

# Multiplayer Game Development Approaches for Student Integration in Universities

**André da Cruz Rodrigues**

Dissertação apresentada à Faculdade de Engenharia da Universidade do Porto para  
obtenção do grau de Mestre em Multimédia

**Orientador:** Professor Doutor António Coelho

**Co-Orientador:** Professor Doutor Armando Sousa

Julho de 2011



© André da Cruz Rodrigues, 2011

Multiplayer Game Development Approaches for Student Integration in Universities

**André da Cruz Rodrigues**

Mestrado Integrado em Multimédia

# Abstract

For every human being, The contact with new social environments requires some degree of adaptation. This is no exception for students when arriving at a new school, especially in large Universities. Although the process of integration can be primarily influenced by the psychological profile of each individual, this work tries to evaluate how digital games played by several players simultaneously can further promote this process and which methodologies and techniques can be used, in the context of game creation.

This work proposes the adoption of virtual environments to support the development of a multiplayer game that promotes teamwork as the basis for cooperation competition with other students. One of the challenges was the creation of a game concept adjusted to this objective and applicable to a large number of students, during their initial weeks of the first year.

For the implementation, it was developed a set of collaborative game levels on a virtual campus, where activities were group oriented and conducted in a controlled environment for the students. These levels were mainly focused to obtaining specific reactions from the students, like communication attempts, synchronization, leadership actions, decision making and so on. Therefore, the core set of game mechanics heavily consisted in tasks very difficult to perform alone, like moving large objects through specific paths and performing actions at specific moments altogether. The game design of each level also tries to offer multiple strategic options, promoting also group discussion and decision making of the better tactic for the game objective at hand.

In this work, was also produced a game prototype for testing the game concept results over a sample population of students. An adequate platform for implementation was used, based on the game goals and design. During development some good practices and approaches become notorious in the technical scope. There was also evidence of limitations and problems verified during implementation, where some workarounds are tried and presented in this dissertation.

Finally, a statistical evaluation is conducted over the prototype and students performance during gameplay. Some observations and data results from inquiries pointed some conclusions and also shaped some ideas for future work.

# Resumo

O contacto com novos contextos sociais requerem, muitas das vezes, algum nível de adaptação. Isto não é excepção para estudantes que chegam a uma nova escola pela primeira vez, principalmente em grandes universidades. Apesar de o processo de integração poder ser principalmente influenciado pelo perfil psicológico de cada indivíduo, este trabalho procura avaliar como é que jogos digitais compostos por vários jogadores em simultâneo, podem facilitar este processo e que metodologias podem ser utilizadas.

Este projecto propõe a adopção de um ambiente virtual que suporte o desenvolvimento de um jogo multi-jogador que promova o trabalho de equipa como a base de uma cooperação competitiva com outros estudantes. Um dos desafios foi exactamente a criação do conceito para o jogo, ajustado aos objectivos propostos, e aplicável a um elevado número de estudantes durante as semanas iniciais do primeiro ano lectivo.

Para implementação, foi desenvolvido um conjunto de níveis colaborativos num mundo virtual, onde as actividades foram orientadas para trabalho em grupo e realizadas num ambiente controlado para os alunos. Estes níveis foram especialmente concebidos para obter reacções específicas por parte dos alunos, como por exemplo tentativas de comunicação, sincronização, acções de liderança e tomadas de decisão. Deste modo, o elemento nuclear nas mecânicas de jogo concentram-se principalmente em tarefas muito difíceis de realizar sem ajuda, como por exemplo mover pesados objectos através de um trajecto e realizar acções em simultâneo. O desenho de cada nível procura também oferecer múltiplas opções a um nível estratégico, procurando assim promover a discussão em grupo e a tomada de decisões em conjunto para o desafio apresentado durante o jogo.

Neste trabalho foi também desenvolvido um protótipo para testar os resultados do conceito concebido sobre uma amostra do público-alvo. Procurou-se utilizar uma plataforma consistente com os objectivos do projecto e mecânicas de jogo. Durante a implementação foram tornando-se óbvias algumas abordagens e boas práticas na componente técnica. No entanto, foram também verificadas algumas limitações e problemas, que foram procurados ultrapassar com estratégias devidamente apresentadas nesta dissertação.

Finalmente, foi conduzida uma avaliação dos resultados sobre a eficiência do protótipo e performance dos alunos durante a execução do jogo. Os resultados provenientes destas observações e de inquéritos facultados aos alunos no final, ajudaram a definir algumas conclusões e a moldar ideias para trabalho futuro.

### Acknowledgements

I would like to thank my teachers António Coelho and Armando Sousa for their support, patience and guidance.

André Cruz

## Contents

<b>Introduction.....</b>	<b>1</b>
Context.....	1
Motivation and Objectives .....	2
Thesis Structure .....	2
Contributions.....	3
<b>State of the Art .....</b>	<b>5</b>
<b>Related Work</b> .....	<b>6</b>
eScape.....	6
EduTeams .....	7
Revolution.....	7
<b>Technological Platforms</b> .....	<b>8</b>
Unreal Development Kit.....	8
Adobe Flash .....	9
Unity.....	9
Second Life .....	9
Summary and Conclusions.....	10
<b>Game Concept – Ideas and Requirements.....</b>	<b>11</b>
Game Design and Proposed Game Levels .....	12
Platform Selection.....	15
<b>Game Development in Second Life</b> .....	<b>15</b>
Summary and Conclusions.....	16
<b>Implementation .....</b>	<b>19</b>
Asset Creation.....	19
Level Editing and Game Assemblage .....	22
Programming.....	27
Summary and Conclusions.....	29
<b>Prototype Evaluation .....</b>	<b>31</b>
Summary and Conclusions.....	37
<b>Conclusions and Future Work.....</b>	<b>38</b>
Future Work.....	39
<b>References .....</b>	<b>41</b>





## Figures List

Figure 3.1: Map Design from Level 1 - Crate Carrier. ....	13
Figure 3.2: Map Design from Level 2 - Sliding Spheres .....	14
Figure 3.3: Map Design from Level 3 - Jumping Platforms .....	15
Figure 4.1. Texture mapping template for character cloth .....	19
Figure 4.2. Texture applied to character avatar.....	19
Figure 4.3. Animation development to apply into player characters (pushing objects).....	20
Figure 4.4. Crate sculpting for better movement during player interaction .....	21
Figure 4.5. Asset modelling in Autodesk Softimage .....	21
Figure 4.6. Avatar animation during player interaction with object .....	22

Tables List	
Table 3.1: Team Distribution by Level .....	12
Table 5.1: Likert Results from students inquire .....	33

# Terms

AAA	Triple A Title
HUD	Heads-Up Display
NPC	Non-Player Character
OOP	Object Oriented Programming
LSL	Linden Script Language
SDK	Software Development Kit
HTTP	Hypertext Transfer Protocol



# Introduction

The integration of newcomer students at the university can be an overwhelming experience, especially for individuals with a more introverted personality. In extreme cases, it can even lead to the abandonment of the course.

