for using unfair tactics or cheating in these virtual worlds, he might associate the same outcome in real world situations[44].

Unfortunately, there are also downsides resultant from playing games. There are some possible negative side effects, including health issues, such as headaches, fatigue or mood changes and psycho-social issues, like social isolation and substitution of social relationships. There are also some known issues caused by violent video games and visual violence in general, such as aggressive behavior and negative personality development. Baldaro[45], researched some of the short term effects of games in terms of physiology (arterial pressure and heart rate) and psychology (anxiety and aggressiveness). This particular study showed that this short-term effects happen in violent games and not in non-violent games. Also, a lot of the negative effects are caused by excessive time spent playing video games and even violent games can be beneficial in terms of alleviating stress or frustration[18].

The next section approaches some of the most important application domains of Serious Games.

2.4 Application Domains

A lot of commercial sectors are beginning to realize the potential benefits of using Serious Games and are actively replacing traditional methods with such tools. Serious Games can be applied to a large set of domains, with different goals in mind. The following sections describe some of these domains, as well as how Serious Games are being used to bring improvements to them.

2.4.1 Military

Contrary to general belief, military games are far from a modern trend. These games originated over 4.000 years ago and allowed training and improvement of battle planning and strategies[15]. Obviously, nowadays they have evolved from board games, into highly intricate and complex experiences, resembling increasingly more with the actual activities than with a game itself. Game technologies allow the development of realistically engaging simulations at a relatively low cost, when compared to traditional simulations. This seems to be one of the strongest reasons why interest and support for Serious Games has increased, while traditional simulations are continuously losing supporters. But even though Serious Games have a lower cost than traditional simulations, it does not mean that they are cheap. Video game technology is always evolving and development cost are always increasing, in order to maintain the highest level of simulation possible, which is resulting in spending close to $4 billion every year just for this sole purpose[15].

Over the years, several commercial games have been used by the military. Video game sales seem to indicate that there is a great commercial market for war games and some of those games provide viable military simulations, due to their resemblance to real life war scenarios. Commercial games that have been adapted for military use include Warcraft, Doom or Operation Flashpoint[15]. However, one the first and most well-known Serious Game used by the military is America's Army. At the time (2002), this game tried to look as real as possible, by including such details as weapons and vehicles being exact 3D models of the real objects. This game also featured a first-person view and tried to enact real war scenarios and situations, so that the player could feel like he was actually interacting in a war setting. Games like America's Army can improve skills like hand-eye coordination, teamwork ability as well as problem solving and multitasking, which are essential for combat situations. Figure 2.2 shows an example of America's Army game play.

Figure 2.2: America's Army - Commercial game used for Military training (Extracted from [25]).

Capitalizing on America's Army' success, the U.S. military has funded similar Serious Game related projects. However, more recently and due to the success of those games, the military have started branching out to other skills training besides combat. For instance, the University of Southern California is developing two projects as part of these Military programs: Tactical Iraqi[46] is designed to assist and accelerate the trainees' understanding of the spoken Arabic language, in order to further help them once they are on the field; Virtual Iraq's[46] goal is to address Post-Traumatic Stress Disorder problems in combat veterans. Recent feedback from soldiers indicates that games like Tactical Iraqi are important for missions on foreign countries. For example, a US Marine operative has recently attributed the successful completion of a mission in Iraq (with no casualties) to their enhanced communication skills, acquired prior to the mission[47]. Furthermore, the military Serious Games field is likely to expand to other complex training skills, such as tactical coordination and survival in persistent worlds, with the help massively multiplayer games[15]. Figure 2.3 shows a hypothetical situation simulated by the Tactical Iraqi game.

Figure 2.3: Tactical Iraqi - Cultural training for military operatives (Image extracted from [48]).

From a military stand point, playing not only war games, but video games in general, improves skills like hand-eye coordination, multitasking ability and teamwork abilities. Previously, military simulations involved only combat situations, but more recently have included other skill training, such as foreign languages and cultural training to help facilitate adaptation to foreign countries, prior to foreign missions[15]. Future application areas for the military field also include massively multiplayer online games and virtual reality trainers.

2.4.2 Administration

Serious Games that target administration have a large range of applicability, since it may concern different kinds of tasks and situations such as crisis management, disease outbreaks, healthcare policy issues, traffic control, firefighting, budget balancing and ethics training, among others. In the case of firefighting, providing people with information about how to react in case of fire, tends to be based on written instructions on walls and signs, or evacuation drills at the workplace[49]. These instructions are often overlooked by people, but even if that was not the case, during emergency situations people are under a lot of stress and might not remember the instructions correctly or be affected by the "Tunnel Vision" syndrome[50]. This syndrome is described as losing peripheral vision and consequently losing attention to detail, which is crucial in emergency situations. Evacuation drills can certainly help in emergency scenarios, but are unfortunately not mandatory and therefore not a common thing in most places, which is understandable due to reasons of cost.

A Serious Game can be more effective than reading materials or participating in evacuation drills since it has the potential of motivating people to train. Due to the engaging factor that video games provide, they may encourage people to even play at home and thereby train outside working hours. This would definitely be a great feat for employers, since they could avoid disrupting work activities because of drills and at the same time prepare their employees for emergency situations. Furthermore, a video game can supply personalized advice based on users' errors and make them available at anytime. Figure 2.4 shows a Serious Game developed for fire safety skills training[49].

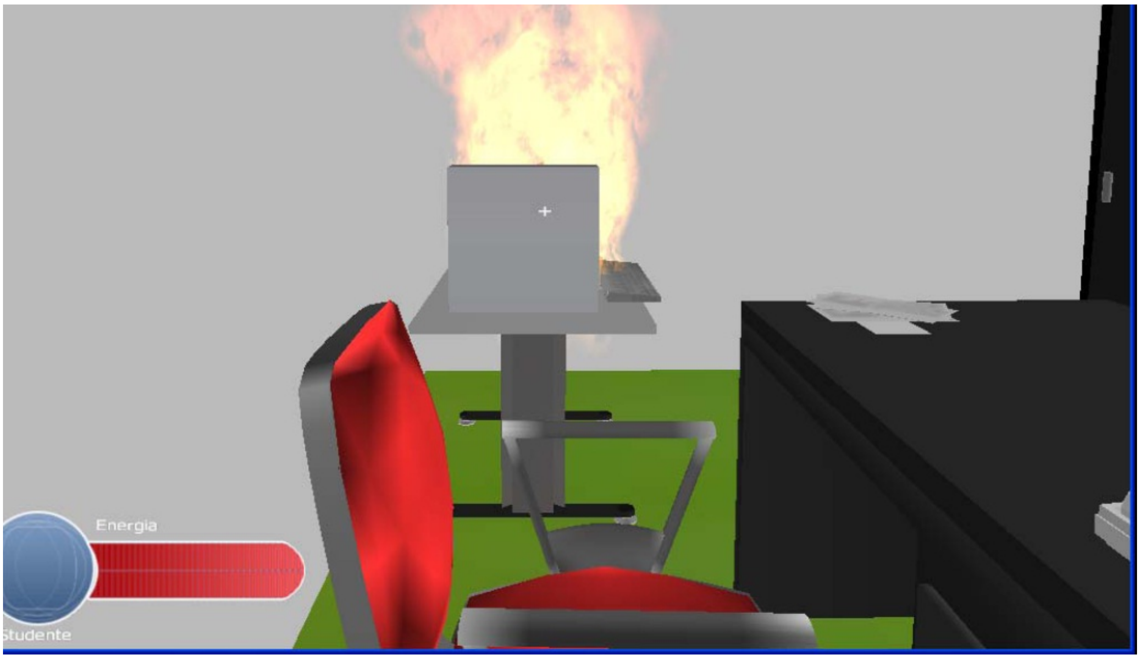

Figure 2.4: Fire Safety Skills Serious Game (Extracted from [49]).

Serious Games allow rich and engaging simulations of crucial situations, but perhaps, their greatest advantage is that they allow different types of first responders, such as firefighters, police and medical personnel, to practice dangerous situations in a safe and controlled environment, while not spending excessive amounts of resources.

2.4.3 Education

Traditional teaching methods usually fail to incorporate and take advantage of today's technology. Nowadays, children have access to technology from a very young age, whether for entertainment or other activities, so it is only fitting that they can also take advantage of that for education. The use of technology for education purposes can have a very positive effect, especially from this generation onwards. Video games particularly, provide an effective way to engage students in learning activities, since they have the ability to stimulate cognitive processes like acknowledging displayed information, deductive and inductive reasoning and also problem-solving[51]. However, a previous attempt, called Edutainment, was made at using video games for education and failed to do so. This attempt steadily lost interest, due to the poor implementation of the games themselves, which disregarded video games' basic features, such as providing fun and engaging experiences[15]. On the contrary, Serious Games include such features as part of their core experience, thereby fitting the role of a great medium to enhance teaching.

Nevertheless, Serious Games acceptance as valuable tools for education is still being questioned, since there is not a large amount of research that unequivocally proves their benefits for education. This is partly due to the fact that this research relates to a large spectrum of subjects and also as a result of the diversity and complexity of the games themselves[20]. But based on the research available currently, video games as educational tools show great improvement when compared to traditional tools. The following list shows some of the research done on applying Serious Games to education:

- Teaching handwriting[52]
- Teaching optic physics[30]
- Teaching physics[53]
- Emotion recognition and logic[29]
- Cultural training[11]
- Teaching social awareness[54]
- Teaching foreign languages[55]
- Teaching safe road behavior[56]
- Teaching sea-related best-practices[57]

Evidentially, Serious Games for educational purposes is a hot topic for video game research. One of the greatest issues in teaching is the difficulty to capture the attention of students, who generally prefer to be doing other activities, where they can have more fun. If teachers could have the students' full attention, the learning process would be less cumbersome, for both teachers and students. Serious Games seem to be the answer to these issues, at least at some degree, since amidst playing video games, children interact with multimedia content, which motivates them to pay more attention and consequently leading to a more effective learning process[52]. Figure 2.5 presents one of these games, where players can improve their social awareness, with the game emphasizing problems like poverty and lack of resources on third world countries.

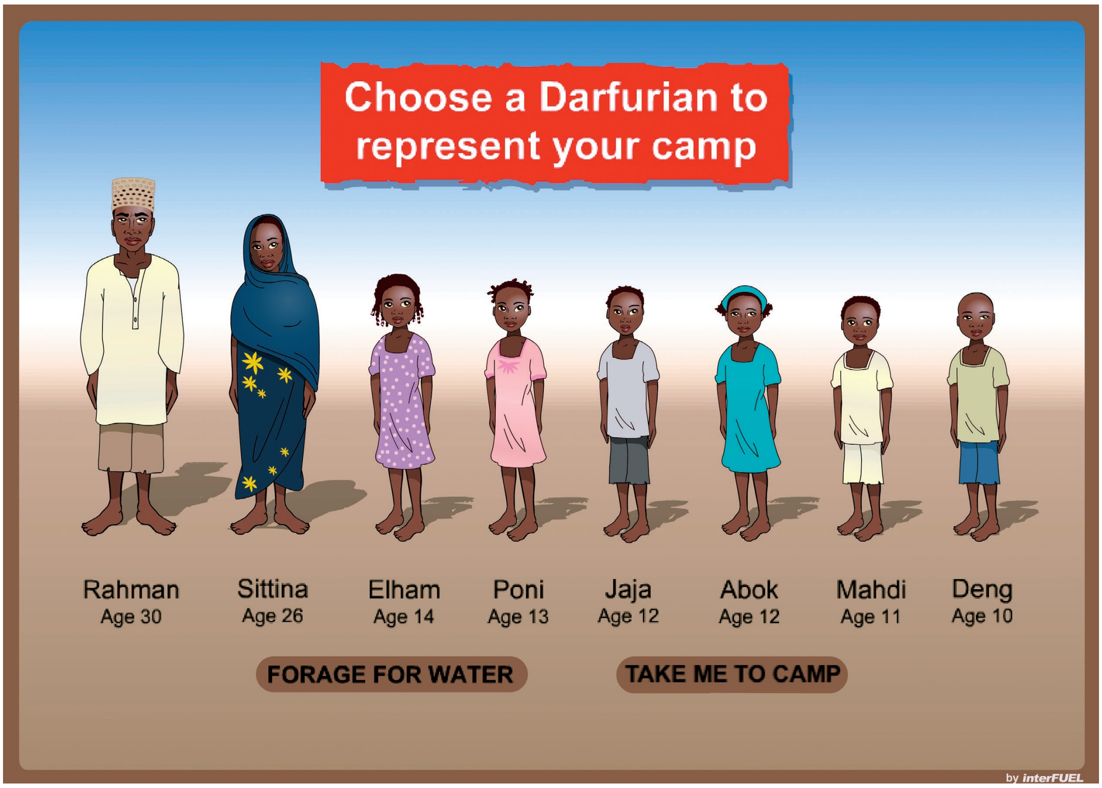

Figure 2.5: Teaching Social Awareness (Extracted from [54]).

As was stated previously, Serious Games for education are a growing part of the industry, with many games already available for teachers to take advantage of. Still, there are some concerns that need to be addressed, in order to effectively spread Serious Games as learning tools. It is necessary to persuade education representatives to consider the benefits of video games and make the effort to upgrade the technologies currently employed in education, since most schools don't have powerful enough computers to run current video games. Additionally, teachers are required to familiarized themselves with the games, in order effectively utilize them for teaching, which is also one of the reasons appointed to avoid using video games for education[58].

2.4.4 Corporate

With the emerging new technologies and increasing company training needs, interest in Serious Games and simulations has increased. Corporations often train their employees in domains as diverse as communication, organization, or job-specific skills. For corporate application, some advantages of using Serious Games are obvious, such as cutting costs for training staff and for equipment, but at first glance, no other significant enhancements seem to be appointed when compared to traditional methods. However, with the number of employees acquainted with video games increasingly rising in correlation with the industry's growth, the use of Serious Games for training seems even more appropriate. If employees enjoy video games, their attention can be easily obtained by Serious Games and thus their training might be more effective. Conformably, Serious Games may ease the learning process of complex or more technical materials. These materials are often considered boring by the general audience, but Serious Games manage to overcome this issue and capture audiences that are otherwise hard to reach[59]. The fact that video games make a change in the learning stance of the trainees, from passively learning, to actively participating in the learning process, is also a valuable asset that makes this process more appealing[60]. All of these reported benefits are direct indication that corporations have in Serious Games an innovative and effective method of training their employees.

Also, researchers claim that video game players are better at dealing with decision making situations, due to their virtual experience of debating risks against rewards[61]. This fact, associated with the previously reported video game-related improvements, such as improved multitasking skills, helps employees achieve better job performances, while using more sophisticated tools. There are already a lot of examples of Serious Games applied to corporate training such as LearningBeans[62] or Virtual Training Bank[63]. Virtual Training Bank is a game for financial analysis and risk management, with the goal of helping trainees detect fraud and also assess financial risks. LearningBeans is game which helps learners understand the interdependencies between all aspects of business, from manufacturing to sales, marketing, finance and distribution, among others. Figure 2.6 shows an example of the LearningBeans Serious Game.

Figure 2.6: A screenshot of the LearningBeans Serious Game (Extracted from[62]).

2.4.5 Healthcare

The Healthcare industry is becoming more active in looking at the Serious Games market for solutions to training or other purposes, like rehabilitation, with various video games being created for these purposes. Serious Games have a great impact on both mental and physical aspects of players, thereby being perfectly applicable on healthcare issues[64]. From information gathered through research, the healthcare industry might be one of the largest markets for Serious Games. Sawyer[12] compiled a list of possible application areas for Serious Games on healthcare, which can be seen bellow on table 2.1.

Area of Health	Personal	Professional	Research	Public Health
Preventive	Exercising Gaming Stress	Patient Communication	Data Collection	Public Health Messaging
Therapeutic	Disease Management	Pain distraction	Virtual Humans	First Responders
Assessment	Self-Ranking	Measurement	Inducement	Interface and Visualization
Educational	First Aid or Medical Information	Skills and Training	Recruitment	Management Simulations
Informatics	Personal Health Records	Electronic Medial Records	Visualization	Epidemiology

Table 2.1: Different applications for Serious Games in Healthcare (extracted from [12])

The healthcare industry has a wide variety of applications for Serious Games. The following sections detail some of those application areas.

2.4.5.1 Training and simulation

Serious Games have a lot of different applications for training and simulation, essentially for surgery simulations. Positive impacts are being reported regularly, such as the positive relation between experience in video games and performance in training medical students for surgery[32][33][15]. This particular positive effect is assigned to the 3D perception experience that comes from playing 3D video games. With these positive outcomes in mind, developers are creating video games specifically for training other important skills for healthcare students, such as communication skills. Figure 2.7 shows JDoc[65], a Serious Game that is designed to train and teach junior doctors interpersonal skills, communication skills, medical information and decision making skills. This game has the goal of acquainting junior doctors with the daily stressful events of working in a hospital environment, as well as reducing the time senior doctors have to devote to training junior doctors.

Figure 2.7: JDoc screenshot (Extracted from [65]).

Serious Games for healthcare are not exclusive to training doctors. Vidani[66] proposed a game for training EMS (Emergency Medical Services) nurses for emergency situations. This game aims to teach the trainees to, given an emergency situation, decide what tasks to do, in what order and timing them correctly, to successfully complete each procedure. It also shows the trainees how the tasks are correctly performed, through realistic animations of the computer-controlled game characters. Another example of a game that aims to help healthcare students' training is Cancer Space[67] (see Figure 2.8). This game aims to facilitate cancer screening and consequently increase cancer-screening rates in qualified health centers. Cancer Space's design encourages self-directed learning by presenting the players with real-world situations, about which they must make decisions similar to those they would make in real clinics.

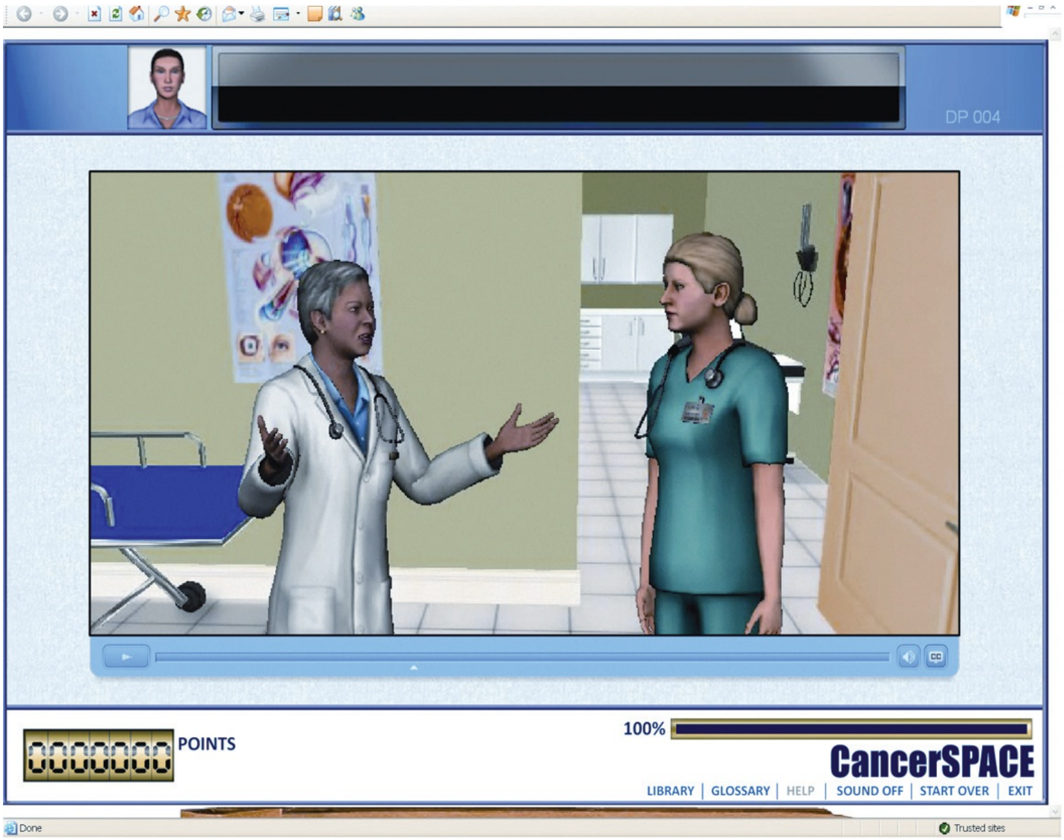

Figure 2.8: Cancer Space screenshot (Extracted from [67]).

2.4.5.2 Rehabilitation

Serious Games can be used to accelerate the recovery process for certain operations and conditions, such as therapy following stroke and traumatic brain and spinal cord injury[68], among others. In this case, video games often use special devices that can capture the players' motion (through accelerometers, for example), so that the players use their bodies to interact and consequently exercise their affected body parts. There are already a lot of applications for rehabilitation Serious Games, such as:

- Stroke rehabilitation[69]
- Balance training[70]
- Wheelchair mobility[71]
- Parkinson's disease[72]
- Orthopedic rehabilitation[73]

Researchers indicate that the reasons behind the success of Serious Games for rehabilitation purposes lie in some elements that are inherently present in video games, such as repetition, providing feedback and motivation[68]. These applications have proved to be successful, due to the fact that by playing games, rehabilitation patients distract themselves from the rehabilitation process itself, focusing instead on playing the game and having fun. But at the same time, the game leads the patients to do repetitive movements that will ultimately work as rehabilitation or physiotherapy. This can also be applied to Parkinson's disease patients, since healthcare recommends them to do regular exercise to avoid losing mobility[72]. The fact that video games make this process less cumbersome is a great achievement, since patients are often prescribed home exercises, in order to fully recover from their injuries or deal with diseases like Parkinson, but rarely perform them[74]. Therefore, finding a motivational factor for these patients is essential for them to achieve a better recovery. Figure 2.9 shows an example of a Serious Game developed to ease upper-limb rehabilitation[69].

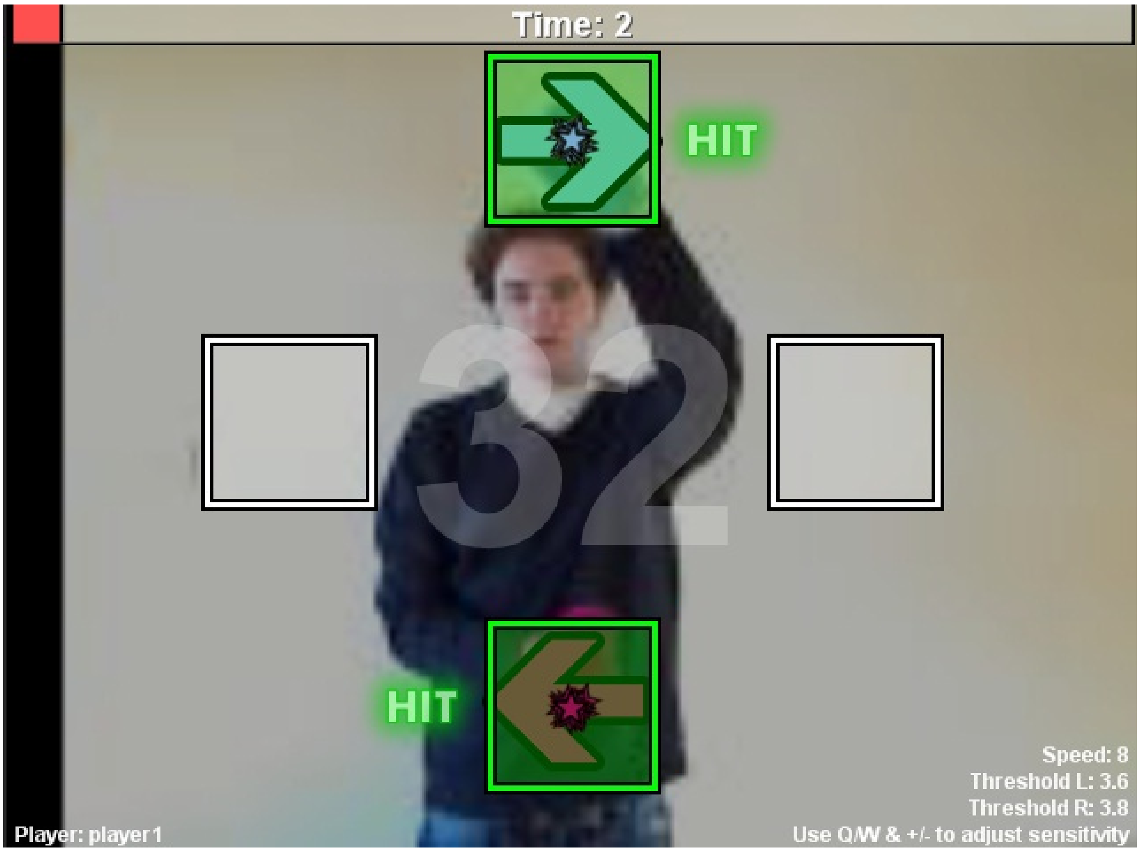

Figure 2.9: Serious Games for Upper Limb Rehabilitation Following Stroke (Extracted from [69]).

2.4.5.3 Physical Fitness

Games for physical fitness are quite popular nowadays, especially on the Wii Console with Wii Fit[75]. Despite the common knowledge that people should do exercise, in order to maintain healthy lifestyles, some people just don't enjoy doing it. However, looking at the success of video games, such as the above mentioned Wii Fit, it becomes obvious that an engaging and fun experience might be what it takes to get these people to exercise themselves. Research backs these claims and adds that the engaging elements present in video games, make exercising appealing to people who are not sports enthusiasts[15]. Games like 3D Coach[76] incorporate the fundamental elements of the coaching process into a comprehensive package, which enables trainees to practice their skills in controlled virtual environments, with the help of stretching and warm up games. Additionally, Serious Games for physical fitness improvement are now being developed with not only trainees in mind, but also fitness coaches. For instance, within StoryTec fitness coaches can define fitness programs, as well as other game-related applications and provide them to players within the game[77]. An example of a 3D coach Serious Game can be seen on Figure 2.10.

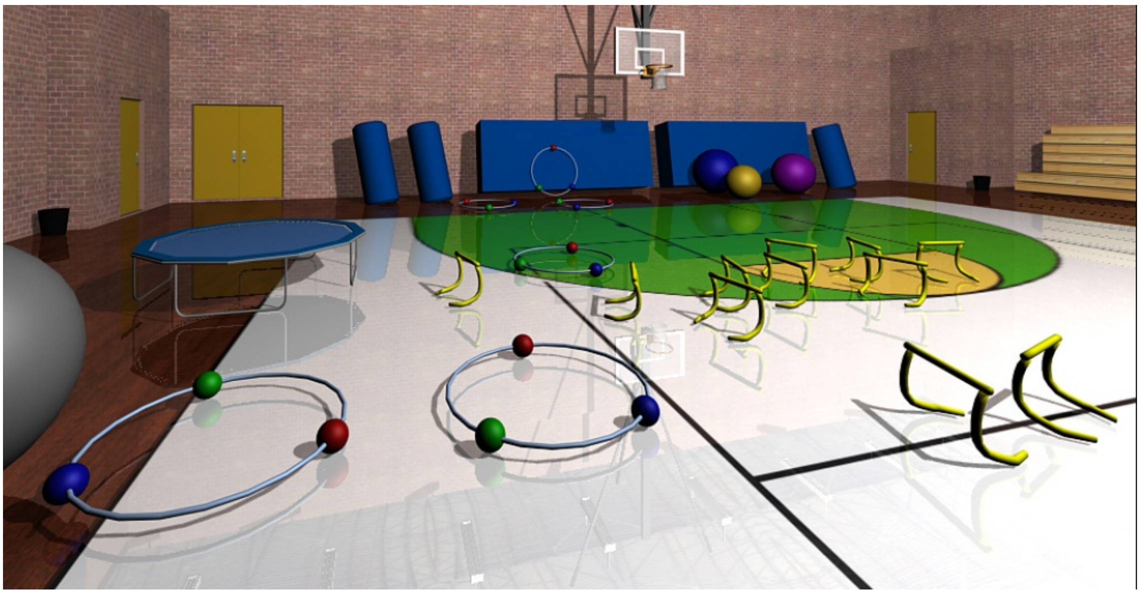

Figure 2.10: 3D Coach virtual environment (Extracted from [76]).

2.4.5.4 Education in health

Education in health is an acknowledged area for Serious Games, with several international entities behind it. For instance, the Danish National Food Institute has funded a Serious Game project, in order to motivate children to register their food intake. This game will ultimately serve as a survey on the eating habits of Danish children[78]. Educating children in how to eat healthy is becoming increasingly important, since obesity and overweight problems are getting worse. The overweight population under the age of five has now reached 42 million children worldwide[79]. Taking into account that many of these children have access to and spend time playing video games, Serious Games developed for obesity prevention and teaching good eating habits seems like a strong approach to address this overweight problem. Figure 2.11 shows a game developed to address this issue.

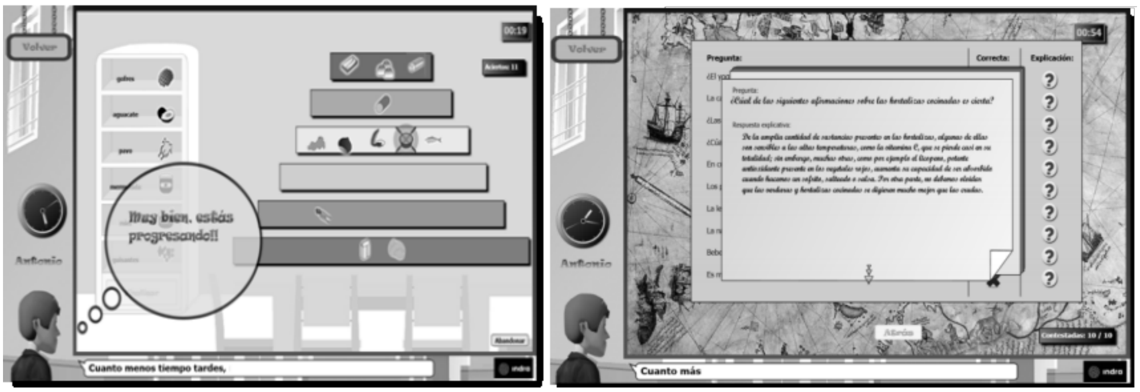

Figure 2.11: Obesity Serious Game (Extracted from [79]).

Other important issues related to healthcare education are also being addressed through Serious Games, such as raising awareness of deadly diseases like HIV. The HIV Game[80] is a video game that proposes to inform adolescents to the dangers of this disease. Within this game, players obtain information about the risks of transmission, basic sexual anatomy and common misunderstandings, among other related subjects.

2.4.5.5 Mental Health

Serious Games application to mental health problems is very diverse, ranging from memory training in elder people, to hyperactivity or post-traumatic stress disorder. For instance, elder people can benefit from video games that lead them to use their cognitive skills to solve problems. On a different note, hyperactivity problems are often due to attention disorders. Serious Games are now attempting to garner the attention problem by including attention cues[81]. Playing video games is an especially engaging activity, but it is also a demanding cognitive experience. Therefore, in order to capture the attention of players and particularly hyperactive players, video games need to imply attention cues (visual or auditory), so that players don't get overwhelmed by so much information[81]. Post-traumatic stress disorder is yet another important issue that is being addressed by Serious Games developers and researchers. This is a common issue for the military and their families, but it's not exclusive to them. However, most of the research and development of Serious Games addressing this issue is done for the military. Enduring combat stress and family separation are among the reasons why soldiers suffer from these conditions[82]. By deploying Serious Games on the field, the military have a cheap and effective way of dealing with these issues. As an example, Virtual Iraq[46] is a game directed to combat veterans who suffer from these exact problems and is already being used by the US government (see Figure 2.12 for an example of Virtual Iraq being used by the military).

Figure 2.12: Virtual Iraq - A Serious Game that deals with military post-traumatic stress disorders (From [46]).

Obviously, the healthcare industry is one of the domains with greater potential for

Serious Games, whether financially or in terms of application diversity. Hospitals, clinics, private practice physicians, therapists and personal trainers are just some of the diverse stakeholders in this market.

The next chapter includes a discussion regarding technology for Serious Games and how the different available technologies can be adapted to our goals.

Chapter 3

Development Platforms

Nowadays, video game technologies are abundant in both quantity and quality and target various platforms, from mobile to desktop and even the web. The aim of this chapter is to present several game technologies or engines and compare their main features, in order to choose which of them serve our purposes better.

3.1 Game Engines

Game engines are systems designed to ease the development of video games, by supplying several built-in functionalities that prevent developers from spending time developing them and more time focusing on building the game itself. Most game engines provide core functionalities that range from rendering, physics, sound, scripting, animation, artificial intelligence and networking, just to mention a few. These tools form an invaluable part of the gaming industry and allow developers to achieve greater goals in less time.

The next subsections present some of the most relevant game engines available at this time.

3.1.1 XNA

XNA[83] is a Microsoft technology that was a result of attempting to bring the C++ DirectX API, over to .NET and to the C# programming language[84]. Basically, XNA tries to map the DirectX API onto .NET, while at the same time providing a game development environment, complete with content pipeline, differentiating itself from solely graphics programming API's such as DirectX or OpenGL[85]. XNA's content pipeline is rather powerful and greatly simplifies the process of dealing with game content. In a full-fledged 3D game, the developer has to manage tasks, such as loading audio, graphics, 3D models, among other game contents. It is necessary to know where the content is located, how the game is going to interpret a large variety of content types and formats, along with other tasks. The content pipeline is extremely helpful in this particular issue, since it simplifies the whole process and also because it is ready to process several well known file formats, which include .x, .fbx, .fx and .mp3, among several others. The content pipeline procedure is mainly composed of 3 stages (see also Figure 3.1):

- Content importers interpret the content and transform it into a recognized format.
- Content processors interpret this recognized format.
- Finally, content compilers compile them to be ready for use.

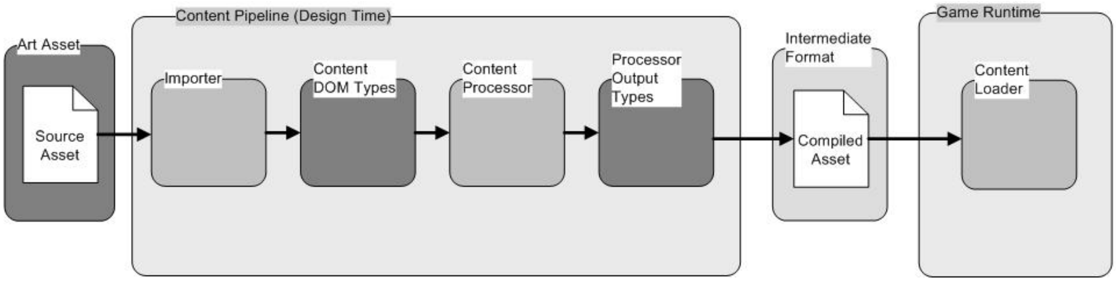

Figure 3.1: XNA's Content Pipeline (Image extracted from [83]).

One of its greatest features is how simple it is to use, in comparison with other programming interfaces for both PC and consoles, mainly due to the abstraction level of its structure. XNA provides both the same integrated development environment (XNA Game Studio) and the same framework (.NET) for the development of video games whether for Windows, XBOX 360 or Windows Phone 7, assuring a great level of compatibility for all platforms. However, both the XBOX 360 console and the Windows Phone 7 device run the .NET Compact Framework, which is a limited version of .NET, thereby some functionalities that work on Windows will not be available on those devices. Figure 3.2 shows the XNA structure.

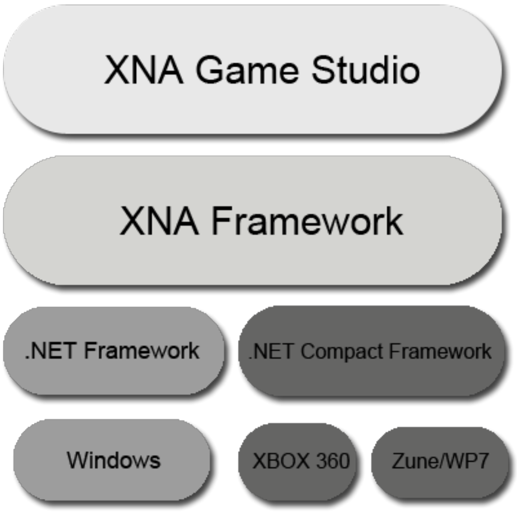

Figure 3.2: XNA's Structure.

XNA is a valuable platform, since it has the capability of allowing the deployment of video games that can be played on PC and on current generation consoles and smartphones, the XBOX 360 and Windows Phone 7, respectively. Games can be deployed on the XBOX 360 through Microsoft's distribution system, onto the XBOX Live Marketplace. Several Serious Games are already being developed using XNA such as BINX[86] a game developed with the goal of easing engineering education (see Figure 3.3).

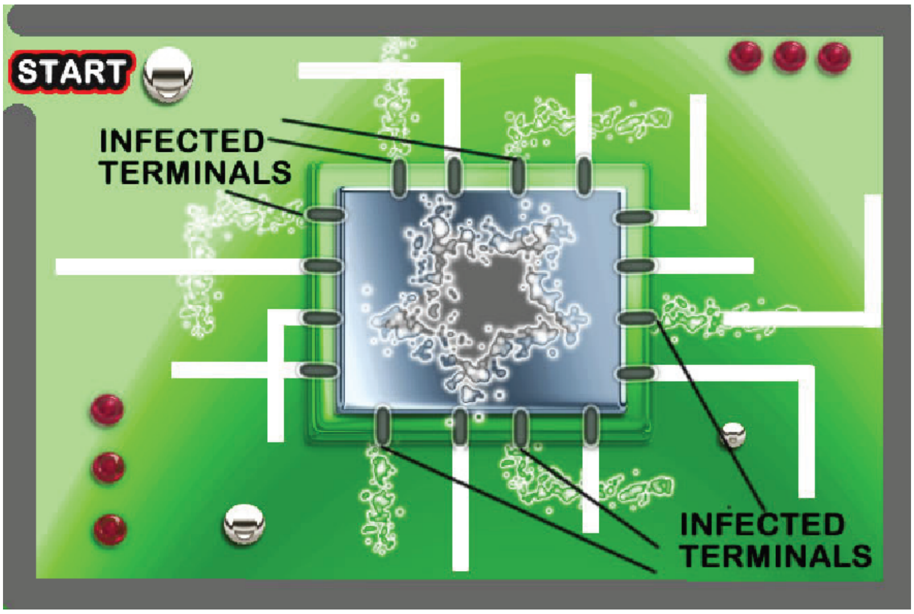

Figure 3.3: BINX: A 3D XNA Serious Game for Engeneering Education (Extracted from [86]).

3.1.2 UDK (Unreal Development Kit)

UDK[87] is a professional 3D software development framework, based on the Unreal Engine 3. It is one of the most powerful and most used game development tools in the industry and allows the creation of video games for PC and iOS platforms, such as the iPhone or iPad. UDK provides powerful and yet low-cost development, since it is available for free. However, it uses a custom internal scripting language, the UnrealScript, which, unlike other platforms like XNA or Unity 3D, requires the developers to learn a new programming language that they won't be able to use elsewhere. Nonetheless, the UnrealScript is a powerful high-level programming language and so it's not too cumbersome to learn. These scripts can be arranged and associated to objects with ease, through a visual editor called Unreal Kismet. UDK also provides a powerful 3D object editor, with texture and level editors, built-in standard objects and effects, such as trees, clouds and water, among other features. UDK is also great at providing high-quality physics simulations. Figure 3.4 shows an example of UDK's development environment.

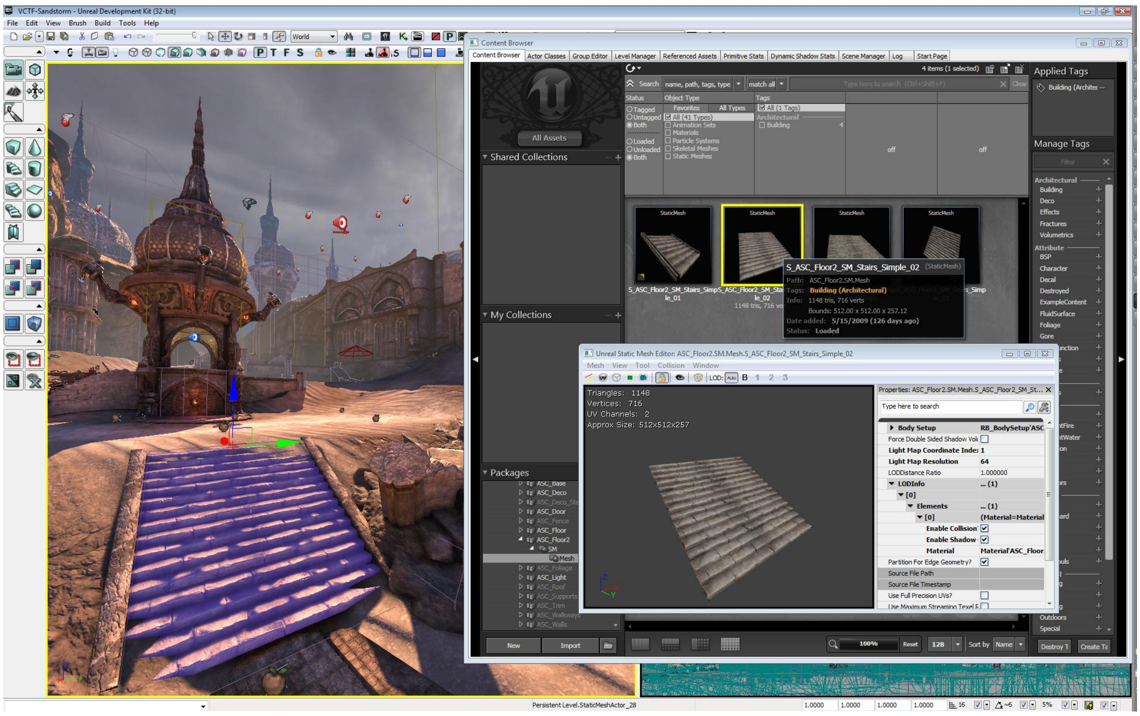

Figure 3.4: UDK - Unreal Development Kit (Extracted from [87]).

With the help of NVIDIA's PhysX[88], developers can easily manage collisions, soft body physics, rigid body physics, as well as dynamic fluid simulations and other complex behaviors. But yet again, in order to achieve this, the developer has to learn a new tool called Unreal PhAT. UDK also has a particle system editor, called Unreal Cascade that allows real-time pre-visualization of any given effect, before actually implementing it into the game.

Evidentially, the Unreal Development Kit is proving to be a very powerful tool for video game development, with sophisticated graphical capabilities and a plethora of specialized tools that enhance the development of quality games, but requires a large effort by developers to master these several tools. Therefore, this engine might represent a stepper learning curve, but may also present much better results in terms of quality. Figure 3.5 shows an example of an interesting Serious Games developed with UDK, to help players cope with diabetes management[89].