Having in consideration the huge impact and constant evolution of online communities, social networks and virtual worlds for collaborative activities, the use of these realities in the context of student integration seems like an interesting option.

Another important goal is to promote common experiences between newcomer students and promote their communication and collaboration. With all this in mind, it has slowly become apparent that the creation of a digital game could be a possible and viable solution, capable of solve some of the actual problems. Therefore, the core task of this work involves the development of a multiplayer digital game with very simple and specific objectives.

Games are commonly associated with the goal of entertainment, but they can also be used for learning, training, marketing and community-building [Nov05a]. They are also segmented into genres, determined by the game mechanics [Nov05b]. When developing a game, the orientation of their genre can be often conditioned by the main goals, as for instance, it's commonly used the simulation genre for training purposes.

The development of a game can span from a single person to a team of hundred professionals and this process can take from a few days to several years. Clearly, the resources and the duration of the development of a game depend heavily on their complexity, but generally their development process is similar. As refereed in [IB10b], only after determining the game goals and game concept should be selected the delivery platform/s and development tools, although in many cases this is already pre-established. Even if the concept had made these choices obvious, it may also work in several platforms.

## Context

In an attempt to help student's integration processes, the Faculty of Engineering of the University of Porto (FEUP), at Portugal has provided several activities organised in a preliminary mandatory course named "Projecto FEUP" with 2 ECTS. This work tries to answer if a multiplayer digital game could be used as one possible and valid measure, and eventually contribute to main objectives of this project.

Real world activities could obviously be implemented, but in this case, with near one thousand new students each year, the infrastructure, costs and time needed to implement could be heavily reduced with the use of a digital approach.

From here, and accordingly with the results, this work could evolve to achieve an optimal use and be actually implemented in the first year of each course, if the conclusions prove that this type of solution could be beneficial for the “Projecto FEUP” objectives.

## Motivation and Objectives

There are several reasons that impel this work, and as mentioned earlier, the use of a virtual environment can help the adaptation process of a student in a more flexible way to implement. It can offer an alternative rich experience, and at the same time, save resources and money. But probably the most important are definitely the results that can be eventually retrieved, to help answering some of the following questions:

- How important to the students is the integration process and how can they benefit from it?
- What strategies can be used to help student’s adaptation in university through the use of a virtual environment?
- Is this approach viable, and if it is, what can be done in the future to improve it even more?

Therefore, the main objective is to develop a multiplayer digital game supported in a virtual environment that provides those results with the possibility of being scalable and ameliorated over time. To achieve this, a series of methodologies are foreseen as vital for the adequate implementation of this work and its validation:

- Benchmark analysis of works similar in context and possible platforms.
- Definition of a game design solution compatible with collaborative tasks and group dynamics.
- Selection of a technological platform adequate to game requirements - for prototype implementation and testing.
- Development of assets and game logic (programming) for prototype.
- Prototype testing with student sample and results evaluation.
- Final conclusions and reflexions about future work.

## Thesis Structure

Beyond this introductory chapter, this thesis presents five more chapters. In chapter 2, is described the state of the art in the area of multiplayer serious games, presented some related work and also several technological platform for prototype implementation. Chapter 3 presents the platform and game design options that were made due to the specific characteristics and goals of the work. In chapter 4, it is described what technical methodologies were used for the

implementation of the prototype, as well the difficulties and limitations found during the development process. Chapter 5 presents the evaluation results and statistical data performed by the prototype in a real case scenario. Finally, in chapter 6, some conclusions, reflections and ideas for future iterations are drawn.

## Contributions

- Using games to promote student integration in universities through the use of virtual worlds, SLACTIONS 2010
- Technical Analysis and Approaches for Game Development in Second Life, CISTI 2011
- Game Design Evaluation Study for Student Integration, SGDA 2011 (yet to be published)





# State of the Art

Although industry-standard terms in the videogame area can become a little blurry, one major categorization resides in the context of the game goal [Nov05a]. And having into consideration this project game goal, probably the conclusion is that it falls under the section of serious games.

As presented in [IB10a], “*serious games are designed to act as conduits for each of the following:*

- 1 – The transfer and reinforcement of knowledge and skills.*
- 2 – Persuasive techniques and content aimed at changing social or personal behaviour (this would include games that promote, market, and recruit).”*

The company Forester Research describes serious games in more broad fashion: “*the use of games and gaming dynamics for non-entertainment purposes*” [SHS05]. In this case, the end project its not aimed at the goal entertainment and tries more to follow the purposes of “changing social or personal behaviour”.

Another game element relevant for this state of the art analysis is the player mode. Games are usually segmented into single-player, local multiplayer and online multiplayer [Nov05c], although other more detailed configurations could be used, like for instance, local two-player, co-op or massive multiplayer game. The possibilities offered in online multiplayer games are very interesting for this work objective, although it could also be implemented as local multiplayer game. But if is true that the videogame industry effectively started at the late seventies, the online multiplayer games only become more abundant during this last decade [Mak10]. The reason is obviously related with the advances in server side technology and internet connection speed.

The best approach to implement a suitable interaction design and adequate game mechanics in this work are not obvious, but a survey on some works that are related with the area of multiplayer games, team integration and collaborative game design can help to shape some structural concepts.

Also, an analysis of the actual engines and platforms for the prototype implementation should be performed for the selection of the more suited in the context of this work.

## Related Work

The following game concept references are in some degree related with the design of collaborative/social games or with some sort of an integration process. They are also presented independently of the platform in which they were implemented.

### eEscape

A project called eEscape [MK05] studied and analysed the adequate implementation of puzzle games in a virtual multiplayer environment. The experiment was applied to a total of six groups of four players, where each one was in an isolated room. It also collected the interaction player data through the use of several methods. The conclusions that resulted from this work provided a better perception and knowledge in multiplayer puzzle design and collaborative gaming in general.

The main objective of eEscape project consists in the study and analysis of an adequate implementation of multiplayer puzzle games in a virtual environment [Man02]. This project collected iteration data from players through several methods and their conclusions provide a better perception and guidance in the game design of multiplayer puzzle games.

eEscape tries to answer some questions like:

- “How can be developed a puzzle in a sense that collaboration is required?”
- “Is there any possibility to go further and use other methodologies than trial and error?”
- “How much can creators advance in collaborative development field?”

The eEscape game consists in total of eight different puzzle/levels. For the purpose of this experiment, six groups of four players were conducted to individual and isolated rooms to avoid any interference. Although some levels could be solved individually, the majority needed that the group work together to accomplish the task ahead. Nevertheless, some problems had several ways to be solved.

The results were collected through the use of inquiries, video capture of player behaviour, in-game video recording, audio recording during game-play and individual interviews in the end of the session. After the data analysis, some of the results and conclusions were as follow:

- The players had the perception that they work constantly in team although that was not exactly the case. In several situations, the players were dispersed and some of the tasks were performed only by part of the team or by a single player.
- In general, players more familiarized with digital games ended as group leaders. Nevertheless, in situations composed by puzzles that required diverse skills, that leadership could change, usually to the players more comfortable with that kind of puzzle.
- The lack of pressure under the resolution of puzzles, in conjunction with the absence of rewards and penalties, lead essentially to individual action and chaotic trial and error attempts by the players.

- It was detected some conflict between players, not exactly derived from the competition factors, but more by some divergences in opinions and points of view.

In the context of Project FEUP, there are some important characteristics in this project that should be taken into account. The first level was conceived to avoid distracter elements, where each player was concealed in a cell to promote conversations with colleagues. After that the cell was opened the gathering between team members was experienced with some excitement.

The design of some levels contemplated several possible solutions. Still, some players even surprised with some unpredictable solutions on their own.

Also, according to the author, establishing a balance between the easiness of playing for casual players and the use of reward/penalty mechanics can be complex to implement

## **EduTeams**

Another project named EduTeams [ET04] was initially developed as a case study, becoming afterwards a commercial product. The main goal was to broaden the core skills of students in elementary and secondary schools. That includes teamwork, communication, planning, problem solving and logical thinking. Following an activist methodology (learning by doing), the result was a multiplayer system with several team based activities. EduTeams was implemented in disciplines like math, geography, technology and languages, but others fields like chemistry and biology are expected to be explored in a near future.

During the evaluation of the case study, professors reported some benefits in classroom. They perceived that some students became more motivated, more extroverted and better aware of the importance of teamwork. There were also published three final reports where some conclusions were drawn:

- Computer games can potentiate the learning process of soft skills.
- All reports pointed that the communication between students improved substantially.
- The major identified problem was related to the time needed to learn how to play the game.

In the context of Project FEUP, it's important to notice some curious events in this project. First of all, the behaviour of a student that tried to impose his leadership slowly shifted to a more teamwork attitude and accepting orders from other classmates, especially after verifying that his group results were the lowest. This behaviour change was never detected before by the teachers.

Other relevant situation was the case of an introverted student, that although clever and very hardworking, hadn't enough confidence to speak in classroom, even when asked a direct question. By using the game, teachers observed that the student had become the team leader and her confidence and participation in classroom had increased substantially.

## **Revolution**

Revolution [REV05] is another serious game that gathers students in a virtual world. Here, the main objective is to give players better perceptions of the American Revolution, granting the possibility to interact as an inhabitant character of the colonial city of Williamsburg. This game uses role-playing principles and provides seven class options of characters and a non-linear

narrative story determined by the player's choices, resulting in a complex simulation of the political and social system of that time.

This is a multiplayer game with an exact length of 45 minutes where the authors recommend that students play simultaneously, in a way to provide rich interacting experiences through the role of character class in that historical context. In a very fast pace, the narrative will present different situations and problems where players must take important decisions.

In the end, this game turns itself into a complex social/political simulation, with several elements of role-playing and a strong storytelling component.

Nevertheless, the results of this project revealed that the gap between experienced and non-experienced users can compromise the end result experience. More experienced players even exploited some of the engine functionalities to "cheat" in some way during the game, like stealing items for instance. This also brings implications about the expectations of hardcore gamers and casual gamers.

Some observations in this case study could be interesting for the Project FEUP design process:

- To the contrary of the majority, Revolution implements in a multiplayer online game a very strong narrative and dramatic content.
- Each character has a set of specific objectives, according to its profession/class.
- There is a story and a past to each character, seeking to generate empathy in the player and thus achieve a better understanding of their motivations.
- The players can talk to each other and solve problems in a collaborative way.
- The authors entitled the game a "Multiple single-player game" because there were included elements usually present in campaign game modes. Prove of that are the long list of non-player characters and the narrative evolution that can be radically changed over time.

## **Technological Platforms**

The following section presents some platform options for the purpose of game development and analyses each system advantages and disadvantages in the context of this work.

### **Unreal Development Kit**

UDK is a high-end game engine powered by Unreal Engine 3 technology. This application can be used free of charge for non-commercial projects and offers essentially the same level of technical capabilities that are used to produce AAA games. There are available an enormous amount of features like a complete editing environment, real-time shaders, real world physics, artificial intelligence, cinematic animation, built-in networking, particle effects, scripting, etc.

It's a very complete system that allows top graphical results, but is also complex and very resource demanding. The publishing options are also limited, as for the time of this work only the windows version was available.

#### **Advantages**

- High performance engine with top graphic capabilities.
- Integrated editor available with long list of features.