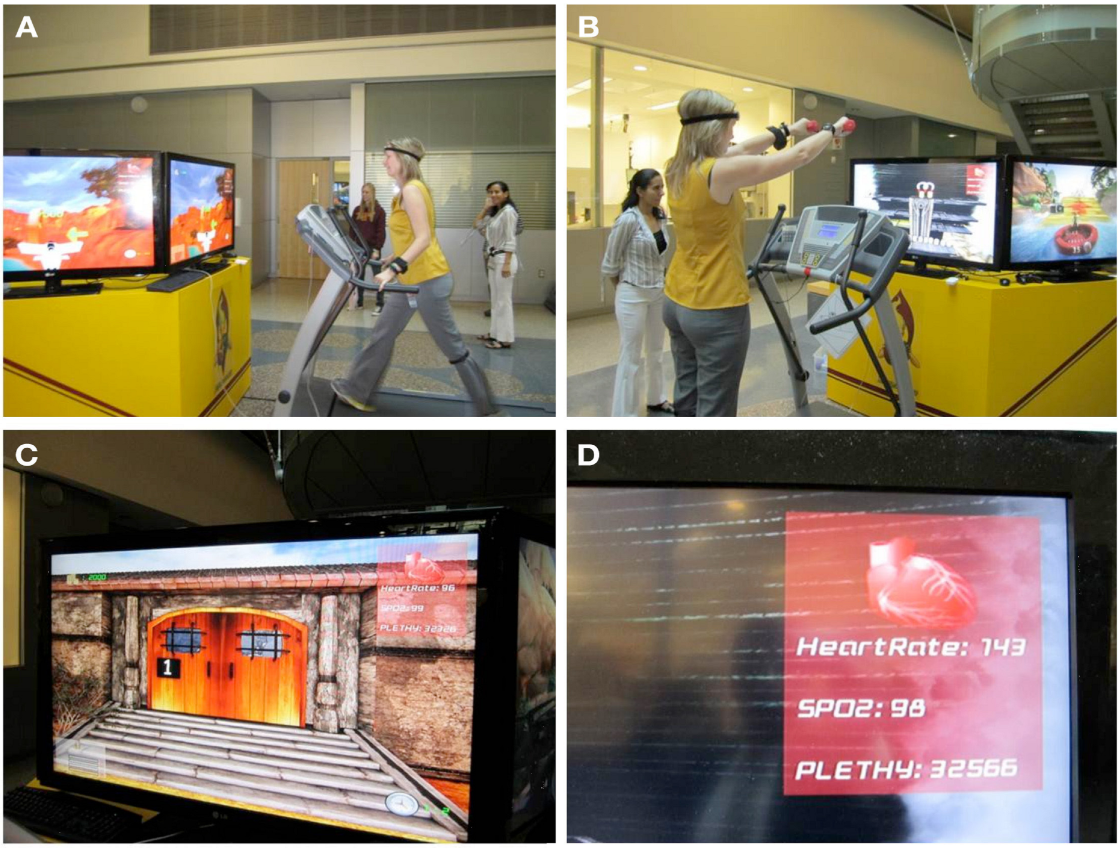

Figure 3.5: Several stages of a game used for diabetes management (Extracted from [89]).

3.1.3 Unity3D

Unity3D[90] is a powerful cross-platform engine for game development. It supports development across a great range of platforms: XBOX 360, PlayStation 3 and Wii Console, PC, iOS devices, Android and the web browser. Unity3D is an increasingly popular engine, with over 250.000 customers in the world[91]. One aspect that contributes to such a wide developer community is its resource store, a place where developers can share or sell their Unity3D resources to other developers. Figure 3.6 shows an example of the Unity3D development environment.

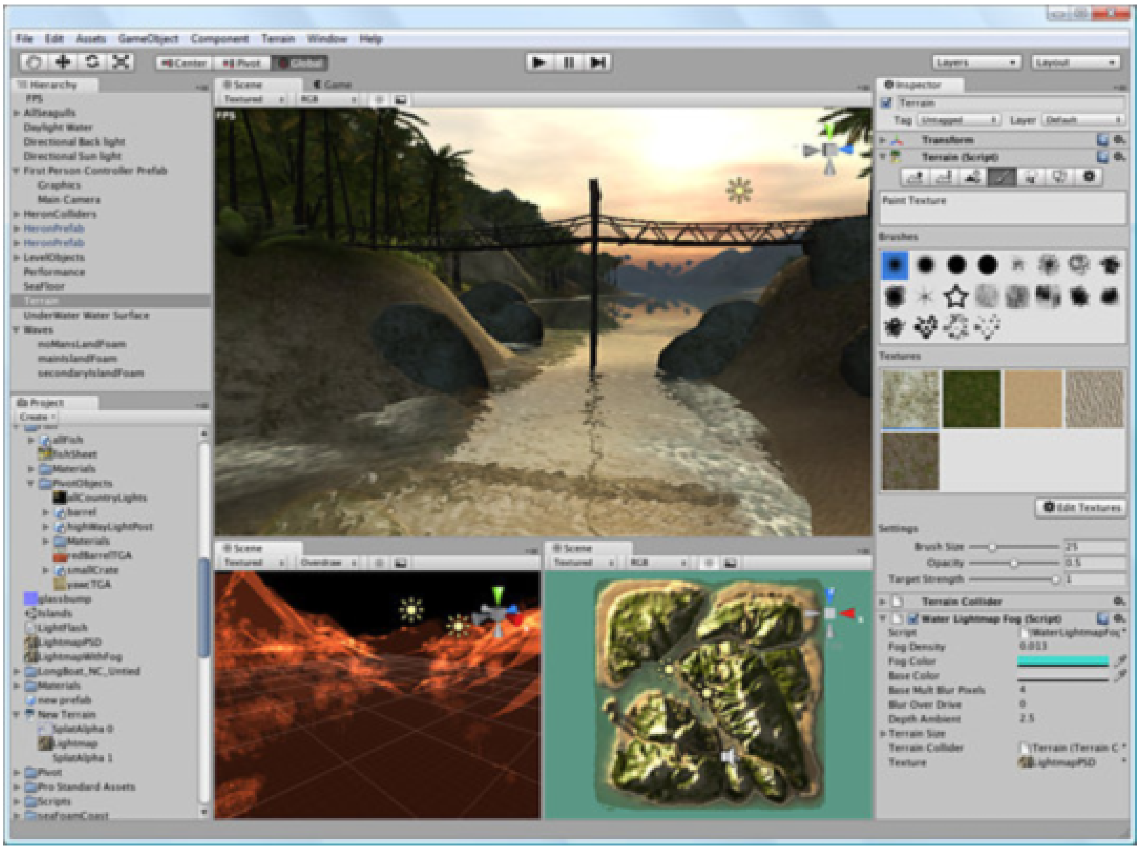

Figure 3.6: Unity3D - Development Environment (Extracted from [90]).

When comparing to UDK, Unity3D has the advantage of supporting several well know programming languages, instead of forcing the developer to learn a custom one. Unity3D supports JavaScript, C# and Boo (similar to Python) as scripting languages. Although scripting is often considered a less effective approach, scripts in this platform run almost as fast as the C++ language, which is the benchmark for video game languages in terms of performance. However, in order to get such a good performance, developers are required to have high quality scripting skills, otherwise the result won't be as good. All of the scripting languages mentioned above can be mixed together and have equally good performances.

Unity3D allows developers to use Visual Studio[92] as IDE, which is a great feature, since Visual Studio is a very powerful tool, either for writing clean code or for debugging purposes. Unity3D provides all of the key elements required to create a quality game, such as effective skeletal mesh animation, advanced lighting, physics simulations and particle systems. The physics simulations in particular, are based on the PhysX engine, similarly to the UDK engine. This engine provides rigid and soft body physics, ragdolls, among other standard physics simulations. Unity3D also contains several built-in effects (e.g. rain and smoke), and shaders (e.g. Diffuse, Bump Mapping), in order to ease the developers work and help them avoid wasting time "reinventing the wheel". In addition, the developer can build custom shaders and effects to improve the game's graphical quality. Figure 3.7 presents a Serious Game that uses the Unity3D game engine in combination with motion capture technologies, in order to achieve motor rehabilitation for patients[68].

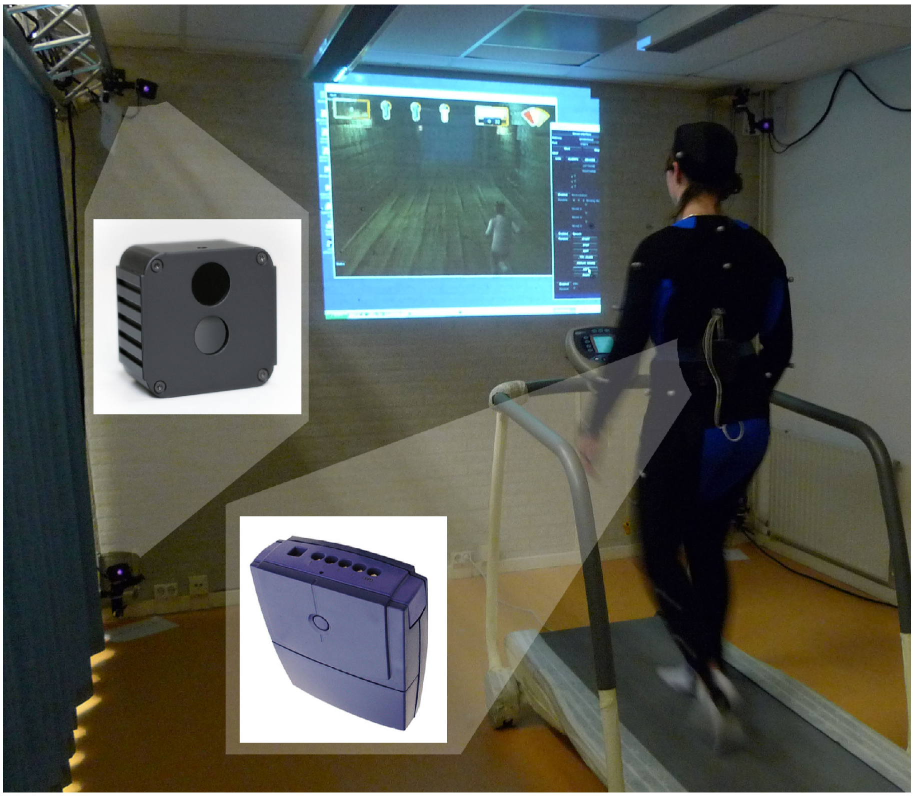

Figure 3.7: Serious Game for motor rehabilitation using Unity3D (From [68]).

Video games developed with the Unity3D framework can be published and played online, with the help of the Unity Web Player plug-in, hence providing an easy way of distributing games. Unity3D is yet another powerful tool for video game development, with the advantage of providing high quality features, with an equally high level of flexibility in terms of not only programming, but also distribution wise.

The next section approaches the platforms that were targeted for the deployment of the Serious Games developed in the context of this dissertation.

3.2 Targeted Platforms

For the development phase of this dissertation, a large range of platforms were initially targeted (mobile, desktop and web platforms). In this section, several of those platforms are approached, in order to discuss which of them are more suitable for creating and deploying compelling Serious Games.

3.2.1 Mobile Platforms

Mobile platforms are widely spread and provide easy distribution of games and applications through each platform's distribution system. Also, gaming is nowadays widely spread across these devices. In 2009, Apple announced that their iPhone[93] device had more than 20.000 games or game-related applications, whilst Sony's PSP[94] and Nintendo's DS[95] combined, only had a little over 4.000 games available for their customers[96]. Additionally, the current generation of mobile devices supply accelerometers and touch interfaces, which contribute to innovation in terms of interaction and allows new types of games to be designed, without the need of new hardware. For instance, DroidGlove[73] is a Serious Game for the Android platform[97], which uses the device's accelerometer to try to enhance wrist rehabilitation processes. However, mobility has its shortcomings, such as low processing power or battery life. Hence, games have to be shorter, less compelling graphically and they can't have so much content, thereby restricting Serious Games developers on a great level.

In order to decrease some of the shortcomings associated with video game development for mobile phones, only the most powerful current generation mobile phones were targeted. This group of mobile phones is composed of Google's Android[97], Apple's iPhone[93] and the newcomer Microsoft's Windows Phone 7[98]. However, the most successful of them are undoubtedly the Android and the iPhone platforms, which gather 37% and 27% of the smartphone market, respectively. Windows Phone 7 is still a rather new platform, thereby currently having a much smaller market share. Notwithstanding, it is a powerful platform with a lot potential, so it will also be targeted as possible development platform. Figure 3.8 contains a graphic that details the smartphone market share, as of March of 2011 in the United States of America[99].

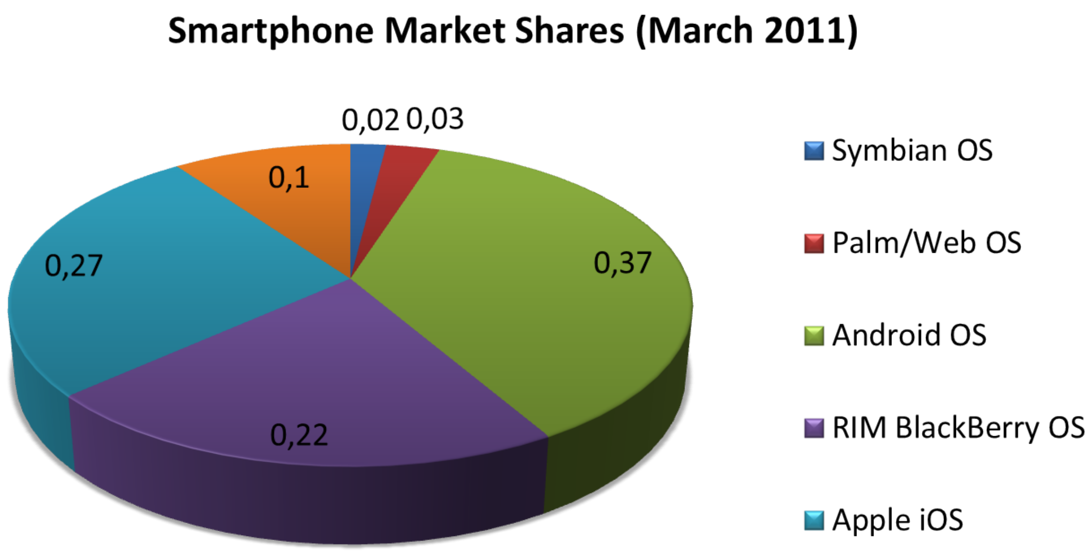

Figure 3.8: The Nielson Company's report of March's Smartphone market share in the US (Extracted from [99]).

Each of the chosen platforms is strong in their own right and provides both development tools and features required to develop Serious Games. Despite its large market share on the US, the RIM BlackBerry OS was overlooked, since it fails to display the visual quality that the other chosen platforms can. As far as mobile technologies are concerned, the Android, iPhone and Windows Phone 7 mobile phones provide the best experience possible and their games are usually on par technically. However, there are some general differences in what concerns application development, device target and application distribution. These differences are exemplified on Table 3.1.

Criteria		Android	iPhone	Windows Phone 7
IDE		Eclipse (recommended)	Xcode	Visual Studio/XNA Game Studio
Operating Systems for Development		Windows, Mac OS X, Linux	Mac OS X	Windows
Programming Language		Java	Objective-C	C#
Mobile Devices	**Phone**	Available in most countries	Available in most countries	Only available in selected countries
	Similar Media Device	None	iPod Touch and iPad (Worldwide)	Zune (US only)
Programming Interface		Key high-level abstractions	Mainly low-level standards, such as OpenGL ES	High level abstractions
Distribution		Android Market	iTunes Apple Store	Microsoft MarketPlace

Table 3.1: General comparison between mobile platforms.

Evidentially, each of the three platforms provides similar options in terms of development. Nonetheless, the iPhone may represent a stepper challenge in terms of development, due to the low-level programming with OpenGL. But with the various game engines available today, this issue is easily surmountable, which pretty much puts the iPhone one par with the other platforms. In terms of deployment and target audience, the Windows Phone 7 clearly loses in comparison with the other two. It is available only in selected countries, wherein the Android and the iPhone are widely available worldwide. So currently, the best mobile platforms available for the development of a Serious Game seem to be the iPhone and Android.

Regardless, gaming on mobile phones is still being criticized, due to the interactions based on the touchscreens. Even though most of these new mobile devices have rather large screens, they still carry the problem of screen occlusion. This issue is particularly prone to happen in touchscreen devices, since users interact by touching the screen, which most of the time results in covering important information displayed on it[100]. Accordingly, research results reported by Zaman[96] indicate that the learning curve required by touch-based games is much stepper, with game completion on touch-based games being 42% slower, when compared to traditional controls. Also, the same study reported that 150% more character deaths were registered on touch-based games, when comparing to games that use physical buttons for interaction. These issues, along with the lower processing power and device battery problems, raise some concerns regarding the adequacy of mobile phones for the creation of Serious Games.

The next section, details the desktop and web platforms which seemed more capable of filling the requirements for the goal we aim to achieve: developing compelling Serious Games that can reach a large number of people.

3.2.2 Desktop and Web Platforms

Desktop technologies have the advantage of providing high quality products, in terms of graphics, sound and artificial intelligence, among other features. This is due to the largely superior processing capabilities that desktop hardware provides. However, they have some drawbacks. For instance, desktop developers have to accommodate a greater number of combinations of hardware, which greatly increases the development time. Also, desktop video games and applications in general, require either a physical format or digital download of rather large files and big installation times, before the player can actually experience the game.

On the other hand, the obvious answer in terms of facilitating accessibility for video games is to use the Web. However, for over a decade, rich 3D content on the web has only been available via external browser plug-ins, which limits the accessibility that we aim to achieve. Also, 3D web technologies are yet to achieve a definitive state, likely due to the high requirements for 3D graphics to be displayed in terms of computational power, especially in a web context.

The first technology to be developed was VRML[101] (Virtual Reality Modelling Language), which is a text file format used to describe 3D scenes as a set of geometry and materials. This technology was later replaced by X3D[102] and requires the installation of a platform specific plug-in. Over the years, other 3D web technologies emerged, but more relevantly Google also made a move onto 3D graphics development for the web, with their 3D graphics engine, O3D[103]. O3D aims to integrate itself into the web browser, while extending JavaScript with 3D capabilities, with the help of both DirectX and OpenGL. However, O3D was still deployed as a web browser plug-in. Adobe Flash[104] is also increasingly adding more and better 3D capabilities to its Flash plug-in. These 3D capabilities can be accessed through the ActionScript[105] programming language. However, this approach is still based on transforming 2D objects into 3D content.

All of the above technologies provide access to 3D graphics in a web context, but only by using plug-ins. These plug-ins provide an effective way of reproducing 3D content on the web, but also diminish the accessibility that the web inherently supplies, since they have to be ported and compiled for each target platform and also because they require an installation by the end-user. Fortunately, a new standard has emerged to address this issue, WebGL[106].

3.2.2.1 WebGL

WebGL is a standard specification for JavaScript bindings to OpenGL, that is under development by the Khronos Group[107]. WebGL is a cross-platform graphics API specification for the Javascript programming language that enables the web browser to access the GPU without the need of any plug-in. It is based on OpenGL ES 2.0 and uses OpenGL shading language GLSL, hence offering an API similar to OpenGL. Also, due to its complete integration into the web browser, WebGL applications can take advantage of DOM (Document Object Model) and the Javascript infrastructure, which are fundamental to HTML documents. WebGL is exposed through the new <canvas> element in HTML5[108] and is essentially just another rendering context within this element, so it can be easily combined with HTML, as well as other web content. Other efforts have been made to develop canvas based 3D graphics libraries for web browsers, like Glypher3D[109] and Canvas 3D[110], but with the standardization process of WebGL these efforts lost some relevance.

Nowadays, every major modern web browser supports Javascript. Also, WebGL is currently being released on the latest versions of all main web browsers, such as Firefox, Chrome and Safari(with the exception of Internet Explorer), which means that WebGL can in principle run on any system (as long as its graphics cards supports OpenGL ES 2.0 or better). It provides hardware accelerated 3D content for the web, therefore, making it a great candidate for a cross-platform game development environment, which can be easily accessed by users. There are already a lot of libraries that can be used to help ease the development time for WebGL applications, such as GLGE[111] (see Figure 3.9), SpiderGL[112] or even O3D which recently made a shift to WebGL, becoming an engine for it, using Javascript.

Figure 3.9: Frankie Demo powered by the GLGE engine for WebGL (Extracted from [111]).

These capabilities ensure means for the creation of new and innovative web based applications that were previously exclusive to desktop applications, such as Serious Games. However, the content quantity and also quality is still an issue if compared with traditional desktop video games or applications. Content on web games is still restrained, not only by low bandwidth, but also by low specification (for instance, WebGL uses OpenGL ES 2.0, which is commonly used by mobile platforms, such as the iPhone). Current generation video games easily surpass 5 or 6 gigabytes of data, which is impossible for web games to achieve.

3.3 Chosen Platforms

After doing research on all of the above platforms and game engines, a decision had to be made regarding which platforms and engines to use for the development of Serious Games, in the context of this dissertation. I chose to use WebGL for the development of a Serious Game, since it showed a lot of promise in respect of both 3D capabilities

(OpenGL ES 2.0) and end-user accessibility (web deployment without the need for plug-ins). WebGL provides all of the necessary tools required for developing quality video games and Serious Games in particular. It also allows for video games to be accessed effortlessly, within the web browser. Also, if WebGL actually manages to become a web standard, it might be a turning point for web development with 3D web becoming a reality. The web might also be helpful on gathering user feedback, since information retrieval is simpler to obtain with web applications than with typical desktop ones.

Regardless, both the will to create a more visually compelling and feature complete Serious Game and the will to participate in an international Serious Game competition, lead to the decision of creation another Serious Game using the XNA technology. XNA is short on features, when compared to UDK or Unity3D, but still manages to provide a great C# API on top of DirectX and serves our purpose of designing a feature complete and more graphically compelling Serious Game. By using XNA we can also participate on the world's biggest software competition for students, Imagine Cup.

Chapter 4

Developed Work

Designing educational games is a complex task that requires knowledge in different areas, such as game design, education and the subject matter of the game. Also, in order to develop games that can be deployed in every culture, the games' themes have to be universal. For those reasons, the developed video games aim for teaching relatively simple and universal themes, such as human body functions (circulatory system) and basic mathematical operations (sum, subtraction, multiplication and division).

The work developed for this dissertation comprises of two 3D Serious Games. One using a web technology (WebGL) called Oxyblood and another using a desktop technology (XNA) named Rocky - The Math Cat.

4.1 OxyBlood

Oxyblood is a Serious Game that aims to teach young students how the human body works internally, more specifically how the circulatory system works. The development of this game has resulted in one published short paper on an international conference: SGamePlay 2011[113]. An additional full paper has been submitted for another international conference (ACE 2011) and is awaiting approval.