### **Disadvantages**

- High specification system for the end-user
- Not possible to distribute game for multi-platform systems
- Not possible to publish on browser.

## **Adobe Flash**

Adobe Flash is and multimedia production platform, especially designed for 2D animation. Nevertheless, it possesses a robust object-oriented-programming language (Actionscript 3.0), that enables the development of games through scripting. Content implemented in Flash can be viewed and played essentially in any system, device or browser.

This application is very flexible for the creation of 2D games, but it is also possible to use 3D in Flash through third-party classes like Papervision3D or Away3D, although performance can be easily compromised when more than 1500 polygons are being used [PAC10].

### **Advantages**

- Easy distribution for the majority of systems.
- Very flexible for 2D game development.

### **Disadvantages**

- 3D technology is limited and only available through coding (no editor available).

## **Unity**

Unity is a game development tool with an optimized pipeline for DirectX and OpenGL graphics with an easy-to-use editing environment. Content developed in Unity is usually aimed for web browsers (plug-in required), but recently the distribution options in this platform have become broader, allowing its use in mobile devices and even consoles. This platform is available in a commercial and non-commercial version.

### **Advantages**

- Easy distribution for major systems.
- Good graphic capabilities, especially in 3D.

### **Disadvantages**

- Plug-in not so well distributed.

## **Second Life**

Second Life is an end-user virtual online world developed by Linden Lab since 1999. Virtual worlds like Second Life have recently become a possible option for game development, offering some interesting and unique characteristics that could be suited for some type of implementations, as they excel in interaction, multiuser experiences and the prebuilt environment engine may decrease development costs and time. It can be also be the best option if the targeted user demographic is already a resident in the virtual world, but new or

inexperienced users may have a lengthier learning curve, as they need to become familiar with the interface of the virtual environment, even before the game starts.

#### **Advantages**

- Uses a persistent online virtual world.
- Ready to use functionalities like avatar creation and communication interfaces.

#### **Disadvantages**

- Subscription fee needed for advanced content creation.
- Game must be build under the virtual world environment.

## **Summary and Conclusions**

This chapter focused on related works that, in a way or another, share some characteristics with what is pretended in this project. From this benchmark, some important facts should in fact be taken into account, especially during the game design process. One of them is the possibility of players lose their focus on the objectives and disperse or become distracted inside the virtual environment. Other is the awareness needed to create a balanced experience to different types of players and avoid disruptions possibilities during gameplay (although always very difficult to prevent unforeseen flaws without heavy beta testing). Even so, there are also good indications that multiplayer games can eventually help individuals to become more confident, communicative and integrated in a social community.

There was also conducted an analysis of possible platforms for prototype implementation. The selection must have in consideration the objectives of the project and the target system/s for distribution.

# Game Concept – Ideas and Requirements

For the creation of a game concept, some elements must be well. First and foremost it is necessary to define the game goal. In this case, the goal consists essentially in providing a multi-user experience to promote team building, collaborative activities and decision making. The user demographics were also important and clear in this case: university students from both genders, with an age of about eighteen years old.

Experts in the field of psychology were consulted for some guidelines and orientations to an adequate approach of the game concept. Several ideas were drawn and although no general consensus was achieved, the use of competition inside the game was pointed as plausible if implemented correctly.

With this, and to achieve an adequate result, it was needed that the design holds actions between the players that evolved around the concepts of communication, synchronization, leadership, task roles and team dynamics; with the main purpose of become a “lightweight” experience generator for newcomer students at FEUP.

As one of the main objectives is to provide amicable experiences to users, it follows naturally that the game should be played in teams of humans, in a cooperative and adversarial environment. Other parts of the “Projecto FEUP” course group students into teams of 6 to 8 players and these teams should preferably be kept and used to enhance “team-effect”.

One of the problems of the project was the high number of students that should be integrated under the virtual platform. Each year FEUP has more than 800 newcomer students, so, coping with such high numbers is a requirement and as such games should have several rounds and or different levels to foster all vs. all experiences. Naturally, levels should have at least variety and preferably increasing complexity in order to create stimulating challenges.

Another strong idea is to have some kind of reward mechanism: in similar projects, some reports state users tend to lose focus and just wander around when there is no reward or competition involved.

It was also considered that fostering communications was interesting and students should preferably (but not mandatorily) seat in the same classroom. As with other “classes”, timing is of the essence and timing constraints should be in place, also in order to cope with the very large number of students.

Another idea is that not all students would be familiar with these technologies and that a small trial area (that would not break the novelty) would be interesting. Users should thus have an area to test the game environment and mechanics.

As with many other games, it was considered that teams should change sides in the games and that the games should, of course, be balanced and equitable. Circumventing game rules and un-ethical behaviours should be avoided and punished in the game rules.

Also due to the large number of students involved, preferably, little or no supervision should be needed. There may also be in-game supervision for remote players.

## Game Design and Proposed Game Levels

The mentioned project game was codenamed “FEUP Adventure” and its concept was a multiplayer action game with three distinct levels. Each level needs to be played by two teams (acting and opposite team). The proposed virtual space has three game rooms and an additional free practice area. The game levels are based in controlling a player's character to avoid obstacles and eventually move objects. The goal is to score more points than the opposing team.

Each level is composed by a specific set of rules and objectives, designed to promote the creation of collaborative and adversarial experiences. Levels were designed to be simple but allowing different strategic approaches and promoting communication and interaction among colleagues, in an attempt to achieve the proposed goal of the game.

The proposed solution to coping with many students is set up the games into groups of 4 teams. A competition between teams is used to reinforce the focus on the tasks ahead, where a classification system is implemented and rewards are available accordingly with the results.

Three levels were found to be an adequate number for the test purposes at hand. Each level needs to be played by two teams (acting team and opposite team), and the three different levels are played with distinct teams, assuring that all teams will face each other (as depicted in table 3.1). For example, initially, team A will oppose team B and simultaneously, team C will play against team D.

**Table 3.1.** Team Distribution by Level.

	Teams
<b>Level 1</b>	A vs. B C vs. D
<b>Level 2</b>	A vs. C B vs. D
<b>Level 3</b>	A vs. D B vs. C

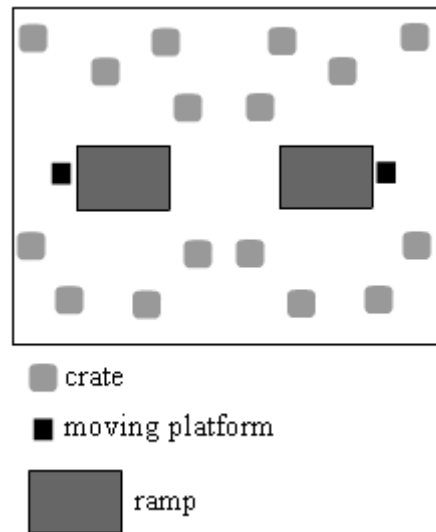
The proposed virtual space has three game rooms and a free practice area. The levels are based in physics, moving objects and orienting the avatar of the player's character. There will no flying available due the nature of the levels, and therefore, the practice area will allow the students to become more aware of the interaction with the avatar and test the actual game mechanics before the game begins.

Each level is composed by a specific set of rules and objectives, designed to promote the creation of collaborative and adversarial experiences. Levels are to be simple but allow different strategies, still demanding interpersonal communication, thus promoting interaction among colleagues.

The game at Level 1 is called “Crate Carrier” (see Fig. 3.1) and is a 6 vs. 6 player level, where the objective is to score the most points possible during a limited time (e.g. 5 minutes). Each team will score points by carrying crates over their ramp and dropping them into a moving platform. To ensure a balanced game for each team, the position of the objects is symmetrical and in the beginning all team members will start on the top of their ramp. The crates were

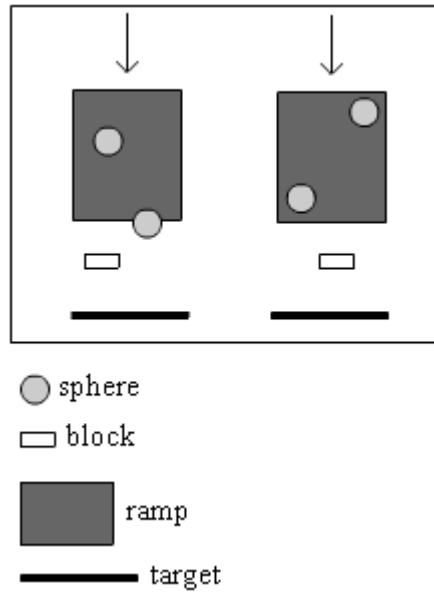


designed to be heavy objects, hard to move and control singlehandedly. To score points is also required synchronisation between the players, as they must all push together at the exact time the platform passes underneath. This level offers several strategies that can be adopted by the teams, but players cannot stand in the opponent ramp.



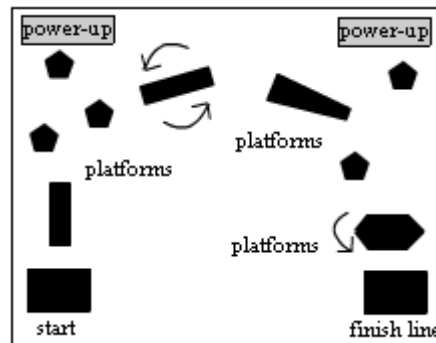
**Figure 3.2.** Map Design from Level 1.

Level 2 is called “Sliding Spheres” (see Fig. 3) and it is proposed that four players of a team play and the objective is to score the points during a short limited time (e.g. two and a half minutes). Each team will score points by direct falling spheres through a ramp into a large target. Each time a sphere hits the target the team will be rewarded with one point, but there will be two members of the opposite team that will try to catch the spheres with a block. The block wall can only be moved along the sideways of the ramp, in this case, only to the right or left. If any spheres touch the block wall, it will disappear. The spheres will appear in intervals of five seconds and the two teams will be present in the same room, with four players in the respective team ramp and two players on the other ramp trying to catch the enemy spheres.



**Figure 3.2.** Map Design from Level 2 – Sliding Spheres.

Level 3 is called “Jumping Platforms” and was initially conceptualized to be a more complex level (see Fig. 4), but technical limitations related with the avatar translation restrained some of the initial ideas. Essentially, the game consists in a large room with several objects that work as platforms and obstacles. Each team will be represented by four players and the objective is to reach the finish line by the same four players the soon as possible. In this case there are no points, only time. The timer will stop only when the last team member reach the finish line. Due to the nature of the game, it will be needed one session by team, in opposition to the other games where one session abridge the results for two teams.



**Figure 3.3.** Map Design from Level 3 – Jumping Platforms.

In related work projects there are also reports that some students tend to use some disruptive behaviour during gameplay [REV05]. It’s necessary to consider that students may use the game mechanics to completely counter the active team tasks. For example, a single player could be on the top a ramp and easily prevent any coordinated work by the other team. To avoid this, students’ avatars cannot cross adversary ramps or platforms. The game has to be well balanced in order to avoid too much frustration, especially when a team is working together.

Each student should create their own avatar with a name and personal characteristics to become easier to recognise, especially among colleagues. The challenges presented in the multiplayer game may even be beneficial to the final grade (as a reward mechanism).

## Platform Selection

The first criteria elements for the platform selection involved the end result distribution and visual appeal. UDK was very limited in the publishing chapter and eventually discarded. And to reach players high expectations, the choice ended in a three dimensional environment instead of a two dimensional, putting Adobe Flash aside as well.

Having into consideration the delineated game goal, game concept and a closer look to the remaining options, some recognizable advantages stand out for the Second Life option:

- Structure already provided several elements built-in at its core that were needed for the prototype implementation, like a 3D environment, character creation and a networking solution. This contributes greatly in the reduction of the development time.
- Second Life's capabilities outside the scope of the game could also contribute for the goal (student integration and student community building), as users could maintain contact with each others inside the virtual world in other activities.

It was also considered that joining "Projecto FEUP" and "FEUP Adventure" with the previously existent Second Life Island branded with the University of Porto (U.P.), was interesting in order to foster both activities and would allow for the necessary basis for development of the game. Naturally, the development platform should have near real time response and some kind of physics engine or at least interactions should be easy. The 3D part and the persistent world were considered optional.