4.1.1 Designing OxyBlood

The goal of this project was to create a Serious Game that would teach young students the basic functioning of our circulatory system. The human body circulatory system contains a vast amount of actors, but in order to adapt the process to the level of young students, some simplifications had to be made. To effectively simplify this process, but at the same time maintain the main functions of blood flow, only the trajectory of red blood cells and their goal of delivering oxygen were considered as main goals for the game. However, other elements present in this process were also included, such as white blood cells, platelets and bacteria.

The next step in designing this game was to choose which game genre was a better match for the game's purposes.

4.1.1.1 Choosing a Game Genre

In order to further design this game, a decision had to be made regarding the type of game it would be. Thus, it was imperative to determine what actors would be part of our game. It was established earlier that the actor group would be comprised of the following entities: red blood cells, white blood cells, platelets and bacteria. The human body contains a large amount of each of these elements and even though some simplifications are required, there is a certain level of fidelity to the actual process that needs to be maintained and therefore there needs to be a rather large quantity of these elements present. So, the game would have to deal with groups of various entities, something that is very common to real-time strategy games.

Real-time strategy (RTS) games are video games where the player has to control a group of entities, generally troops or civilization members. They seem appropriate for the implementation of our game, since they are a perfect example of complex multi-agent systems[114]. In this genre, each unit type has a set of goals and they derive their actions based on these goals. Each unit or each group of units is commanded by the player, but simultaneously has an automated behavior that simulates the unit's intelligence. As was established earlier in this dissertation, the strategy genre has a nice pace and provides information clearly, which makes it appealing for adaption into a learning context.

Following the above mentioned reasons, a real-time strategy game seemed the appropriate game genre to aim for. However, some adaptions to the genre are mandatory, in order to adapt that concept to our goals and develop a set of rules that emulate the blood flow process. In OxyBlood, the player has to create and coordinate units, such as red blood cells, to do different tasks, in order to successfully manage the circulatory system. The player controls a small fictional ship that coordinates all tasks and units. Throughout the game, the player encounters bases for each controllable unit, where he can create and deploy them. The player-controlled units are composed of:

- **Red blood cells** - can harvest oxygen, but are targets for bacteria. Their goal is to deliver oxygen to the body tissues, via blood flow.
- **White blood cells** - have the function of protecting red cells from bacteria.
- **Platelets** - work as healer units, for both red and white cells.

These units are controlled by the player, but must have autonomous behaviors, so that they can react to other units' behaviors without any interference by the player. For instance, white blood cells can spot bacteria units and attack them automatically. Additionally, as in any usual RTS game, there are harvest spots where the player must bring red blood cells, so they can catch oxygen. Then, the player must lead these cells to deliver the gathered oxygen to body tissues, while using the white blood cell units to defend against any bacteria attack. If any cells (white or red) get injured, the player can use platelets to heal them. The goal of the game is to successfully carry oxygen through the circulatory system.

4.1.1.2 Choosing a Graphics Style

The graphics style is very important and has to accommodate for the target audience, which is comprised of young students from 9 to 12 years old. This target audience was decided upon, due to the fact that at this age students already have some knowledge concerning the human body, but yet are unaware of the circulatory system. In order to appeal to this young audience, the game should provide a compelling graphical style that should not be too realistic, but should provide a great level of identification in the way that cartoons often do.

A cartoon look is often composed by rounded edges, vivid colors and exaggeration of preeminent features, but in video games that alone is not enough. The look and feel of a game is defined by the objects' forms, textures, motions and lighting, among other properties. There are various different methods of visually representing these features, depending on their complexity. McLaughlin[115] has identified the taxonomy for visual representation in video games, as seen on Table 4.1.

Graphical Styles	Form	Motion	Surfaces & Light
Simplified	Low detail and symbolic forms	No articulation or low fidelity	Flat shading and vector graphics
Stylized	Identifiable objects with unrealistic proportions	Articulation and magnified expressions	Curved surface shading, transparency, texture mapping.
Realistic	Photo-accurate modeling and high detail	High motion fideliy (Motion capture or physically based).	Photo-realistic shadow casting, reflections, light scattering, radiosity.

Table 4.1: Taxonomy for computer graphics in video games[115].

For this project it seemed appropriate to look for a cartoon look, therefore, based on the above table, the graphics style that resembles the most to that look, is the stylized graphics style. It is important to maintain the same style for the user interface style as well, so the UI should use rounded edges for boxes and text fonts, as well as a fun color pallet.

4.1.1.3 Assuring Learning Elements

In addition to providing an engaging game and in order to ensure an effective learning experience, it is important that players understand the game's concept, functionalities and consequently associate them to the educational subject matter. Therefore, the game design should display timely event-based information, so that the player can have a better understanding of the game. For that effect and taking advantage of the technology, video descriptions are displayed whenever players reach new and unknown aspects of the game. That way, players gradually learn new information, instead of being overwhelmed by all of it, at the beginning of the game. However, these video events should not stop the game, so that players don't lose momentum and eventually become distracted by the interruptions, which don't contribute to achieving the flow state.

Furthermore, it is important to determine whether players are actually learning anything from the game or not. To achieve this, players will be inquired about their knowledge regarding the subjects involved in the game before playing it and also after they successfully finish each level. With the help of this gathered data, knowledge improvements achieved by each player can be inferred upon.

4.1.2 Developing OxyBlood

OxyBlood was developed for the web with HTML5, CSS3[116] and the WebGL technology. A game development library was also used to ease the development of the game, GLGE[111]. This library aims to abstract the developers from the WebGL context and simplify the programming process for creating rich 3D content for the web. GLGE uses the XML format to define meshes, animations, materials and everything else that might be part of each game scene. GLGE's game scenes are managed through a scene graph, with each object of a scene being placed in a node within the respective scene graph. Currently, GLGE provides the best feature set amongst WebGL libraries for the development of video games. GLGE's feature set includes:

- Per-pixel lighting, directional lights, spot lights and point lights
- Depth shadows and fog
- Normal and environment mapping
- Parallax mapping
- Reflections and refractions
- Animated materials
- Keyframe and skeletal animation
- Collada format support (including animations)
- Text rendering and Particles system

GLGE may be one of the best WebGL libraries for game development, but it is obviously not a powerhouse when compared to the best desktop game engines or libraries. Still, it provides the necessary tools for the development of a compelling 3D

Serious Game for the web. The fact that it's possible to combine WebGL and both CSS3 and HTML5 elements, broadens the development possibilities for this game.

The following sections detail the development of each of OxyBlood's features.

4.1.2.1 3D Modeling and Animation

In order to populate our game's 3D environment, a lot 3D modeling and animation was necessary. To do so, a free open source 3D modeling tool was used: Blender[117]. Blender is a powerful tool for the creation of 3D content, with options available for 3D modeling, animating, texturing and applying materials to objects, as well as other more complex computer graphics techniques, such as creating and applying normal and bump mapping. Blender is an excellent tool for this project, since it provides quality export options for the Collada format[118], which is supported by GLGE. Furthermore, Blender allows the integration of Python scripts, which allows developers to add functionalities to it. GLGE's creator took advantage of this Blender feature and created a script that allows Blender to export an entire scene (complete with textures and lighting, among other features) to GLGE's XML scene graph format. With the help of this script, developers can design complete levels in Blender and just export them to GLGE's format. This is incredibly helpful and reduces the game's development time. Blender also has a strong community that is constantly adding functionalities for it, through these Python scripts. Figure 4.1 shows an example of Blender's development environment.

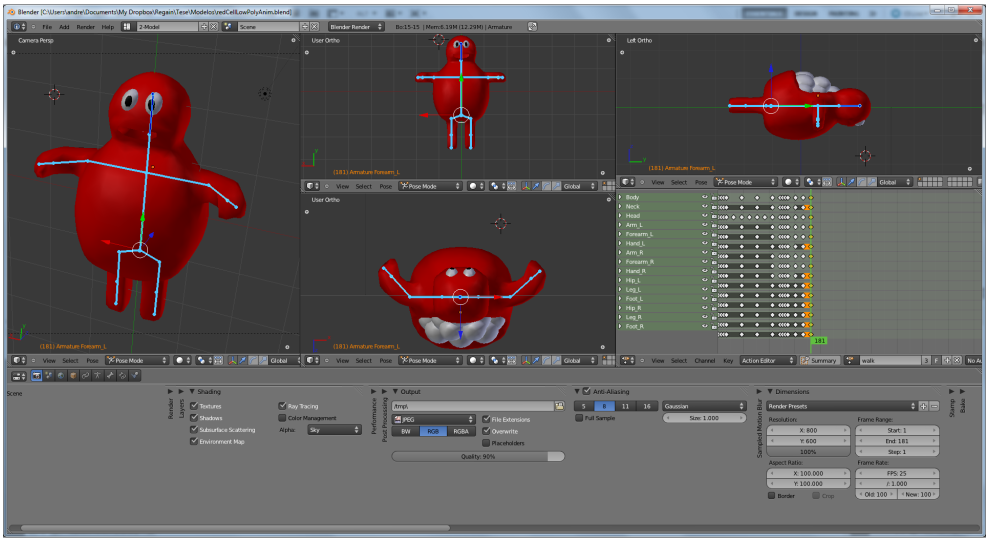

Figure 4.1: Blender - Modeling and Animating a Red Blood Cell.

The objects created for this game follow the stylized graphics style, with identifiable but yet unrealistic objects, animations and UV texture mapping, among other details. The group of 3D objects modeled for this project can be divided in static or animated objects. Objects related to the environment are static and are composed of: terrain levels, deployment bases, harvest points, drop points, among others. Other static objects are present in the game, but are not a result of 3D modeling, such as water (reflections and refractions applied to a plane mesh), oxygen sources (particle system) or fog. Figure 4.2 shows some of OxyBlood's static 3D objects that were created with Blender.

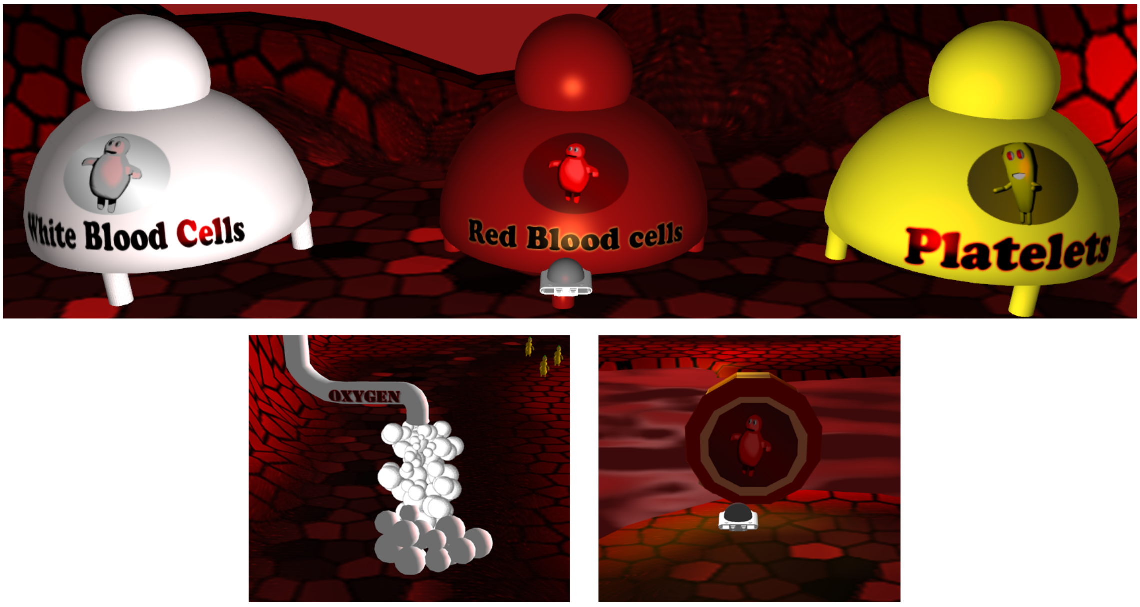

Figure 4.2: Some of OxyBlood's 3D static objects.

The stylized graphics style fits our game perfectly, not only because of the target audience, but also due to performance issues. WebGL is derived from OpenGL ES 2.0 which is mostly used on mobile devices, such as the iPhone, thus its performance is not quite on par with current OpenGL standards. Therefore, 3D models can't be too complex, especially in a RTS game which is characterized by fairly large amounts of artificially intelligent units that must behave in groups and have autonomous behaviors towards other game entities. Additionally, these units must be animated. Skeletal animations were used to achieve that. This type of animations is supported by the Collada format and can be interpreted by the GLGE library. Figure 4.3 shows all unit types created in Blender.

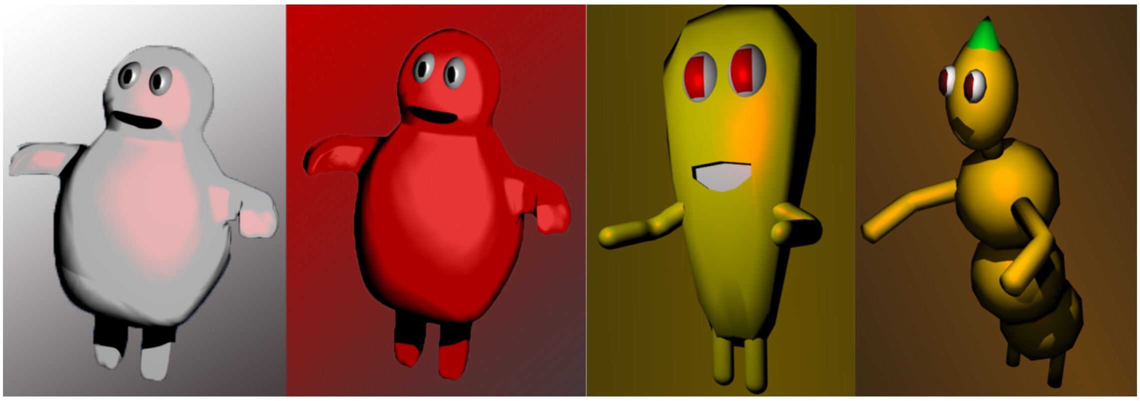

Figure 4.3: OxyBlood's characters from left to right: White Blood Cell, Red Blood Cell, Platelet and Bacteria.

4.1.2.2 User Interface

It is extremely important for the game that the user interface is both easy to learn and simple to use, since its target audience is very young. Therefore, menus, as well as other aspects of the user interface, follow a very simplistic approach, with simple and attractive components. The game's user interface was created using CSS3. CSS3 stands for Cascading Style Sheets 3 and is a language used to describe the look and formatting of documents written in markup languages, such as HMTL, XML or XHTML. CSS3 allows more flexibility in terms of web presentation and can ease the design and development of the user interface for web games, since it can be handled outside of WebGL context. Figure 4.4 shows an example of one the game's menus.

Figure 4.4: OxyBlood's start menu.

The game's menus are designed on top of the game and include transparency, so that if the player needs to pause the game or access the options menu, he/she doesn't lose sight of what was being done within the game. The menus are very straightforward and easy to understand and manipulate. In order to display information that is useful to the user in regards of game play, in a simple and intuitive way, there is a permanent information bar at the bottom of the screen (see Figure 4.5). This bar keeps track of the player's progress towards each level's goal, as well as information about the deploying units. To further facilitate the player's understanding of each unit type, a color was assigned to each of them (e.g. Red Blood Cells were assigned the red color). This way, the process of managing deployment and selection of each unit is more intuitive, as is shown in Figure 4.5, which describes this part of the user interface.

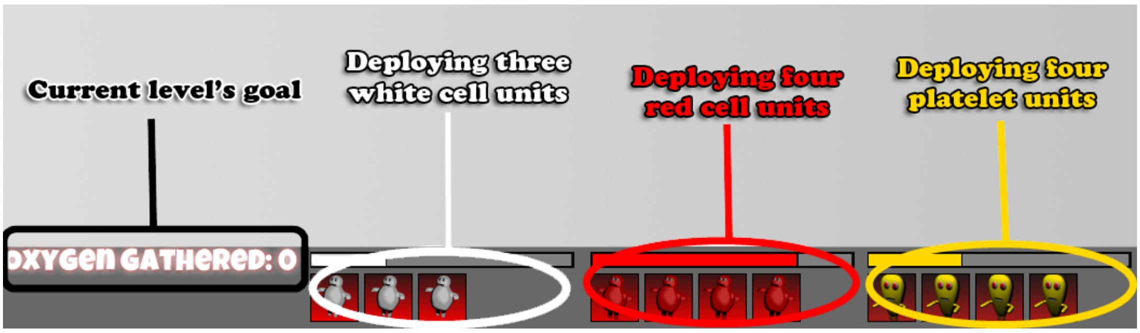

Figure 4.5: OxyBlood's user interface example.

There are also other user interface elements for displaying learning information, such as video tutorials and explanations. These elements that are used for enhancing the learning experience are approached on the next section.

4.1.2.3 Explicit Learning Elements

Similar to other video games, OxyBlood has implicit learning elements that come from learning how to play the game. For instance, by playing the game, the player learns that red blood cells are responsible for transporting oxygen through our bodies. But that is only basic information about them and there is so much more to know about that subject. Similarly, white blood cells and platelets also have more complex functions beyond those represented in the game. However, some of these functions are either too complex to simulate in the game environment or too difficult to translate into compelling game design. Therefore, some explicit learning elements are also used, in order to give the player more information about the subject they are learning.

These explicit learning elements shouldn't however prevent the player from continuing game play, otherwise they could contribute to disengaging the player and lowering the learning potential of the game. Additionally, these elements should be dynamic and related to the game, so that players take interest in them. Due to these reasons, this relevant information is displayed through video, while using footage captured from the game. This way, the game can simultaneously explain its functionalities to the player and also associate the actions done within the game to the real functioning of blood flow, as well as other relevant information on the subject matter. Figure 4.6 shows the initial video tutorial of the OxyBlood game.

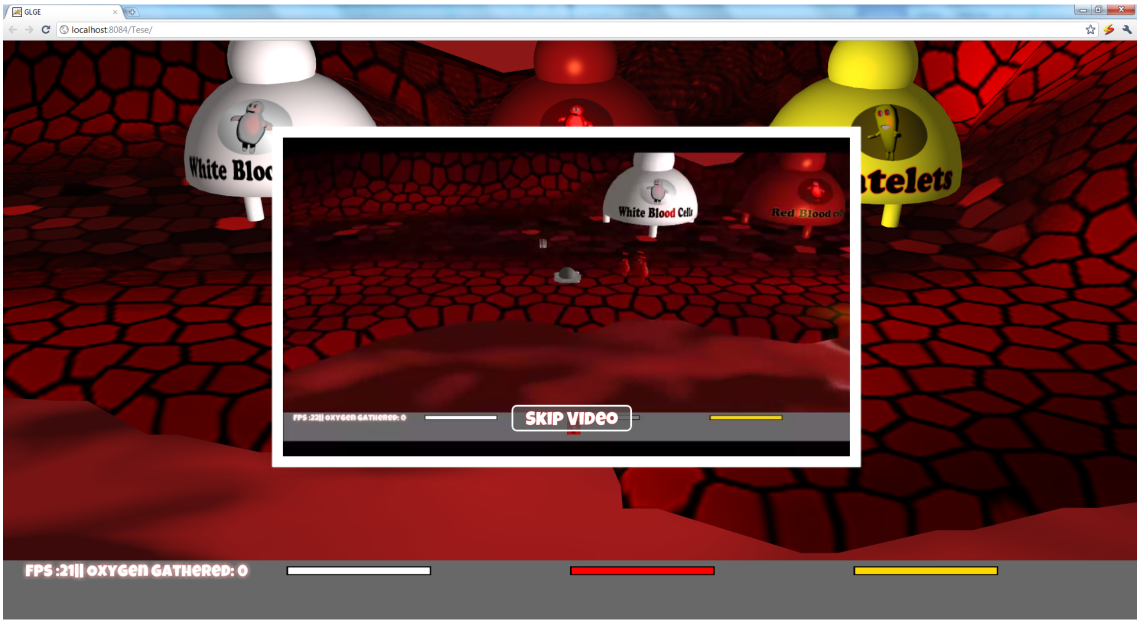

Figure 4.6: OxyBlood's initial video tutorial.

The initial video tutorial explains the basic functioning of the game, as well as the basic functioning of the human circulatory system. The player may skip watching the video if he/she has watched it previously, but the game doesn't begin until the player either skips the video or watches it until the end. However, when the game begins there are no more game-imposed interruptions, even when displaying informational videos. Due to the slow pace of game, videos can be displayed during game play and the player can watch or just listen to them without interrupting the playing process. This way, the player can stay emerged in the game, while information is being displayed, which contributes to maintaining a focused state that is known to increase the learning potential. Figure 4.7 shows the placement of these video elements during game play.

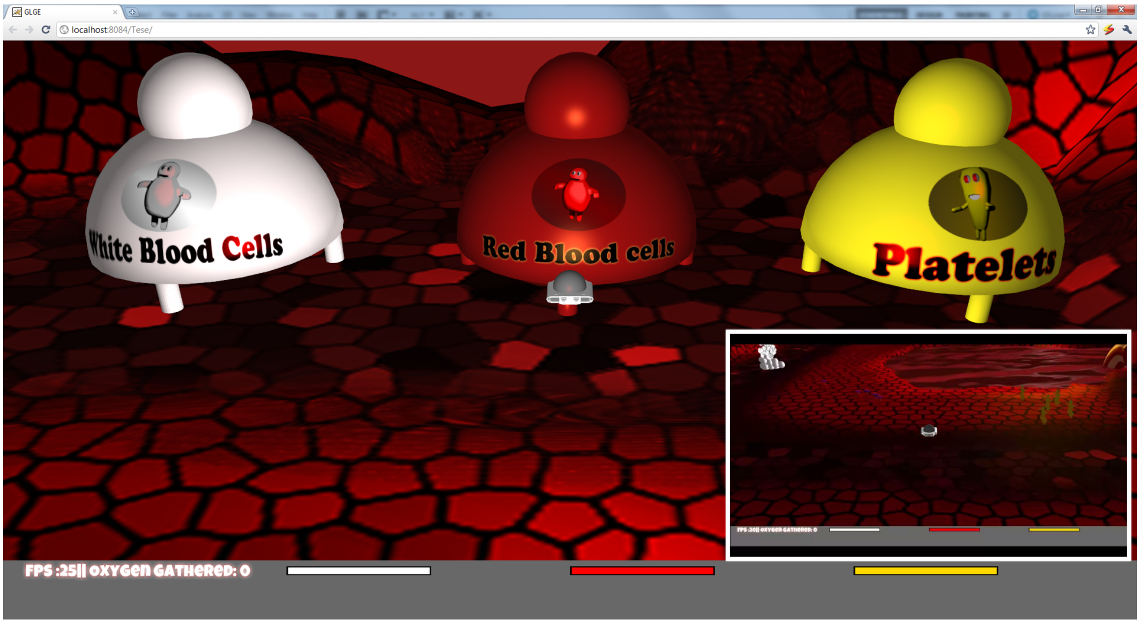

Figure 4.7: OxyBlood's video information during game play.