However, using the Second Life Platform also carried some identified drawbacks for the specific application at hand:

- Terrain must be purchased to Linden Labs (one time only)
- Non negligible minimum time to start playing the game
- Must have internet access to a "dedicated" server
- Executable client application must be installed – administrative user rights needed
- Adequate hardware needed (including 3D video card)

Clearly, the strong points surpass the drawbacks for the game FEUP Adventure game, developed in a very short time of development. Finally, further testing was needed to acknowledge if the platform was flexible enough for the game mechanics that were needed to implement, already defined in the concept.

## Game Development in Second Life

Second Life is composed by regions, which some are completely dedicated to work as a game. The fact that Second Life offers a prebuilt engine with a solid networking system and a multi-platform web client were key factors for its selection in this present work. Of course, Second Life platform may present limitations for game development, like the use of a non-object-oriented scripting language and a somewhat rigid interaction model, but those are minor drawbacks for the purpose of this project, as the mechanics need to be simple and straightforward for an inexperienced user target. The technical implementation is being

developed in a parcel of the University of Porto's island with the use of custom assets (models, textures and scripts) for the construction of the game and its levels.

Other aspects also need to be taken into account, like the adaptation of the student to the Second Life client interface and in-world interaction controls and the overall performance. As stated in [DGP06], it's not recommended that the game possess complex mechanics and it shouldn't rely heavily on numerous prims and high frame rate scripts. Usually, attempts of that kind resulted in a poor user experience and presented high latency problems.

While some private regions in Second Life are completely dedicated to work as a game, they can also be available in the top of a simple table or panel board.

Darklife [DL03] is an example of a complete region dedicated to a multiplayer role playing game, where players can work solo or as a team to defeat monsters, gather gold to buy items and receive experience points to develop their character. Beyond the fact that uses items, like swords and shields, to increase the character attributes, it uses two very special items that are essential to play the game. One is the backpack and the other is the Heads Up Display (HUD). The backpack is a wearable object that controls the interaction between the player and the environment present in the game. Without it, it is not possible to play or interact with the other objects. On the other hand, the HUD provides the statistics data for the game, like the health points, magic points, gold, etc, and also some command operations like create a team, level up and define options.

The Pot Healer Adventure [PHA05] is also a game implemented inside an entire region, but in this case is aimed specially for single player use. This is an adventure game where the player must explore a strange world and solve some riddles to progress in the story. It has a narrative that is essentially described in a written form, available in most cases through note cards. It is also used a wearable item, in the form of a notebook, to save the game progress.

The previous case studies have the primary goal of entertain the player, but probably Second Life is a more adequate platform to the goals of learning and training. It is often used by universities and schools for teaching and research, as educators are constantly looking for the potential of games in virtual worlds to foster experimental and constructivist learning methodologies [Inm10]. Ohio University within the Science and Technology Enrichment for Appalachian Middle-schoolers (STEAM) project developed a series of educational games for science classrooms [CCL08]. Some examples are the Fruit Fly Genetics and Weather Challenge where both use extensively the HUD for the player interaction with the game.

As observed, in many situations of game development under the Second Life environment, it's common practice the usage of wearable objects that controls aspects of the game and HUD's for present game information and offer additional interaction options. Nevertheless, this two implementation approaches are not mandatory for the creation of a game in Second Life.

## Summary and Conclusions

In this chapter the game concept was designed accordingly to an attempt to promote communication, peer recognition, leadership and above all collaborative work. The result was three proposal game levels, besides a training area, based in the action/platform genre, where the main mechanics work around transport large and heavy object in group and with synchronization.

There was also showed the main reasons behind the selection of Second Life, which accordingly to the game goals and concept, appeared to be a plausible option with distinguished advantages. In this case, Second Life, in spite of being conjectured some identifiable limitations, offered two major advantages: a built-in engine structure with important functionalities already implemented and the possibilities that the virtual world offers for students outside the scope of the game concept, but still attending the project objectives.

Some game projects developed in Second Life were also analysed for better shaping the prototype implementation methodology. It was observed that it is very common the use of

wearable objects for the control of the game workflow and avatar animation, as well as HUD's to present game information to the player screen.



# Implementation

Independently of the platform and tools, any project to develop a game is essentially segmented in four distinct parts during production: a design part that encloses the game concept and game mechanics; a coding part that is related to programming and aggregates the game engine and game logic; an art part that includes all the assets needed to the game like sprites, 3D models, animations, audio, etc; and finally a management part that is specially important in large projects [Bet03]. Although the design and management parts are not immensely affected by the delivery platform, the development code and art parts could be heavily conditioned.

The technical development of this prototype work was segmented in three implementation phases: asset creation, level editing and programming logic. In this chapter it is presented a general overview of each one of these phases, known issues and the overall result.

## Asset Creation

The modeling of objects in second life can be created by a set of tools available in the Second Life Viewer or by a 3D application. The tools in Second Life only permit the construction of basic primitives, that when assembled in group, can recreate more large and complex forms. When this approach wasn't enough, the modeling task was developed in 3D software and additional pug-ins were used to convert the model to an image file. That's because a model in Second Life is just an uncompressed image file format, where each pixel colour represents a vertex coordinate in 3D space. In this case, the red portion of the pixel represents the x axis, the green the y axis and the blue the z axis. As the image that represents the model has a fixed dimension, the number of polygons cannot be changed during modeling operations. This means that any attempt of extrusion in the model would ruin the file conversion and it would not represent the intended result.

As the conception of each game level defined that objects needed to move and react in pre-established conditions, the use of physics and collision detection methods determined an important role between interactive objects. One of the key aspects of the game-play was the balance between a not too heavy object that must be moved and a too light one, as the final objective is to produce an object that is difficult to transport alone, but without making it impossible or too frustrating.

By activating the physics functions on the objects, some technical issue have arisen in the "Crate Carrier" level, as the crate boxes turn out to be constantly stuck in the junctions of the

floor blocks, even with them aligned in same “pixel perfect” coordinate. The solution was to use a sculpting tool to produce some round edges in the corners of the crates.

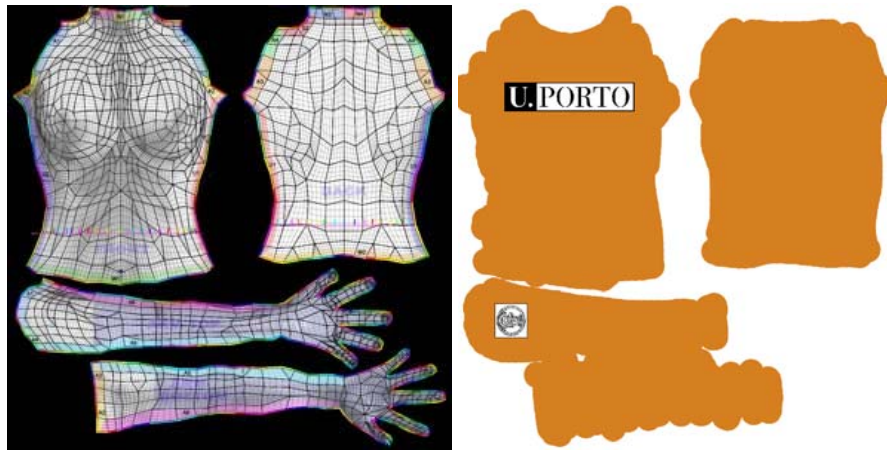
In Second Life, the available types of content for creation are 3D models, raster images, sounds and animations. 3D models can be developed directly in the virtual world, through the use of the Build tool, allowing the creation of three-dimensional primitives, usually called prims, like cubes, spheres, pyramids, torus, etc. These are prebuilt models limited to a minimum dimension of 1cm x 1cm x 1cm, and a maximum of 10m x 10m x 10m. The user can alter some of the object parameters like size, rotation, colour/texture, and other features related with the behaviour of the object within the virtual world. There are also some limited support for the modification of the primitives form, including hollowing, path cutting, and some rudimentary implementations of twist and other functions. A large part of the existing models in Second Life are simply constructed with this tool, for although the available prims are only fifteen, the most practical approach is to group several prim instances together (like building blocks) to create more intricate models.

There is currently no support for mesh files in Second Life, although it's aimed to be implemented in a future version of the player and a new beta mesh viewer is available for testing with COLLADA file support. The closest approach to mesh creation available at the moment is through the use of sculpted prims, also designated sculpties. A sculpty is a Prim-type polygon mesh with a fixed count of vertices (usually 1024) that is applied to a single prim. In reality, the predefined shapes can only be sculpted and bended, but the mesh itself must be kept with the same number of polygons. This mesh data is stored in a RGB image (usually 32x32) where each red, green and blue colour value of each pixel represents a coordinate in space (x, y, z). This shape-defining texture is called the sculpt-texture or sculpt-map and in this way the base model becomes deformed by these colour stored values. Nevertheless, this is still highly limited and due to their graphical nature they might load slowly. It should be used lossless compression or the pixel colours would become defaced and the model will have a different representation relatively to the original design, usually a complete deformed and misrecognised one.

There are several modeling packages specifically designed for the production of sculpted prims, like Sculptypaint, Rokuro, and Tokoroten [Hea07]. But other 3D software packages can also be used for modeling sculpties, usually with plug-ins. Other practical approach is to export the model to an OBJ file and convert it to the sculpty image through a converter [SM11]. The most limiting factor is that the model must have a precise count of 1024 vertices at their higher resolution and therefore no geometry can be added or removed.

Prims and Sculpties support the appliance of textures with some parameters related to tiling, transparency, bumpiness and a few more. For mapping, only the default and the planar projection can be used, meaning that in cases where detailed texturing is needed, baking the unwrapped render result in a proper software is often used to workaround this situation, especially where pixel density across the sculpty surface is not uniform. Character cloth texturing is also available and common templates are used for mapping (Fig. 4.1 and 4.2).





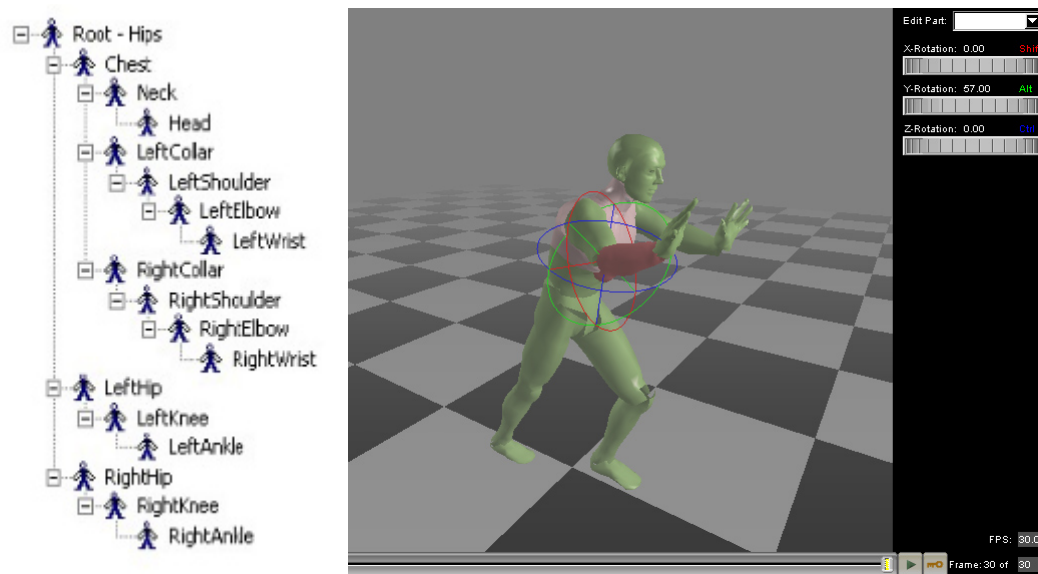
**Figure 4.1.** Texture mapping template for character cloth.



**Figure 4.2.** Texture applied to character avatar.

In Second Life, animations files are only applicable for player characters (avatars). There are several built-in animations available, but animations can be created in the Biovision Hierarchy (BVH) file format. The BVH is a non-XML hierarchical file format that specifies the angles between the joints of an avatar's body parts.