4.1.2.4 Artificial Intelligence

As previously stated, RTS games are known to feature autonomous characters, which can behave in groups and automatically react to other character's behavior. Autonomous characters are a type of autonomous agents that are often present in video games and have the ability to improvise their actions without player input. Each of OxyBlood's units is an autonomous character. The implementation of these behaviors is based on Reynolds' research on steering behaviors for autonomous characters[119]. In this approach, each unit follows a simple vehicle model that reacts based on steering behaviors.

Simple Vehicle Model

In this model, each vehicle is seen as a point mass. Therefore each autonomous unit is characterized by a three-dimensional position and unitary mass value. The position value is changed based on the unit's velocity, which is a result of forces applied to that same unit. In order to maintain a natural locomotion behavior, units must have limitations regarding the maximum values for velocity and forces that can be applied to them. Furthermore, the vehicles have an orientation parameter that indicates the direction that the vehicle is facing.

Each unit's locomotion is based on the forward Euler integration method. This method is simple and fast, but depends on the size of the integration step, specified by the user. Euler's forward method states that the area below the curve, between a known value of a function and the next value in time can be approximated by a rectangular representation[120] (See Figure 4.8 for an example on acceleration variation in time).

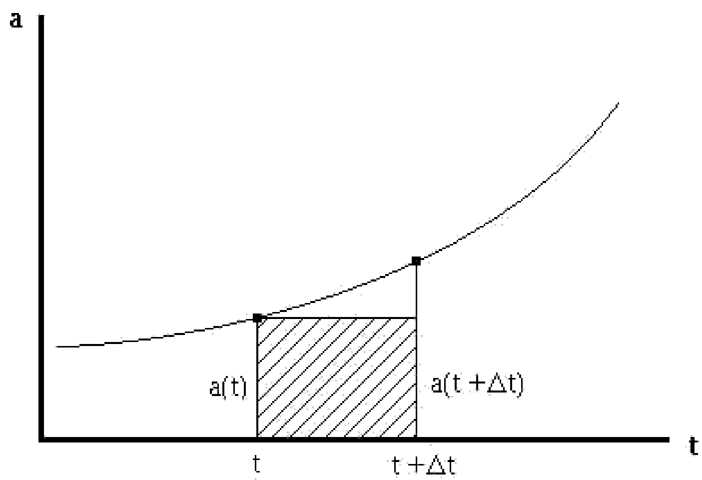

Figure 4.8: Euler's forward integration method.

Therefore, to iterate a step forward in time, it is only necessary to replace the area of the triangle for the real integral. For instance, the velocity (v) value of an object at time (t2) is calculated based the acceleration (a) (see Equation 4.1).

$$v(t2) = v(t1) + a(t1) \times (t2 - t1) \tag{4.1}$$

For OxyBlood's autonomous units, the process is similar but with a few additions. With each update step, forces (truncated to the maximum force) are applied in order to steer each unit's mass in the desired direction (see Equation 4.2). This produces an acceleration (see Equation 4.3), which is then added to the velocity in the previous time instant, to generate the current velocity (see Equation 4.4). The unit's position is then updated by adding the current velocity (truncated to the maximum velocity) to the previous position (see Equation 4.5).

$$f(t2) = v(t1) + a(t1) \times (t2 - t1) \quad (4.2)$$

$$a(t2) = f(t1)/m \quad (4.3)$$

$$v(t2) = v(t1) + a(t1) \times (t2 - t1) \quad (4.4)$$

$$p(t2) = p(t1) + v(t1) \times (t2 - t1) \quad (4.5)$$

The locomotion process described above heavily depends on steering forces, which define the direction in which each unit is moved. The following sections describe the various steering behaviors that all units follow. While OxyBlood's units only use the arrival and leader following behaviors, there are other behaviors that are combined to achieve these more complex behaviors. All of these behaviors are based on Reynolds' work[119] and are described below.

Arrival Behavior

The arrival behavior, as the name implies, aims to simulate the arrival of units at a determined position or target. This is achieved by steering the units towards the target's position, while they are far from the target. As each unit approaches its intended target's position and enters a slowing distance radius, its velocity is reduced to the point of stopping when the position coincides (see Figure 4.9).

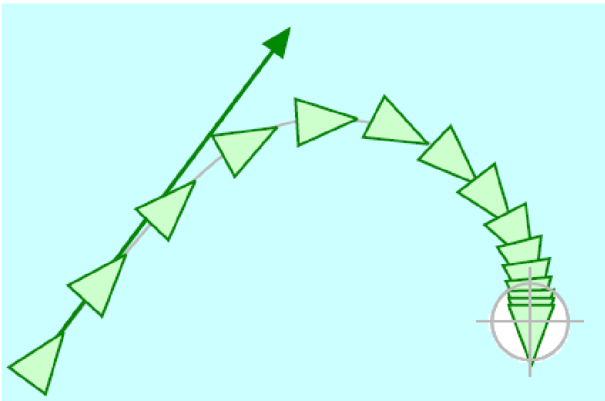

Figure 4.9: Arrival Behavior (Extracted from [119]).

This desired velocity is a three-dimensional vector that points towards the target and its magnitude depends on the slowing distance value. This value determines a radius, within which the units either maintain their velocity magnitude, or decrease it in order to stop at the target. If the unit is outside the slowing distance radius, its desired velocity is just truncated to the maximum velocity value. On the contrary, if the unit is inside the slowing distance radius, its desired velocity is linearly decreased until reaching zero.

Obstacle Avoidance Behavior

In order to avoid collisions between units and also other objects, the implementation of an obstacle avoidance behavior was also required. This behavior is equally important in simulating autonomous characters, since it prevents units from colliding and also gives them the ability to manage crowded environments by avoiding obstacles. For an effective and fast obstacle avoidance behavior, objects are considered to have a simple bounding sphere around them to simplify the process.

However, the collision avoidance is not done solely regarding the objects' bounding spheres. For each active unit, a virtual cylinder is projected in front of the unit's current direction (see Figure 4.10).

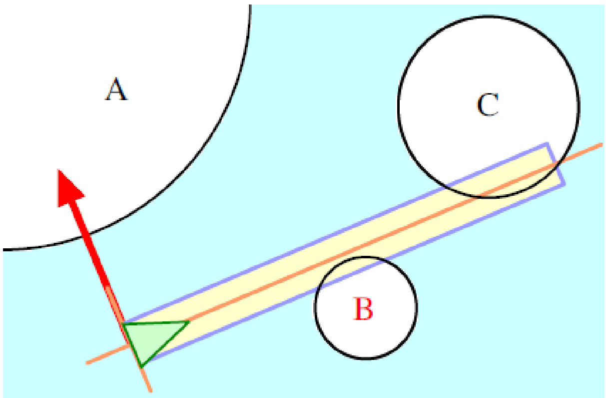

Figure 4.10: Obstacle Avoidance Behavior (Extracted from [119]).

The unit tries to keep a distance equal to its cylinder's volume (v) between itself and other objects. The volume (v) of a cylinder is calculated through its radius (r) and height (h) (see Equation 4.6).

$$v = \pi \times r^2 \times h \tag{4.6}$$

The cylinder's radius (r) is equal to the unit's bounding sphere's radius and the cylinder's height (h) is related to the unit's current velocity, so that the greater the velocity, the greater the height and consequently the greater the total volume (v). The obstacle avoidance is then achieved by detecting collision between the cylinder and the obstacles, so that the higher the velocity is, the farther away the collision detection process reaches. This behavior prioritizes obstacles in terms of collision risk. For instance, if a unit's cylinder doesn't intersect any obstacle's bounding sphere, there is no risk of collision. On the contrary, if a unit's cylinder intersects a group of obstacles' bounding spheres, it is necessary to calculate the distance between this unit and each of the obstacles. The closest obstacle to the unit is considered the most threatening and so the unit steers to avoid it, by applying a force in a deflective direction, such as a tangent direction to the obstacle.

As shown in Figure 4.10, unlike obstacle A, both obstacles B and C intersect the cylinder. However, obstacle B represents the greatest collision risk, therefore the obstacle avoidance method returns the steering force required to avoid this obstacle. If there was no risk of collision, this method would return a zero vector, so that no steering force would be applied to the unit, because there was no obstacle to avoid.

Separation Behavior

The separation behavior concerns a group of units and how they interact within the group itself. As the name implies, it relates to how a group of units manage to stay separated from each other, even if they have the same goal (e.g. following a leader unit). This behavior is responsible for steering a group of selected units for separation and does so by applying a repulsive force. This force is calculated by subtracting the positions of the current unit and its surrounding group members, normalizing and adding a position offset to maintain a specified distance from other units. Figure 4.11 shows a representation of this behavior.

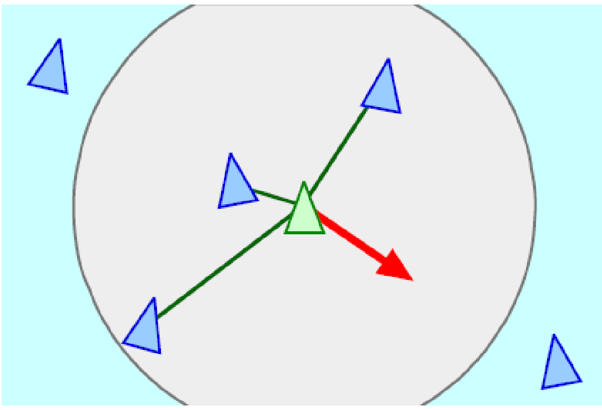

Figure 4.11: Separation Behavior (Extracted from [119]).

Leader Following Behavior

The leader following behavior is used to steer a group of units towards a designated leader unit. In OxyBlood, the leader unit is a player-controlled spaceship that can select units and integrate them into groups, in order to give these groups specific tasks or lead them to complete certain goals. This behavior is characterized by several aims:

- Following units must stay close to the leader unit, without crowding it.
- Following units must avoid getting in the way of the leader.
- Following units must stay organized.

The first two goals of the leader following behavior are achieved by relying heavily on two behaviors: the arrival and separation behaviors. Units want to arrive at a position slightly behind the leader and do so through the arrival behavior, described previously. Additionally, in order to avoid crowding, the units also follow the separation behavior, also described previously. The conjunction of these two behaviors results in a combination of cohesion forces with repulsive forces, which consequently results in the group maintaining an apparent regular formation. The grouped units are attracted towards the leader, but are also repulsed by each other, which provides an effective simulation of a group of units following a leader unit. Furthermore, to avoid getting in the way of the leader, if following units are positioned directly in front of the leader unit, they steer away from the leader unit's current path and then resume the previous behavior. Figure 4.12 represents the leader following behavior.

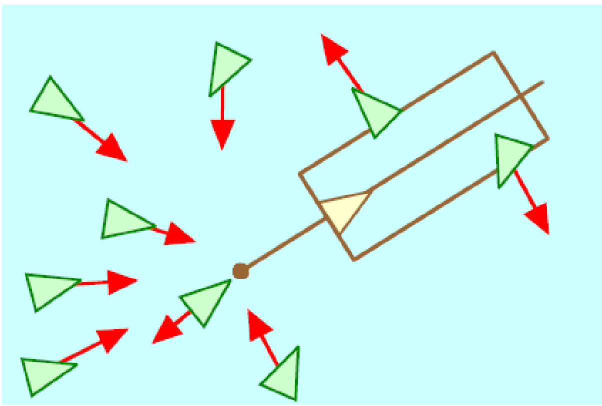

Figure 4.12: Leader Following Behavior (Extracted from [119]).

General Behavior of a Unit

Each unit's behavior is not limited to a unique behavior, but is actually a combination of various behaviors that together contribute to a more complex simulation of autonomous characters. If the player selects a group of units, this group automatically adopts the leader follower behavior and follows the player's space ship, wherever it goes. During this behavior the units also adopt the obstacle avoidance behavior, in order to avoid obstacles, such as other units or game environment objects, while following the player. If any of these units get close enough to a target, they adopt the arrival behavior, to arrive at the destination and perform the respective target action. These actions range from collection oxygen (for red blood cells), engaging enemies (for white blood cells) or healing friendly units (for platelets). All of these actions and other game play possibilities are approached on the next section.

4.1.2.5 Game Play

OxyBlood is a real time strategy game and includes much of the genre's typical features, such as deploying several types of units and arranging them in a strategic manner, depending on the goals. The player controls a small spaceship that symbolizes the coordination of the circulatory system and can create and deploy the following units: red blood cells, white blood cells and platelets; and needs to be aware of enemy bacteria flowing in the system. To deploy each of these unit types, the player must be near one the unit's bases.

In order to keep the control scheme simple, OxyBlood avoids the typical RTS controls that use the mouse to select the units. In this game, the player can only select a group of the same unit types at a time. The game uses the following keyboard layout configuration (Figure 4.13):

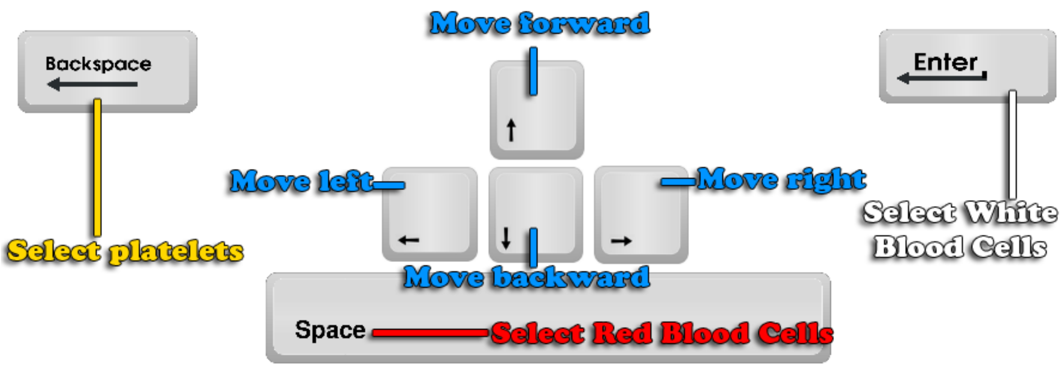

Figure 4.13: Keyboard layout configuration.

By pressing a specific unit type key near the respective unit type base, a unit will initialize its deployment process. The player can start the deployment process of five units of the same type at a time and the deployment of each unit takes about three seconds. Figure 4.14 shows an example of the deployment of red blood cell units, white blood cell units and platelet units.

Figure 4.14: Deploying units.

After deploying a group of units, the player can select units that are within the player's selection radius. This radius is represented with the respective unit type color, so that the player knows what type of units are going to be selected. The selected units automatically start following the player's spaceship, through the leader following behavior described earlier (see Figure 4.15 for an example).

Figure 4.15: Ordering red blood cells to follow the leader.

After selecting a group of units, the player may lead them to their primary goals.

When a unit reaches a target objective, it automatically changes its behavior to the respective task. For instance, if the player leads red blood cells near oxygen harvest spots, these units instantly approach these spots and start harvesting oxygen. This is achieved through the arrival steering behavior described on the artificial intelligence section. As indicated above and following the RTS rulebook, there are harvest spots where the player must bring red blood cells, so they can harvest oxygen. Subsequently, the player must lead these units to deliver the gathered oxygen to the body tissues. This process is represented on Figure 4.16.

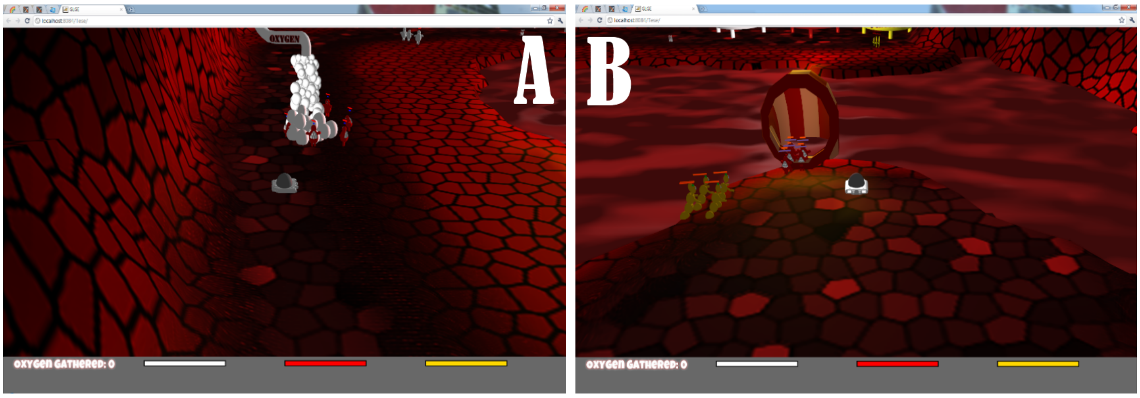

Figure 4.16: A - Red blood cells harvesting oxygen. B - Leading red blood cell to circulation.

Additionally, the player needs to beware of enemy bacteria units. These units attack primarily red blood cell units and the player must use the white blood cells to defend against these attacking bacteria. If any cells (white or red) get injured, the player can use platelets to heal them. These units follow the same behavior as the red blood cell units, but with different tasks as goals. For instance, if the player has selected a group of white blood cell units and these units get close enough to enemy bacteria units, they will automatically attack the bacteria units, because their goal is to defend the system from these units. The same can be said about the platelet units, which target injured units and automatically start healing them, if they are within their action radius. Each unit has a health bar, which indicates their current health. When their health bar is depleted, the unit falls and the player can deploy a new unit from the respective deployment base. Health is lost in confrontation with bacteria units, which are computer-controlled enemy units. These units appear from time to time, to try to disrupt or hamper the work of player's units. They target primarily red blood cells, but if confronted by white blood cells, they become a target as well. The following section describes the combat system in more detail.

4.1.2.6 Combat System

The combat system is based on each unit's life, attack rate, strength and defense capabilities. Red blood cells and platelets are weaker units, with lower combat capabilities, while white blood cells and bacteria are strong combat units. Table 4.2 shows each unit type's characteristics.

Unit Type	Life	Strength	Defense	Attack Rate
Red Cell	5	3	3	1
White Cell	10	5	3.5	2
Platelet	5	3.25	3	1
Bacteria	10	5	3	2

Table 4.2: Unit's combat capabilities.

The attack rate determines the rate at which units can perform attacks, so the white bloods cells and bacteria are twice as fast to attack, when compared to the other two units. The damage done with each attack is obtained by calculating the difference between the strength of the attacker and the defense of the attacked unit. Also, in terms of combat, if an attacking group attacks a group of the same size, every attacking unit is assigned to a specific unit from the defending group. However, if any group is outnumbered, one of the members of that group will be surrounded and attacked by multiple units (see Figure 4.17).

Figure 4.17: Red blood cells surrounded by bacteria units.

Units also have primary and secondary targets. For instance, all bacteria units have red blood cell units as their primary target. However, if during an attack by these bacteria units on red cells, white blood cells come to the rescue of red blood cells and start attacking bacteria, the bacteria units automatically change their primary targets to these white cells and begin counter-attacking. This mechanism is important because otherwise, if bacteria units were locked on attacking all red cells before attacking other units, the combat would be uneven since white cells are the strongest units. Target change also occurs in regards to platelet units. The player can form a powerful combination by maintaining platelets near the white cells units, in order to heal them during the combat. When this happens, bacteria also change focus and start attacking these healer units. Figure 4.18 shows an example of these combat mechanisms.

Figure 4.18: White blood cells attacking bacteria units.

4.1.2.7 Assessing Learning Improvements

It is important to understand whether or not players are improving their knowledge on the subject matter that OxyBlood aims to teach. Therefore, two small inquiries are used to assess the game's performance in teaching. Some parts of these inquiries are based on the Likert scale type, which measures either positive or negative responses to a given statement, hence the bipolar method definition. The typical five-level Likert scale consists of the following possible answer choices:

- Strongly agree.
- Agree.
- Neither agree nor disagree.
- Disagree.
- Strongly disagree.

The first inquiry is done before playing the game and the second one at the end of the experience, in order to ascertain whether participants learn the basics of how the human body's blood circulation works. These inquiries are also important to determine other features that may help us to improve our game, such as the game's learning curve or presentation, among others.

The first inquiry is composed of 10 questions and its structure is divided in 2 distinct sections: demographic region and previous knowledge on the subject matter. On the demographic region section, the goal is to determine the age, current grade and prior experience with video games. Age and current grade are classified easily, but the experience with video games is harder to classify. In order to do so, participants are asked for an estimate on how many games they have played, how much time per week, on which platforms and also their favorite genre. With this data, it is possible to have an idea of what kind of experience the participants have, not just in terms of video games in general, but also with the type of experience OxyBlood provides (Web RTS game). Additionally, in order to determine the participants' knowledge on the subject matter (the circulatory system), the players are inquired with a set of questions regarding this matter. These questions target each of our game's actors and therefore aim to determine the participants' knowledge regarding red and white blood cells, platelets and bacteria.

On the other hand, the second inquiry is composed of another 10 questions and replaces the section regarding demographic region, by a feedback section. This second inquiry, repeats the section regarding the subject matter, in order to determine each participant's learning progress. Subsequently, the participants are asked for feedback regarding the game play. This feedback consists on gathering information on the game's difficulty, interaction and also obtaining suggestions for potential improvements.

Currently, no results can be provided regarding the learning ability of this game because the evaluation phase hasn't started yet. However, progress is being made to conduct an experiment in a local public school, which may provide a valuable initial insight on that matter.

4.2 Rocky - The Math Cat

Rocky - The Math Cat was developed with two other colleagues for Microsoft's International Imagine Cup 2011 Game Design competition and has successfully qualified for round 2 of the competition. This means that the game reached the top 50 games worldwide. Unfortunately, the game did not go through to round 3, which only allowed 5 games to go through to the finals. This game has also been submitted for the Creative Showcase at ACE 2011 (awaiting approval) and will also be submitted for the Game Competition of the same international conference.

4.2.1 Designing Rocky - The Math Cat

The goal of this game is to help children learn basic math operations, such as sums and subtractions, by stimulating them to enjoy learning these contents. A different approach was taken when designing this game, when compared to the OxyBlood game introduced previously. That game aimed to adapt real life processes, like delivering oxygen throughout our bodies, to game play, so that the player could learn by doing. In the case of Rocky - The Math Cat, the approach was to create a traditional video game with a reward system that was based on the player's performance in math. For instance, if the player defeats an enemy, he is presented with a math quiz. If he answers correctly, his score is increased, otherwise the score is decreased.

4.2.1.1 Choosing a Game Genre

For this game, choosing a game genre was not tied to the subject that it aimed to teach. Therefore, the goal was to choose a game genre that would provide engaging game play and also be adequate to young students. The genre that looked more fitting for those purposes was the platform genre, represented by iconic games like Super Mario. Accordingly, a decision was made to develop a 3D third person platform game, where the player could run and jump to pick up objects, while simultaneously avoiding enemies. This game genre is also known for its boss battles, which was also considered as a goal for implementation.

The platform game genre allows an easy integration of a reward system. The idea was to, whenever the player managed to defeat an enemy, a pop-up window would appear with a math operation for the player to solve. If the player answered correctly his score would increase, otherwise it would decrease. Additionally, each time the player managed to pick up eight items a question would be asked, which could result in further increasing or decreasing his score, depending on the answer given. Also, at the end of each level, the player would be required to defeat an operation-specific boss that would resemble a final exam. During the boss fights the player would also need to answer correctly several math questions, in order to progress.

4.2.1.2 Other Design Choices

In terms of graphical style, a similar approach to the one applied for the OxyBlood game was used, with stylized graphics that resemble cartoon characters and environments. This choice was based on the fact that the target audience is similar, but potentially even younger. Moreover, the main character of the game was created based on a cat that behaves like a human (named Rocky), since this character seemed to be a likeable character for young players. Subsequently, the game was named Rocky - The Math Cat, since the goal of the game was to teach math skills and the main character was a cat called Rocky. Additionally, Rocky's enemies were designed to resemble math operation symbols. For instance, if Rocky managed to defeat an enemy that resembled the sum operation symbol (+), the player would need to answer a sum quiz.

A decision was also made to make the game customizable, so that the players could learn only specific operations or all of them and also adjust the difficulty of the questions. These customizable features may be a good asset for teachers to use, since they can customize the learning content they want to teach, according to the students' needs. Also, a score system is a way to motivate the player to improve his performance in the game and simultaneously, to continue learning and improving his math skills.

4.2.2 Developing Rocky - The Math Cat

Rocky - The Math Cat was developed in XNA and uses an open source physics engine, called JigLibX[121], to manage physics interactions and collisions. JigLibX is a physics engine written in C#, using the XNA framework and is based on the C++ JigLib physics engine. It provides a great collision system and rigid body physics simulation, which makes it one of the most favored physics engines specifically for XNA.

As stated previously, this was a team project. My role in this project was lead programmer; I was in charge of programming everything game-related, except for the formulation of the math questions. I also aided in animating the characters of this game. The following sections detail the development steps in which I was involved for this project.

4.2.2.1 3D Modeling and Animation

The 3D objects were designed and modeled by one of my team partners on this project, Pedro Pereira. However, I was responsible for animating some characters since I had experience of doing it previously on Blender. Rocky - The Math Cat has a great number of different 3D objects, from big and rich levels, to the main character and his enemies, as well as level bosses. Each object was designed to resemble a cartoon character and to be easily identifiable. For instance, each of the player's enemies is designed to look like the operation it represents (see Figure 4.19).

Figure 4.19: Rocky and the sum enemy.

4.2.2.2 Computer Graphics Techniques

During the development of this game, several shader techniques were developed to improve the game's overall look, from simulating Rocky's fur, to simulating water reflections and ripples, as well as lighting. Shaders are a set of low-level software instructions that are mainly used to render customized effects on the GPU. In order to create and use shaders on XNA, it is necessary to use the High Level Shader Language (HLSL)[122]. This shading language provides a high level approach to shader programming, with support for types, expressions, statements, functions and syntax similar to the C language syntax.

The simulation of Rocky's fur, was achieved through a method called Shell Rendering[123], which simulates the creation of patches of fur by rendering several horizontal slices through each patch (see Figure 4.20).

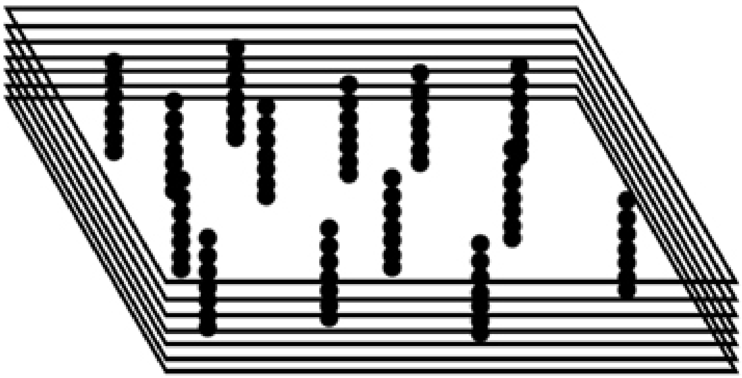

Figure 4.20: Shell Rendering - Example of a patch of fur.

As the above figure shows, points are drawn with each consecutive horizontal slice, which gives the impression of fur. These points are drawn based on an input texture, which is mapped so that transparent pixels represent areas without fur and colored pixels represent areas with fur. The points used to simulate fur that were mentioned previously, are drawn in the mapped positions corresponding to the colored pixels. The texture is wrapped around the model as many times as the amount of horizontal slices specified and is moved consecutively slightly further along the normal, building up the layers of points that ultimately simulate strands of fur. Figure 4.21 shows a comparison between rendering Rocky's model with this technique and without it.

Figure 4.21: A - Rocky without fur. B - Rocky with fur.

As seen in the above Figure, the addition of fur increased the visual quality of the main character (Rocky) and made it look more rich and full of life.

Subsequently, to add different game play mechanics on some of the game's levels, it was necessary to simulate a water component. These game play mechanics are detailed throughout section 4.2.2.5 (Game Levels). Simulating water is a rather complex task and especially in a video game, where simulating water is just one of multiple elements that need to be rendered and simulated. Therefore, the water visual effect should look convincing, without being excessively heavy to process. In order to do so, specular water reflections and ripples were implemented, but water refraction was ignored.

Water reflection is achieved through simple physics rules: the reflection angle (beta) equals the incidence angle (alpha), measured from the normal vector to the water surface (see Figure 4.22).

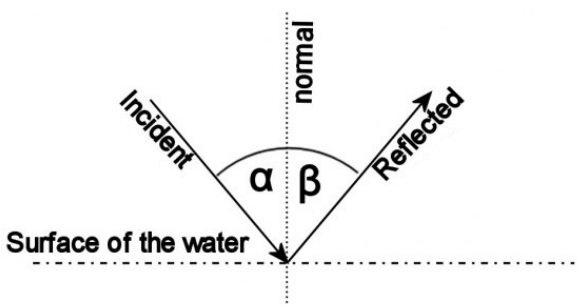

Figure 4.22: Water reflection (Image adapted from [124]).

The next step in simulating water reflection, is to determine what color is actually reflected on the surface of the water. Implementing real reflections would make the simulation more impressive, but due to processing constraints, only specular reflections were implemented instead. These kinds of reflections resemble bright spots and are more apparent on shiny surfaces (e.g. water surface). They are easily implementable because if a material has a flat surface, the light waves are parallel to the reflection. The reflection is calculated through the set of physics rules defined previously (Figure 4.22). Based on the camera's position and orientation, the direction of incidence to the water surface can be obtained. Subsequently, it is easy to obtain the angle between this direction and the normal vector to the surface, defined as the alpha angle on Figure 4.22. So, at this stage, the direction of the reflection is known and it is only necessary to apply the specular reflection parallel to this direction.

There are several methods to create specular effects, but in this case, the method used was the Phong reflection model[125]. According to this model, surfaces reflect light as a combination of ambient[1], diffuse[2] and specular[3] reflections, which form a specular highlight that is more intense, when close to the direction of reflection. Figure 2.23 shows a representation of the Phong reflection model.

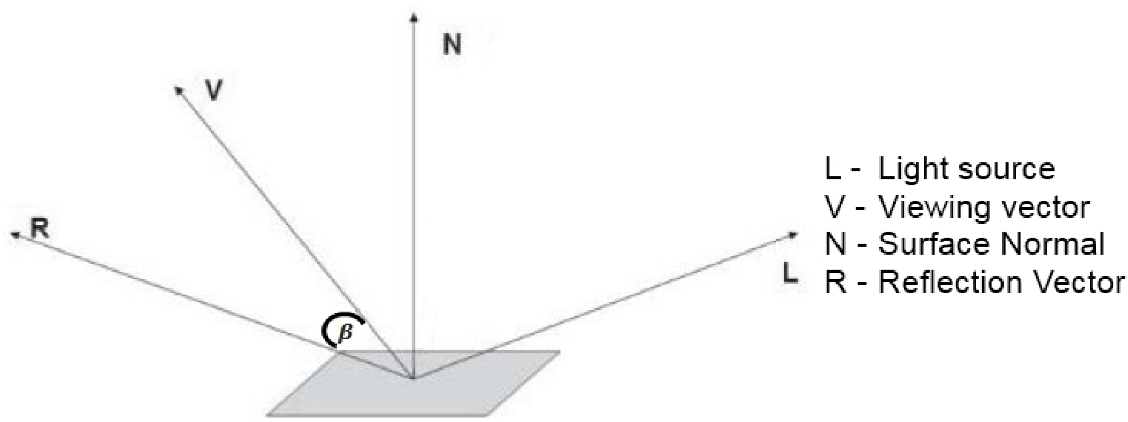

Figure 4.23: Water specular reflection: the Phong method (Image adapted from [124]).

The Phong illumination model follows the following formula (4.7):

[1]ambient: general brightness due to the light source, regardless of surface orientation and light position or direction.

[2]diffuse: light diffusely reflected off surface.

[3]specular: light specularly reflected off surface.

$$I = k_{ambient} \times I_{ambient} + (Ip/(d))[k_{diffuse} \times (N \cdot L) + k_{specular} \times (V \cdot R)n] \tag{4.7}$$

On the above formula, the different K's mean coefficient and the I's regard the different intensities. In short, the intensity of a point is equal to the sum of the ambient light intensity and the sum of the diffuse and specular intensity, scaled to the distance of the light-source. In terms of the specular reflection alone, it is calculated through the power of the cosine of the angle between the reflection vector(R) and the viewing vector(V), scaled by the specular coefficient (see Equation 4.8)

$$k_{specular} \times \cos^n(\beta) \tag{4.8}$$

This representation can be simplified, since the dot product of two vectors equals the cosine of the angle between them (see Equation 4.9).

$$k_{specular} \times (V \cdot N)^n \tag{4.9}$$

The exponent value (n) is used to increase or decrease the shininess of the surface.

After dealing with the reflection effects, it was time to create small ripples to simulate water movement. In order to achieve this effect at small expense, a technique called bump mapping[126] was used to simulate a wave motion effect. Bump maps are often created in order to simulate certain surfaces, without the need of adding a lot of additional geometrical complexity to the object. In the case of water simulation, they are usually utilized to simulate water movement, like waves or ripples, through the movement of the bump map texture on the water plane surface.

The creation of the bump map texture is fairly simple. A sample image of the ocean that clearly contained ripples, can be used in Blender to automatically create the bump map texture. The process consists of converting each of the original image pixel's RGB components from a [0;255] value range, to a [-1;1] value range. With this process, instead of storing the RGB color values of each pixel, the image stores the normal values (x, y, z) of each pixel. For instance, if a determinate pixel in the bump map texture, had a normal vector pointing upwards (0, 1, 0), the surface would be completely flat at that point. The x and z axis values determine how flat or rippled the water will be. Thus, the larger the values for these axis are, the less the normal vector will point upwards and consequently the more rippled the water will be. The water relief is then created based on this bump map's normals values, recurring to traditional bump mapping techniques. Figure 4.24 shows an example of the resulting water effect being rendered within the game.

Figure 4.24: Water rendering.

4.2.2.3 User Interface

The user interface is divided in two different components: the game menus and the heads-up display (HUD). The game menus give the player the ability to navigate the game's options outside of game play (e.g. starting or pausing the game, among others). The HUD is used during game play to display information for the player (e.g. character health and items, among others).

The menus are simple to use and follow the traditional menu system approach. For instance, when beginning the game the player can customize the game options, by accessing the options menu. Here he can choose the interaction device, the game's difficulty and the type of math questions that he wants to be asked. The interaction device can be chosen from a list of connected devices, such as the mouse and keyboard, the XBOX 360 controller and the Wiimote controller. The game's difficulty is classified as easy, medium or hard and affects the range of number used for the math questions:

- Easy settings - numbers ranging from 0 to 10 (e.g. 2x5=?).
- Medium settings - numbers ranging from 0 to 50 (e.g. 12+46=?).
- Hard settings - numbers ranging from 0 to 100 (e.g. 99-72=?).

Additionally, the player can customize the game to use only certain math operations, in order to adjust the game to the player's difficulties. If the player has trouble solving division problems at school, he can customize the game to focus on division operations, which may help him improve his division skills. This choice also affects the game's enemies. For example, by choosing only the division operation, the player will only face division enemies. Notwithstanding, the player will still face the end of level bosses that ask questions involving all operations. Figure 4.25 shows the options menu, where the player can configure the game's options.

Figure 4.25: User Interface - Customizing the experience.

In Rocky - The Math Cat, the HUD is used to display information about the player's current health, remaining lifes and number of milk cups obtained. Occasionally, a non-playable character (Professor Felix) shows up, in order to give tips to the player. These tips are usually related to new game play situations or hints on how to defeat the end of level bosses. Figure 4.26 shows a description of the HUD.

Figure 4.26: User Interface - Heads-up display.

4.2.2.4 Game Play

The main goal of the game is to obtain the highest score possible, in the shortest amount of time. In order to increase the score, the player can catch milk cups that are spread around the levels or earn points by defeating the enemies and solving the respective mathematical questions correctly. The milk cups are spread throughout the game's levels, in various platforms that Rocky must traverse, in order to successfully progress. When the player catches a milk cup, he automatically earns one point. Additionally, if he defeats an enemy and answers the corresponding question correctly, he earns three points, otherwise he loses three points. Moreover, each time the player catches eight milk cups, a question arises to allow the player to obtain or lose three additional points, depending on the outcome of the answer. See Figure 4.27 for examples of Rocky catching milk cups and engaging an enemy.

Figure 4.27: A - Rocky catching milk cups. B - Rocky engaging an enemy

When the player is prompted with a math question, the game pauses to let the player answer. Figure 4.28 shows a player answering two math questions, one correctly and another incorrectly.

Figure 4.28: A - Rocky answering correctly. B - Rocky answering incorrectly.

Rocky starts each level with three lifes and a full health bar, which is comprised of three health points. Whenever the player engages an enemy and the enemy manages to hit the player, the player loses one health point. If the player has no health points left, he loses one life. If the player loses all of his lifes, he loses the game and has to start over from the beginning. The player can recover health points by catching health items (hearts), which are scattered around the levels. Each regular enemy has one health point and in order to defeat it, the player has to either jump on top of it or use the spin attack to hit it.

The player has to progress through several platform zones, where he needs to jump from stationary or moving platforms, while simultaneously catching as much milk cups as possible, in order to obtain good scores. At the end of each level, the player finds a special challenge, an enemy boss that he will have to defeat, in order to proceed to the next level. Each boss confrontation is different and requires the player to solve different problems, since these enemies can't be hit directly by the player. The end of level bosses also work as final exams, where besides solving the boss-specific problems, the player needs to answer several math questions, ranging all operations. Only by answering correctly to the math questions, will the boss lose health.

After defeating each boss, the player is presented with a level score screen that shows the number of points he obtained on that level and allows him to go through to the next level. The player's score is accumulated throughout the levels and when he finishes the game, a final score screen is shown. If the player beats any of his previous scores, this final score screen allows the player to save his new high score, along with his name, into the high scores panel (see Figure 4.29).

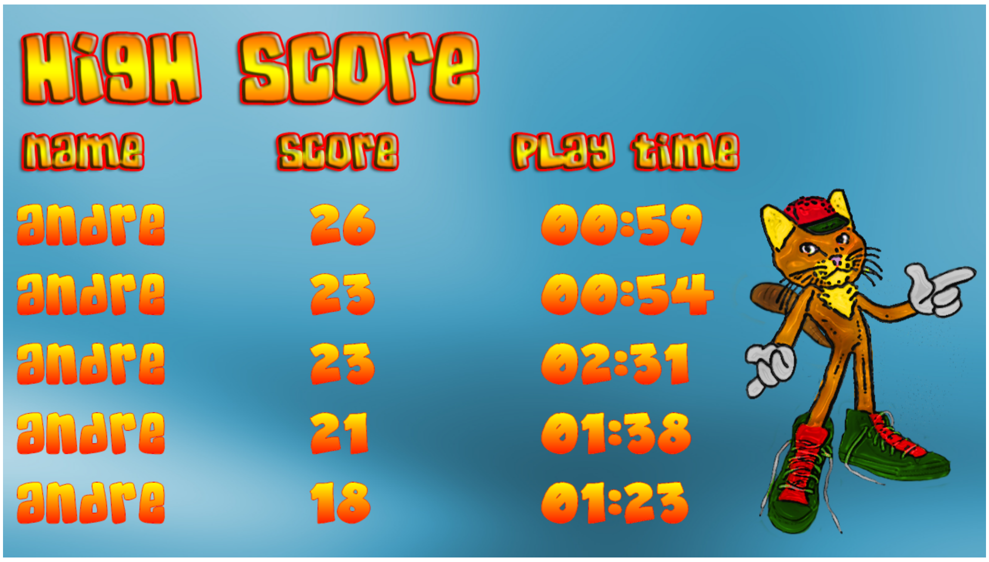

Figure 4.29: Highscores panel.

As stated previously, the game allows the player to control Rocky with three different peripherals: mouse and keyboard, XBOX 360 controller and the Wiimote. Each controller uses the following configuration layouts:

1. Mouse and Keyboard
   - W key – move forward
   - A key – move left.
   - S key – move backward.
   - D key – move right.
   - Space key – jump.
   - Left mouse button – attack.
   - Move the mouse – rotate the camera.
2. XBOX 360 controller
   - Left analog stick – move Rocky
   - Right analog stick – move the camera.
   - A button – jump.

- X button – attack.

3. Wiimote controller

   - Nunchuk analog stick – move Rocky.
   - Rotate the Wiimote – rotate the camera.
   - Upward movement with the Wiimote – jump.
   - Z button – attack.

4.2.2.5 Game Levels

The game is composed of five different levels, which detail Rocky's route from home to school, as well as adventures inside the school's gates. The first level is a simple tutorial level, where the player must learn the various game play mechanics and also acquaint himself to the controller configurations. In this level, the player interacts with several question marks, strategically placed near points of interest. These points of interest explain the basic functioning of enemies, milk cups, health items, water and moving platforms. After completing all of these actions, the player progresses to the second level. Figure 4.30 shows a screenshot of this first level.

Figure 4.30: Level 1 – Tutorial level.

The second level is where Rocky's adventure really begins. In this level, the player must take Rocky from home to the school's gates. The player needs to follow directions placed throughout the level, in order to reach the school. However, along the way the player can deviate from his course, in order to explore the city and find places like a football field where he can play (e.g. by kicking the balls), among others secret places. At the end of the level, the player finds the school's gates and is challenged by the first boss, the subtraction boss. After defeating this boss, the player proceeds to enter the school's gates. Figure 4.31 shows a section of this level, where a part of the city can be seen, as well as the football field at the distance.

Figure 4.31: Level 2 - From home to school.

The third level takes place inside the school's gates. Here, Rocky has to make his way from the school's gates, through the parking lot, pass by the basketball field and the playground, to find the school building. In this level, the player is introduced to moving platforms for the first time. In this case, the moving platforms are used for the player to start over, if he falls off a stationary platform. Arriving at the entrance of the school, Rocky faces the sum boss, in order to be able to enter the school. Figure 4.32 shows the basketball field as well as a platform zone.

In the fourth level, the player is finally inside the school building. After moving through some the school's classrooms, Rocky accidentally falls down to the school's sewers. At this stage, the player needs to get Rocky out of the sewers and back to the

Figure 4.32: Level 3 - At the school gates.

school building. In the sewers, the player is introduced to water. Cats can't swim, so if Rocky falls down from a platform and into the water, the player automatically loses one life and needs to start from the last checkpoint. By the end of this level, another boss emerges, the multiplication boss. After defeating this challenge, the player can proceed further into the school's sewers. Figure 4.33 shows the sewers part of this level.

Figure 4.33: Level 4 - The school's sewers.

The fifth level is the biggest and most complex level of the game. In this level, the player must lead Rocky out of the sewers and into the school's gymnasium. To achieve this, the player must raise the sewer's water level, in order to use inflatable boats that are floating on the water. Only by raising the water level and jumping onto these floating boats, can the player succeed in reaching the end of the level. The water is raised after the player jumps on giant buttons that can be accessed by using the moving platforms. However, the player needs to remain cautious because the water can harm Rocky. After arriving at the gym, the player needs to face the final boss battle against the division boss. If the player manages to defeat this final challenge, Rocky can finally reach his classroom and successfully attend classes, subsequently concluding the game. Figure 4.34 represents the final level of the game.

Figure 4.34: Level 5 - Getting to the gym.

The following sections detail the game's enemies and final boss battles, as well as the algorithms used for their behaviors.

4.2.2.6 Enemies

The game's enemies resemble the math operations that the players chose, while customizing the game. These enemies aim to make the player's progress throughout the game more challenging, but are also essential to the process of learning since they are responsible for generating math questions. Figure 4.36 shows the game's enemies.

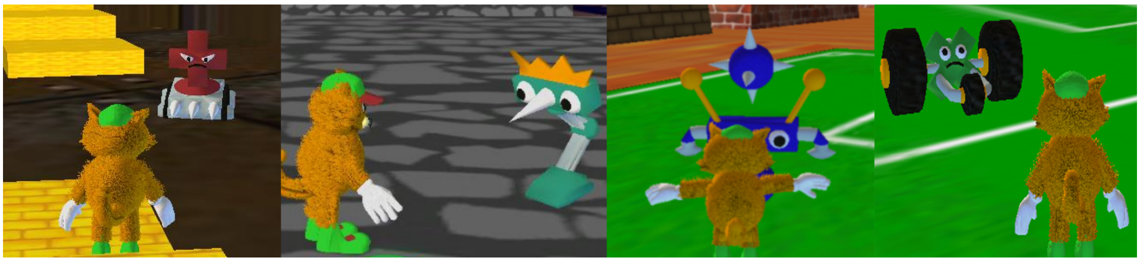

Figure 4.35: From left to right: sum, subtraction, division and multiplication enemies.

As stated previously, enemies are computer-controlled autonomous characters that try to create bigger challenges for the player to overcome. Rocky's enemies follow two sets of behaviors: looking at and engaging the player.

Look at

The look at behavior has the goal of making enemies look at the player, when the player is within a certain range. When enemies enter this state, they are observing the player's actions and are ready to attack. Figure 4.36 shows an example of this behavior.

Figure 4.36: Look at behavior.

This is achieved through the following operations:

```
(1) difference = enemyPosition - playerPosition
(2) right = verticalOrientation * difference
(3) normalize(right)
(4) backwards = right * verticalOrientation
(5) normalize(backwards)
(6) up = backwards * right
(7) normalize(up)
```

In **(1)**, we calculate the difference between the enemy's position and the player's position. The resulting three-dimensional vector contains the direction that the enemy will be facing. Figure 4.37 shows a visual example of an operation of this kind.

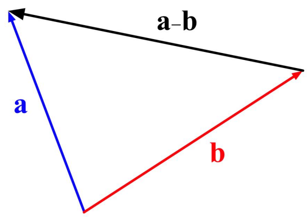

Figure 4.37: Vector difference.

However, this vector is not enough to rotate the enemy to face the player. The orientation matrix must also be built, based on this direction vector. The orientation matrix is composed of a rotation sub-matrix and a translation vector. The rotation sub-matrix has three distinct 3D vectors: the *right*, *up* and *backward* vectors; while the translation vector contains the position value for the object. Figure 4.38 presents the structure of the orientation matrix.

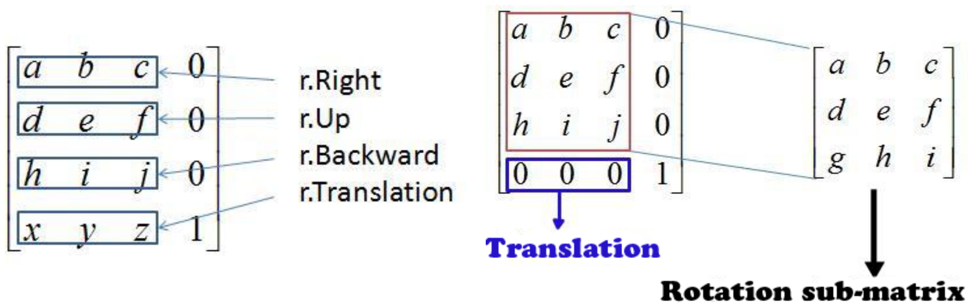

Figure 4.38: Orientation Matrix Structure.

So, as seen in the previous Figure, the rotation sub-matrix is comprised of *right*, *up* and *backward* vectors. Figure 4.39 shows how these vectors are related to each axis.

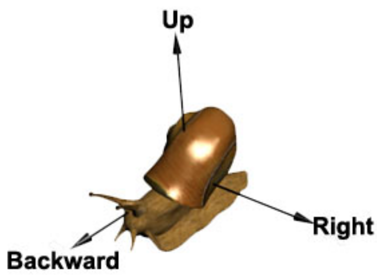

Figure 4.39: Rotation sub-matrix.

An object's direction and vertical orientation determine its orientation in a three-dimensional space. The object's direction is obtained after **(1)** and the vertical orientation is a known value for each object, thereby the rotation sub-matrix can be easily created. The creation of this matrix is described in steps **(2)** to **(6)**.

In **(2)**, the *right* vector is obtained by calculating the product of the direction vector and the vertical orientation vector. The resulting vector is orthogonal to both of the original vectors (see Figure 4.40 for an example of this process).

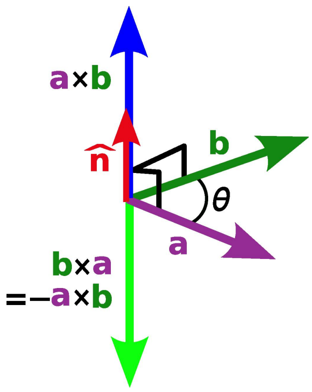

Figure 4.40: Vector product.

In **(3)**, the *right* vector is normalized, in order to maintain the rotation sub-matrix orthogonal and normalized. The normalization process consists of transforming the raw vector into a unit vector that points in the same direction. This process involves calculating the vector's length and then dividing each of its components (x, y and z) by the length obtained previously. Figure 4.41 shows the result of normalizing a vector.

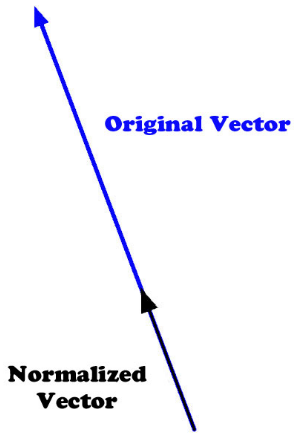

Figure 4.41: Vector Normalization.

The previous processes are similarly applied for the remaining rotation sub-matrix components. In **(4)**, the *backward* vector is a result of the product between the *right* vector, obtained in **(3)**, and the original vertical orientation vector. This results on a vector pointing towards the Z axis. Furthermore, in **(5)** and following the same creation process as the *right* vector, it is necessary to normalize this *backward* vector, in order to keep the orthogonal and normalized characteristic of the orientation matrix.

Accordingly, in **(6)** the last remaining element of the rotation matrix is calculated: the *up* vector. This vector is obtained by calculating the product between the *right* vector (X axis) and the *backward* vector (Z axis), which results in a vector pointing to the Y axis. As in the above steps, the resulting *up* vector is normalized to maintain the matrix's orthogonal and normalized characteristics, as described in step **(7)**.

Subsequently, the obtained *right*, *backward* and *up* vectors are used to create the rotation sub-matrix, as can be seen on table 4.3. Since the orientation matrix is only modified in regards of its rotation sub-matrix, when following the look at behavior, the enemies only rotate towards the position where the player is standing, but don't move.

$$\begin{bmatrix} Right.X & Right.Y & Right.Z & 0 \\ Up.X & Up.Y & Up.Z & 0 \\ Backwards.X & Backwards.Y & Backwards.Z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Table 4.3: Resulting Orientation Matrix

Attack

After an enemy enters the observing state describe above, if the player gets closer and enters the enemy's attack radius, the enemy will attack. In order to attack the player, the enemy must first chase the player and then, when the player is within reach, attack him. This behavior is done simply by using the look at behavior described previously, where the enemy is rotated towards the player and then by applying a forward force that ultimately drives the enemy onto the player's position. When the enemy comes close enough to the player, it tries to hit him. Enemies have an attack rate that indicates how often they attack the player. Figure 4.42 shows an example of an enemy attacking the player.

Figure 4.42: Sum enemy attacking the player.

4.2.2.7 Boss Battles

Bosses are more complex enemies that require the player to approach them differently, when compared to common enemies. Each boss has a unique way to be defeated, which requires the player to answer several different math questions, thus these opponents serve as a final exam for each level. The HUD suffers a small modification during boss battles: the bosses' health bar is displayed on top of the screen, so that the player is aware of the remaining health of these challenging opponents. This section presents details on how each of the game's boss battles unravels.

The first boss battle occurs near the end of the second level, at the entrance to the school's gates and involves the subtraction boss. Figure 4.43 shows a scheme that summarizes this battle.

Figure 4.43: Rocky fighting the subtraction boss.

This boss remains stationary but constantly throws big spiked balls towards the player. These balls explode upon contact with the player, so the player must be aware of them at all times. This boss uses the look at behavior, to aim at the player and throw the spiked balls (see part 1 of Figure 4.43). In the surrounding area, there are four cannons pointing at the boss, which the player must use to defeat him (see Figure 4.43 – part 2). Each of these cannons has an item that the player must activate, in order to shoot a bomb against the boss (Figure 4.43 – see part 3). These items correspond to every operation available and once the player activates one of them, he is prompted with a math question about the corresponding operation. If the player answers the questions correctly, the cannon shoots a bomb that explodes upon touching the boss, reducing his health (see Figure 4.43 – part 4). After activating one of the items, the activated item is disabled, in order to to force the player to answer a math question involving another operator. Once the player answers one question regarding each of the different possible operations, all the items are re-enabled.

The second boss battle involves the sum boss and occurs at the top of the school building, by the end of the third level. This boss battle is summarized in sequence on Figure 4.44.

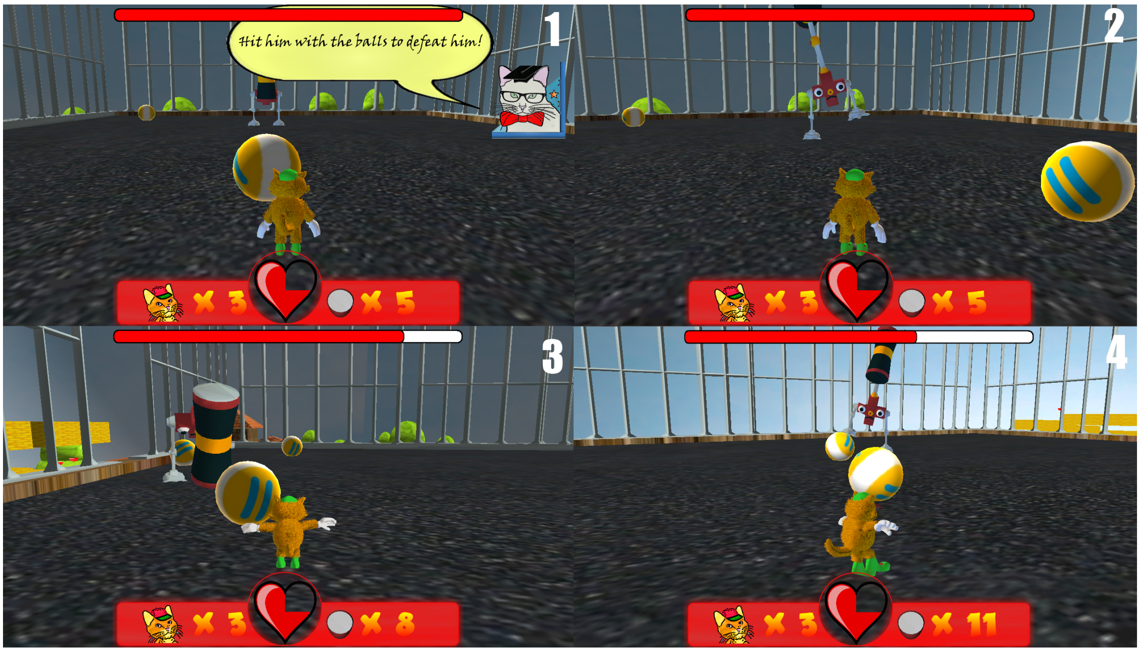

Figure 4.44: Rocky fighting the sum boss.

Unlike the subtraction boss, this boss is very mobile and quickly chases the player around the scenario (see parts 1 and 2 of Figure 4.44). This boss follows the look at and attack behaviors, but has a much greater chase velocity and attack radius. Due to the fact that this boss possesses a giant hammer as weapon, he can hit Rocky from a considerable distance (see Figure 4.44 - part 3). The only way for the player to attack this opponent, is to kick one of the giant balls placed around this area (see Figure 4.44 - part 4). Therefore, the player must run way from the boss and take advantage of failed attack attempts, to try and kick balls towards the boss. If the player manages to hit the boss, he needs to answers a math question regarding a random math operation. Only by answering correctly to these questions, will the boss lose health and be defeated.

The third boss battle occurs in the sewers, at the end of the fourth level. In this case, the opponent is a flying dragon that resembles the multiplication operator (see part 1 of Figure 4.45). This boss aims at the player and after a brief concentration moment, thrusts forward to ram the player (see Figure 4.45 - part 2). If the boss rams Rocky, the player loses one heath point. Additionally, the player must be careful not to be rammed outside of the boss arena, because upon reaching this area the water level automatically rises to the arena's height level. In order to defeat this opponent, the player must avoid being hit and jump onto the stationary inflatable boats (see Figure 4.45 - part 3). These boats are marked with an operator symbol, so that when the player jumps onto them, a question is asked regarding the respective operation. By jumping onto these boats and answering the prompted question correctly, the sewer's water level rises and consequently soaks the enemy dragon. Simultaneously, the player can avoid the water by staying in the inflatable boat, for as long as the water level remains high (see Figure 4.45 - part 4). After using one of the boats, the boat is temporarily disabled, to force the player to use another boat and answer a math question involving another operator. Once the player answers four questions regarding all of the different possible operations, all boats are re-enabled.

Figure 4.45: Rocky fighting the multiplication boss.

The fourth and final boss battle, takes place after Rocky leaves the sewers and reaches the school's gymnasium, the game's final destination. During this final battle, the player faces a giant division boss that tries to squash Rocky with his giant dish-like hands (see part 1 of Figure 4.46). In order to defeat this challenging enemy, the player must use the gym's equipment to his advantage. For instance, the player may use the trampolines spread around the gym, in order to jump much higher and activate question items that are stationed high above the player (see Figure 4.46 - part 2). After activating one of these question items, the player is asked a math question of the corresponding operator and if he answers correctly, a huge anvil is dropped on top of the division boss (see Figure 4.46 - parts 3 and 4). This question item system follows the same behavior as explained for other boss battles. After using one of the question items, it is temporarily disabled, in order to force the player to use a different item and consequently answer a distinct math operation question. Then, if the player answers one question regarding each of the different possible operations correctly, all question items are re-enabled.

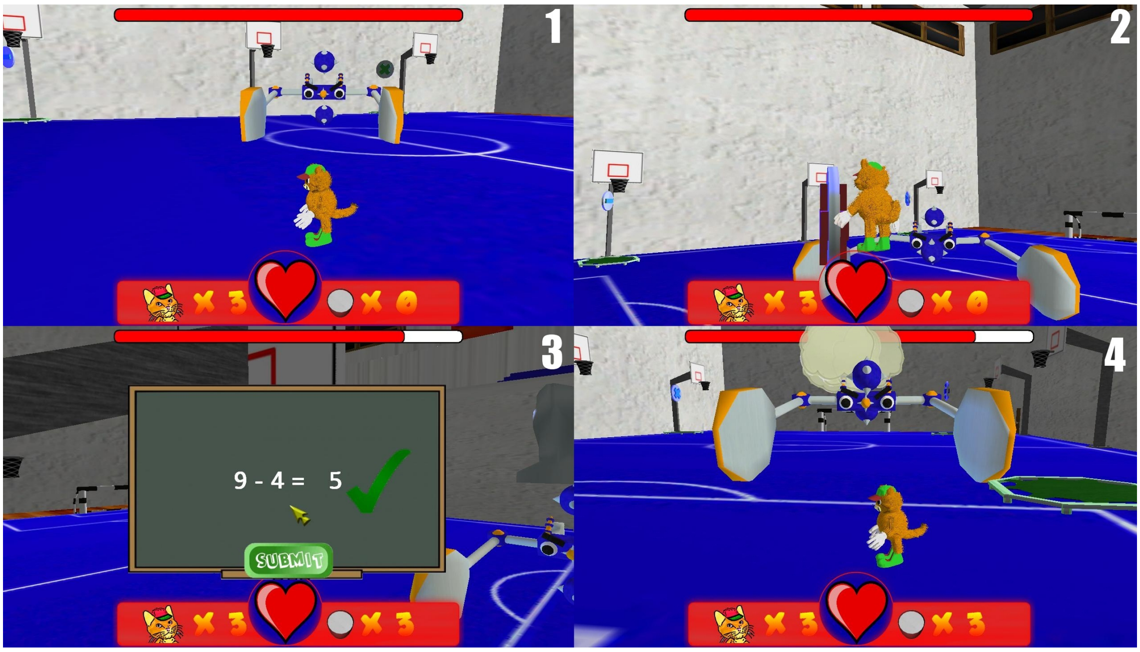

Figure 4.46: Rocky fighting the division boss.

Chapter 5

Conclusions

Serious Games are evidentially earning the interest from both research and professional deployment and application. Video games are being regarded as a very relevant medium for the future and that includes education, training, and marketing, among others. These compelling virtual worlds joined together with engaging gameplay mechanics, along with other entertaining elements, can provide effortless and interested learning and training experiences, which is very hard to obtain through traditional education or training methods. Research is still somewhat contradictory, but in general the benefits and possible applications of Serious Games are overwhelming. Such benefits include improvement of learning skills, motivation, teamwork capabilities, strategy thinking and physical fitness, among others. However, there are also some negative impacts, especially related with violence in video games.

Regardless of the benefits already shown to be resultant of using Serious Games, when compared to traditional video games, Serious Games still lack some of the fun and ultimately engaging elements around which they should revolve, in order to promote the flow state and improve their effectiveness. One of the greatest efforts of the developed work for this dissertation was to address this issue. Both OxyBlood and Rocky - the Math Cat are entertaining video games at their core, but simultaneously try to teach the players educational contents. Still, they have different approaches on how they employ these educational contents. OxyBlood follows the learn-by-doing method, where the player learns by doing tasks that can be associated to real life activities (e.g. by using red blood cell units to catch oxygen and transport it throughout the levels, the player can understand that these cells are responsible for delivering oxygen to our body tissues). On the other hand, Rocky - The Math Cat follows a different approach. It is a traditional 3D platform game that uses a reward system, as a positive reinforcement for correct math answers. In this case, the player can have fun within game without answering correctly to the math questions, but will only succeed and progress in the game, if questions are answered correctly. These are two different approaches, for different games, but that ultimately use video games' engaging and entertaining characteristics and apply them to education, instead of aiming only for entertainment.

Additionally, the effectiveness and applicability of Serious Games is yet to be deeply verified, with just a few studies addressing these issues in depth. This is one aspect on which we hope to contribute in the future, once we have enough data to analyze regarding the evaluation process of OxyBlood.

Currently there a several high performance technologies and tools for video game development, which enable high quality video game design. Web technologies such as WebGL, aim to bring these experiences to the next level, by allowing them to be accessed painlessly, within a web browser. These technologies are also helpful for researchers, since it is easier to retrieve data from a web game than from a typical desktop game or application. However, these web solutions often suffer from performance short-comings, imposing processing and content size limitations, hence restricting the development possibilities. Regardless of these issues, the development of OxyBlood clearly shows not only the great potential that WebGL already has for the development of Serious Games, but also perhaps an even greater potential for the massification of 3D content over the web. On the other hand, PC technologies such as XNA, provide the tools for game creators to develop compelling games without compromising excessively, on both content and processing load. These games have the disadvantage of being more difficult to distribute and also to retrieve information from. With XNA, it was possible to develop a more compelling game, in terms of visual quality and also content size. Rocky – The Math Cat features large and complex 3D models, as well as fairly complex graphical effects, which would be more difficult to achieve with web technologies, such as WebGL.

5.1 Future Work

In the future, we want evaluate both games in order to improve them with the help of this evaluation. This evaluation process also aims to ascertain whether players are actually learning the contents and also measure the effectiveness of learning through Serious Games against learning through traditional methods. We also aim to determine the advantages and disadvantages of playing video games through the web browser, against playing them through traditional PC or consoles, in the opinion of the participants in the evaluations.

Also, a research idea for the future is to develop Serious Games from different game genres that try to teach the same educational content. Then, evaluate each of the games, in order to determine which game genre adapts better to the respective educational content. It would be interesting to see the results of this type of research, for instance when comparing teaching mathematical content, against scientific content.

Another objective is to expand our Serious Games research to other platforms, such as tablets and smartphones, for instance. This way, it would be possible to explore new possibilities, mainly in terms of interaction (e.g. touch-based interaction or motion-based interaction).

In short, our goal is to evaluate Serious Games in actual classrooms, as an alternative method to teaching supported by new technologies.

References

[1] Mizuko Ito, Becky Herr-Stephenson, Dan Perkel, and Christo Sims. Hanging out, messing around, geeking out: Kids living and learning with new media. *Cambridge, MA: MIT Press*, pages 1–28, 2009.

[2] Bradley S. Greenberg, John Sherry, Kenneth Lachlan, Kristen Lucas, and Amanda Holmstrom. Orientations to video games among gender and age groups. *Simulation & Gaming*, 41(2):238–259, 2010.

[3] Eliane Alhadeff. Serious games, serious money: A sizeable market. *Future-Making Serious Game*, 2007.

[4] Jeanne B. Funk, Margaret Chan, Jason Brouwer, and Kathleen Curtiss. A biopsychosocial analysis of the video game-playing experience of children and adults in the united states. *Journal SIMILE: Studies In Media & Information Literacy Education*, 6(3):1–15, August 2006.

[5] Cheryl K. Olson. Children's motivations for video game play in the context of normal development. *Review of General Psychology*, 14(2):180 – 187, 2010.

[6] Cheryl K. Olson, Lawrence A. Kutner, and Dorothy E. Warner. The role of violent video game content in adolescent development. *Journal of Adolescent Research*, 23(1):55–75, 2008.

[7] Nick Yee. Motivations of play in online games. *Journal of CyberPsychology and Behavior*, 9:772–775, 2007.

[8] Leonard A. Annetta. The i's have it: A framework for serious educational game design. *Review of General Psychology*, 14(2):105 – 112, 2010.

[9] Mihaly Csikszentmihalyi. Flow. *New York: Harper & Row*, 1990.

[10] Alex Frazer, David Argles, and Gary Wills. *The Same, But Different: The Educational Affordances of Different Gaming Genres*, pages 891–893. 2008.

[11] Marjorie A. Zielke, Monica J. Evans, Frank Dufour, Timothy V. Christopher, Jumanne K. Donahue, Phillip Johnson, Erin B. Jennings, Brent S. Friedman, Phonesury L. Ounekeo, and Ricardo Flores. Serious games for immersive cultural training: Creating a living world. *IEEE Comput. Graph. Appl.*, 29:49–60, March 2009.

[12] Ben Sawyer. From cells to cell processors: The integration of health and video games. *IEEE Comput. Graph. Appl.*, 28:83–85, November 2008.

[13] Amanda Chaffin and Tiffany Barnes. Lessons from a course on serious games research and prototyping. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games*, FDG '10, pages 32–39, New York, NY, USA, 2010. ACM.

[14] Wee Ling Wong, Cuihua Shen, Luciano Nocera, Eduardo Carriazo, Fei Tang, Shiyamvar Bugga, Harishkumar Narayanan, Hua Wang, and Ute Ritterfeld. Serious video game effectiveness. In *Proceedings of the international conference on Advances in computer entertainment technology*, ACE '07, pages 49–55, New York, NY, USA, 2007. ACM.

[15] David R. Michael and Sandra L. Chen. *Serious Games: Games That Educate, Train, and Inform*. Muska & Lipman/Premier-Trade, 2005.

[16] Bruce Williams. Microsoft flight simulator as a training aid. January 2007.

[17] Michael Zyda. From visual simulation to virtual reality to games. *Computer*, 38:25–32, September 2005.

[18] Alice Mitchell and Carol Savill-Smith. The use of computer and video games for learning: A review of the literature. page 93, 2004.

[19] K. Corti. Games-based learning; a serious business application. *PIXELearning Limited*, November/December 2006.

[20] John Kirriemuir and Angela McFarlane. Literature review in games and learning literature review in games and learning. *Context*, 3(2):208–213, 2004.

[21] Kristian Kiili. Digital game-based learning: Towards an experiential gaming model. *The Internet and Higher Education*, 8(1):13 – 24, 2005.

[22] Thomas W. Malone. What makes things fun to learn? heuristics for designing instructional computer games. In *Proceedings of the 3rd ACM SIGSMALL symposium and the first SIGPC symposium on Small systems*, SIGSMALL '80, pages 162–169, New York, NY, USA, 1980. ACM.

[23] P. Hodson, M. Connolly, and D Saunders. Can computer-bases learning support adult learners? *Journal of Further and Higher Education*, 25(3):325–335, 2001.

[24] Chaoming Du, Honglian Liu, and Liu Wei. On the characteristics and application of integrated e-learning. In *e-Business and Information System Security (EBISS), 2010 2nd International Conference on*, pages 1–3, may 2010.

[25] America's Army Official Website. http://www.americasarmy.com, 2010.

[26] Richard Van Eck. Digital game-based learning: ItÊ$\frac{1}{4}$s not just the digital natives who are restless. *Educause Review*, 41(2):16–30, 2006.

[27] A.P. Macvean. Task-involved versus ego-involved: Motivating children to exercise in a pervasive exergame. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, pages 405–406, march 2011.

[28] Cynthia Putnam and Lorna Chong. Software and technologies designed for people with autism: what do users want? In *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*, Assets '08, pages 3–10, New York, NY, USA, 2008. ACM.

[29] Samantha L Finkelstein, Andrea Nickel, Lane Harrison, Evan A Suma, and Tiffany Barnes. cmotion: A new game design to teach emotion recognition and programming logic to children using virtual humans. *2009 IEEE Virtual Reality Conference*, pages 249–250, 2009.

[30] Neil Peirce, Owen Conlan, and Vincent Wade. Adaptive educational games: Providing non-invasive personalised learning experiences. In *Proceedings of the 2008 Second IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning*, pages 28–35. IEEE Computer Society, 2008.

[31] W. Huitt and Hummel. Piaget's theory of cognitive development. educational psychology interactive. *Valdosta, GA: Valdosta State University*, 2003.

[32] Merrilea J Mayo. Video games: a route to large-scale stem education? *Science*, 323(5910):79–82, 2009.

[33] L. Enochsson A Kjellin L. Fellander-Tsai M. Schlickum, L. Hedman. Video games in health care: Closing the gap. review of general psychology. *World Journal of Surgery*, 33:2360–2367, 2009.

[34] P. Kato. Video games: A route to large-scale stem education? *Review of General Psychology*, 14(2):113–121, 2010.

[35] Keith Nesbitt, Ken Sutton, Joshua Wilson, and Geoff Hookham. Improving player spatial abilities for 3d challenges. In *Proceedings of the Sixth Australasian Conference on Interactive Entertainment*, IE '09, pages 6:1–6:3. ACM, 2009.

[36] R. Lisi and J.L. Wolford. Improving children's mental rotation accuracy with computer game playing. *Journal of Genetic Psychology*, 163(3):172–182, 2002.

[37] James F Knight, Simon Carley, Bryan Tregunna, Steve Jarvis, Richard Smithies, Sara De Freitas, Ian Dunwell, and Kevin Mackway-Jones. Serious gaming technology in major incident triage training: a pragmatic controlled trial. *Resuscitation*, 81(9):1175–1179, 2010.

[38] Navarro J. I., Marchena E., Alcalde C., Ruiz G., Llorens I., and M. Aguilar. Improving attention behaviour in primary and secondary school children with a computer assisted instruction procedure. *International Journal of Psychology*, 38(6):359–365, 2003.

[39] ELSPA. Unlimited learning: Computer and video games in the learning landscape. *Entertainment and Leisure Software Publishers Association*, 2006.

[40] K. Squire and C. Steinkuehler. Meet the gamers. *Library Journal*, 130(7):38–42, 2005.

[41] J.C. Hong and M.C. Liu. A study on thinking strategy between experts and novices of computer games. *Computers in Human Behavior*, 19(80):245–258, 2003.

[42] P. Backlund, H. Engstrom, , and M. Johannesson. Computer gaming and driving education. *Proceedings of the workshop Pedagogical Design of Educational Games affiliated to the 14th International Conference on Computers in Education*, 2006.

[43] Marina Umaschi Bers. Let the games begin: Civic playing on high-tech consoles. *Review of General Psychology*, 14(2):147 – 153, 2010.

[44] Jane Barnett and Mark Coulson. Virtually real: A psychological perspective on massively multiplayer online games. *Review of General Psychology*, 14(2):167 – 179, 2010.

[45] Bruno Baldaro, Giovanni Tuozzi, Maurizio Codispoti, Ornella Montebarocci, Francesco Barbagli, Elena Trombini, and Nicolino Rossi. Aggressive and non-violent videogames: Short-term psychological and cardiovascular effects on habitual players. *Stress and Health*, 20(4):203–208, 2004.

[46] Elizabeth Losh. Making things public: democracy and government-funded videogames and virtual reality simulations. In *Proceedings of the 2006 ACM SIGGRAPH symposium on Videogames*, Sandbox '06, pages 123–132, New York, NY, USA, 2006. ACM.

[47] W. Lewis Johnson. Developing intercultural competence through videogames. In *Proceeding of the 2009 international workshop on Intercultural collaboration*, IWIC '09, pages 99–100, New York, NY, USA, 2009. ACM.

[48] Learning Languages With Games: Tactical Iraqi Animation World Network. http://www.awn.com/articles/gaming/learning-languages-games-itactical-iraqii. 2006.

[49] Luca Chittaro and Roberto Ranon. Serious games for training occupants of a building in personal fire safety skills. *2009 Conference in Games and Virtual Worlds for Serious Applications*, pages 76–83, 2009.

[50] C. Ware. *Information visualization: perception for design*. The Morgan Kaufmann series in interactive technologies. Morgan Kaufman, 2004.

[51] M. Zyda. *Communications ACM*, 50(7), 2007.

[52] B. R. Maxim, N. V. Patel, N. D. Martineau, and M. Schwartz. Work in progress - learning via gaming: An immersive environment for teaching kids handwriting. *2007 37th Annual Frontiers in Education Conference Global Engineering Knowledge without Borders Opportunities without Passports Vols 1 4*, pages T1B–3–T1B–4, 2007.

[53] Karla Muñoz, Julieta Noguez, Paul Mc Kevitt, Luis Neri, Víctor Robledo-Rella, and Tom Lunney. Adding features of educational games for teaching physics. In *Proceedings of the 39th IEEE international conference on Frontiers in education conference*, FIE'09, pages 250–255. IEEE Press, 2009.

[54] Mike Potel and Keri Schreiner. Digital games target social change. *IEEE Computer Graphics Applications*, 28:12–17, 2008.

[55] E. Losh. In country with tactical iraqi: Trust, identity, and language learning in a military video game digital experience. *Proceedings of the Digital Arts and Culture Conference*, pages 69–78, 2005.

[56] F. Bellotti, R. Berta, A. De Gloria, and L. Primavera. Learning contents by videogame tricks. *Proceedings of the Learning with Games Conference*, pages 24–26, 2007.

[57] F. Bellotti, R. Berta, A. De Gloria, and L. Primavera. Enhancing the educational value of video games. *Comput. Entertain.*, 7:23:1–23:18, June 2009.

[58] Richard Sandford, Mary Ulicsak, Keri Facer, and Tim Rudd. Teaching with games: Using commercial off-the-shelf computer games in formal education. *Teaching with Games*, teachingwi:61, 2006.

[59] M. Prensky. *Digital game-based learning*. Paragon House, 2007.

[60] Lucia Pannese and Maria Carlesi. Games and learning come together to maximise effectiveness: The challenge of bridging the gap. *British Journal of Educational Technology*, 38(3):438–454, 2007.

[61] J.C. Beck and M. Wade. *Got game: how the gamer generation is reshaping business forever*. Harvard Business School Press, 2004.

[62] PIXELearning. Learningbeans. *Games-based Learning; a serious business application*, 2007.

[63] BreakAway .Ltd Virtual Training Bank. http://www.breakawaygames.com/serious-games/solutions/corporate/. 2011.

[64] Carolyn Watters, Sageev Oore, Michael Shepherd, Azza Abouzied, Anthony Cox, Melanie Kellar, Hadi Kharrazi, Fengan Liu, and Anthony Otley. Extending the use of games in health care. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences - Volume 05*, pages 88.2–, Washington, DC, USA, 2006. IEEE Computer Society.

[65] Aidan Sliney and David Murphy. Jdoc: A serious game for medical learning. In *Proceedings of the First International Conference on Advances in Computer-Human Interaction*, ACHI '08, pages 131–136, Washington, DC, USA, 2008. IEEE Computer Society.

[66] Alberto Cabas Vidani, Luca Chittaro, and Elio Carchietti. Assessing nurses' acceptance of a serious game for emergency medical services. In *Proceedings of the 2010 Second International Conference on Games and Virtual Worlds for Serious Applications*, VS-GAMES '10, pages 101–108. IEEE Computer Society, 2010.

[67] Jeff Swarz, Anita Ousley, Adriane Magro, Marie Rienzo, David Burns, A. M. Lindsey, Ben Wilburn, and Susan Bolcar. Cancerspace: A simulation-based game for improving cancer-screening rates. *IEEE Comput. Graph. Appl.*, 30:90–94, January 2010.

[68] Christian Schönauer, Thomas Pintaric, and Hannes Kaufmann. Full body interaction for serious games in motor rehabilitation. In *Proceedings of the 2nd Augmented Human International Conference*, AH '11, pages 4:1–4:8. ACM, 2011.

[69] James William Burke, Michael McNeill, Darryl Charles, Philip Morrow, Jacqui Crosbie, and Suzanne McDonough. Serious games for upper limb rehabilitation following stroke. In *Proceedings of the 2009 Conference in Games and Virtual Worlds for Serious Applications*, VS-GAMES '09, pages 103–110. IEEE Computer Society, 2009.

[70] Aimee L Betker, Ankur Desai, Cristabel Nett, Naaz Kapadia, and Tony Szturm. Game-based exercises for dynamic short-sitting balance rehabilitation of people

with chronic spinal cord and traumatic brain injuries. *Physical Therapy*, 87(10):1389–1398, October 2007.

[71] Rory A Cooper, Dan Ding, Richard Simpson, Shirley G. Fitzgerald, Donald M. Spaeth, Songfeng Guo, Alicia M Koontz, Rosemarie Cooper, Jongbae Kim, and Michael L. Boninger. Virtual reality and computer-enhanced training applied to wheeled mobility: An overview of work in pittsburgh. In *Assistive Technology*, volume 17 of *ASSETS '10*, pages 159–170, 2005.

[72] Bob Hone and Wolf Schuster. Pdwii: Using novel interfaces to promote physical rehabilitation & achieve quantifiable results. In *Serious Games Summit Sessions*, GDC 2008, 2008.

[73] D. Deponti, D. Maggiorini, and C.E. Palazzi. Droidglove: An android-based application for wrist rehabilitation. In *Ultra Modern Telecommunications Workshops, 2009. ICUMT '09. International Conference on*, pages 1 –7, oct. 2009.

[74] Gazihan Alankus, Rachel Proffitt, Caitlin Kelleher, and Jack Engsberg. Stroke therapy through motion-based games: a case study. In *Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility*, ASSETS '10, pages 219–226. ACM, 2010.

[75] Nintendo's WiiFit. http://www.nintendo.com/wiifit. 2009.

[76] Christos Gatzidis, Keith Parry, Emma Kavanagh, Amanda Wilding, and Darrell Gibson. Towards the development of an interactive 3d coach training serious game. In *Proceedings of the 2009 Conference in Games and Virtual Worlds for Serious Applications*, VS-GAMES '09, pages 186–189. IEEE Computer Society, 2009.

[77] Stefan Göbel, Sandro Hardy, Viktor Wendel, Florian Mehm, and Ralf Steinmetz. Serious games for health: personalized exergames. In *Proceedings of the international conference on Multimedia*, MM '10, pages 1663–1666. ACM, 2010.

[78] Sniffel Serious Games Institute. http://www.seriousgames.dk/node/614. 2011.

[79] J.R. Gago, T.M. Barreira, R.G. Carrascosa, and P.G. Segovia. Nutritional serious-games platform. In *eChallenges, 2010*, pages 1 –8, oct. 2010.

[80] La Verne Abe Harris and Nicoletta Adamo-Villani. Effects of culture on the pre-production design of the hiv game. In *ACM SIGGRAPH ASIA 2009 Educators Program*, SIGGRAPH ASIA '09, pages 5:1–5:9. ACM, 2009.

[81] E.D. van der Spek, H. van Oostendorp, P. Wouters, and L. Aarnoudse. Attentional cueing in serious games. In *Games and Virtual Worlds for Serious Applications (VS-GAMES), 2010 Second International Conference on*, pages 119 –125, march 2010.

[82] Clint Bowers Anya Andrews, Rachel Joyce. Using serious games for mental health education. In *Serious Game Design and Development: Technologies for Training and Learning*, 2010.

[83] XNA Microsoft's game development framework. http://create.msdn.com/en-us/. 2011.

[84] T. Miller. *Managed DirectX 9: graphics and game programming : kick start.* Kick Start. Sams, 2003.

[85] OpenGL The Industry's Foundation for High Performance Graphics. http://www.opengl.org/. 2011.

[86] R. Morsi, C. Richards, and M. Rizvi. Work in progress - binx: A 3d xna educational game for engineering education. In *Frontiers in Education Conference (FIE), 2010 IEEE*, pages S1E–1 –S1E–3, oct. 2010.

[87] UDK Unreal Development Kit. http://www.udk.com/. 2011.

[88] PhysX Nvidia's powerfull physics engine. http://developer.nvidia.com/physx. 2011.

[89] Kanav Kahol. Integrative gaming: A framework for sustainable game-based diabetes management. *Journal of Diabetes Science and Technology*, 5(2):293–300, March 2011.

[90] Unity 3D Game development tool. http://unity3d.com/. 2011.

[91] Kuang Yang, Jiang Jie, and Shen Haihui. Study on the virtual natural landscape walkthrough by using unity 3d. In *VR Innovation (ISVRI), 2011 IEEE International Symposium on*, pages 235 –238, march 2011.

[92] Microsoft Visual Studio. http://www.microsoft.com/visualstudio/en-us. 2011.

[93] Apple iPhone. http://www.apple.com/iphone/. 2011.

[94] PSP Playstation Portable. http://uk.playstation.com/psp/. 2011.

[95] DS Nintendo's Portable Console. http://www.nintendo.com/ds. 2011.

[96] Loutfouz Zaman, Daniel Natapov, and Robert J. Teather. Touchscreens vs. traditional controllers in handheld gaming. In *Proceedings of the International Academic Conference on the Future of Game Design and Technology*, Futureplay '10, pages 183–190, New York, NY, USA, 2010. ACM.

[97] Google Android Phone. http://www.android.com/. 2011.

[98] Microsoft Windows Phone 7. http://www.microsoft.com/windowsmobile/. 2011.

[99] U.S. Smartphone Market: Who's the Most Wanted? The Nielson Company. http://blog.nielsen.com/nielsenwire/?p=27418. 2011.

[100] Alex Butler, Shahram Izadi, and Steve Hodges. Sidesight: multi-"touch" interaction around small devices. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, UIST '08, pages 201–204, New York, NY, USA, 2008. ACM.

[101] Wolfgang Broll and Tanja Koop. Vrml: Today and tomorrow. *Computers & Graphics*, 20(3):427 – 434, 1996.

[102] Luca Chittaro and Roberto Ranon. Using the x3d language for adaptive manipulation of 3d web content. In *Proceedings of A-H: 3rd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 287–290. SpringerVerlag, 2004.

[103] 3d applications in the browser Google O3D. http:// code.google.com/apis/o3d/. 2011.

[104] Adobe Flash Player Plug-in. http://www.adobe.com/products/flashplayer/. 2011.

[105] Adobe Flash ActionScript. http://www.actionscript.org/. 2011.

[106] Webgl opengl es 2.0 for the web. http://www.khronos.org/webgl. 2011.

[107] Khronos Open standards for media authoring and acceleration. http://www.khronos.org. 2011.

[108] HTML5 W3C Working Draft 25 May 2011. http://www.w3.org/tr/html5/. 2011.

[109] E. Pinto, G. Amador, and A. Gomes. A graphics library for delivering 3d contents on web browsers. *6th International Conference on Digital Content, Multimedia Technology and its Applications (IDC)*, pages 109–114, August 2010.

[110] C. Leung, A. Salga, and A. Smith. Canvas 3d js library. *Proceedings of the 2008 Conference on Future Play: Research, Play, Share*, pages 274–275, 2008.

[111] GLGE Webgl for the lazy Paul Brunt. http://www.glge.org. 2011.

[112] Marco Di Benedetto, Federico Ponchio, Fabio Ganovelli, and Roberto Scopigno. Spidergl: a javascript 3d graphics library for next-generation www. In *Proceedings of the 15th International Conference on Web 3D Technology*, Web3D '10, pages 165–174. ACM, 2010.

[113] André F. S. Barbosa and Frutuoso G. M. Silva. Oxyblood - a serious game in webgl. In *Proceedings of the 1st Iberian Workshop on Serious Games and Meaningful Play @ CISTI2011*, SGaMePlay'2011, 2011.

[114] D. Keaveney and C. O'Riordan. Evolving coordination for real-time strategy games. *Computational Intelligence and AI in Games, IEEE Transactions on*, PP(99):1, 2011.

[115] Tim McLaughlin, Dennie Smith, and Irving A. Brown. A framework for evidence based visual style development for serious games. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games*, FDG '10, pages 132–138. ACM, 2010.

[116] CSS3 Cascading Style Sheet 3. http://www.w3schools.com/css/. 2011.

[117] Blender 2.5 3D Modeling Tool. http://www.blender.org/. 2011.

[118] Collada Digital Asset and FX Exchange Schema. www.collada.org/. 2011.

[119] C Reynolds. Steering behaviors for autonomous characters. In *Proceedings of Game Developers Conference 1999*, GDC 99, pages 763–782. Miller Freeman Game Group, 1999.

[120] J.C. Butcher. *Numerical methods for ordinary differential equations*. Wiley, 2008.

[121] Jiglibx Physics Engine for XNA. http://jiglibx.codeplex.com/. 2009.

[122] Michael Oneppo. Hlsl shader model 4.0. In *ACM SIGGRAPH 2007 courses*, SIGGRAPH '07, pages 112–152. ACM, 2007.

[123] Jayaram K. Udupa and Dewey Odhner. Shell rendering. *IEEE Comput. Graph. Appl.*, 13:58–67, November 1993.

[124] Torsten Moller. Local illumination. In *Introduction to Computer Graphics*, 2010.

[125] B. T. Phong. Illumination for computer generated pictures. In *Communications of ACM 18*, pages 311–317. ACM, 1975.

[126] James F. Blinn. Simulation of wrinkled surfaces. *SIGGRAPH Comput. Graph.*, 12:286–292, August 1978.