One drawback is the lack of a full definition of the basis pose [CIR09] (this format has only translational offsets of children segments from their parent, no rotational offset is defined). It also lacks explicit information for how to draw the segments. While it is possible to create BVH files in a text editor, it's more practical to use third-party programs, such as Avimator [AVI11]. Through the specified keyframes, eighteen body parts can be rotated in any axis and the root (hip) translated for character animation in Second Life (Fig. 4.3). Other types of rigging are not possible and animating prims or sculpts is only possible by scripting.



**Figure 4.3.** Animation development to apply into player characters (pushing objects)

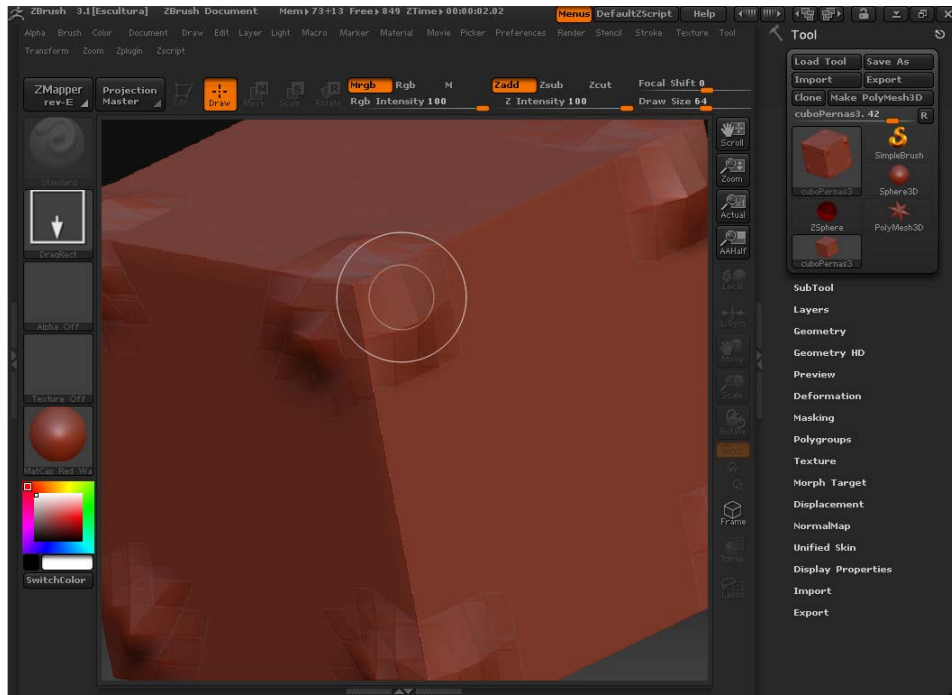
Audio files are also a type of asset that can be used but have specific restrictions. They must be in Microsoft Wave format (wav), have a sample rate of 44100 Hz and a total playtime of less than 10 seconds.

## Level Editing and Game Assemblage

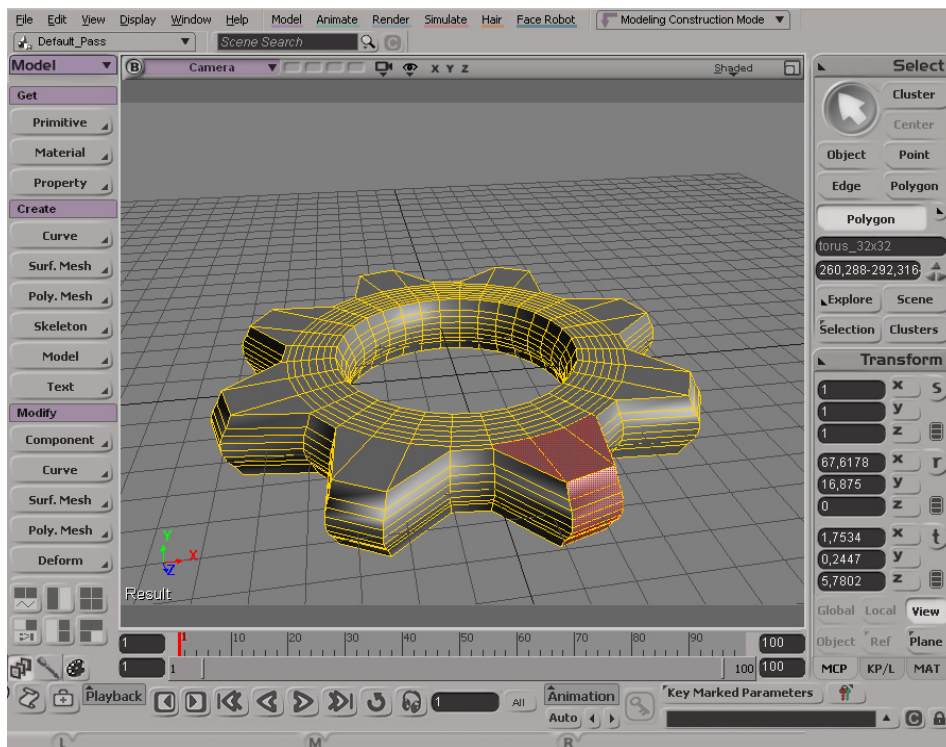
As stated above, the production of a game in Second Life is essentially comprised by art assets, game data and scripts files. For this game prototype the principal art contents were three-dimensional models, textures and animations.

The major parts of 3D models created for this prototype were developed with Build tool available inside the platform itself. But when a more detailed model is needed, probably the best approach is to use a sculpted prim. This method can also be used to reduce the prim count inside a region or parcel, as these have a limit number. Instead of having a group of several prims to represent an object model, it could be used only one sculpted prim with an equivalent representation.

As expected, the fixed number of vertices that a sculpty must have becomes very restrictive in modeling concerns. Any operation that adds or removes geometry, like extrusion, bevel, inset, etc, cannot be used. Usually the process involves the creation of a start-off primitive with a fixed size of 1024 vertices and proceeds with the edition of the model. The most common operations for these tasks are the appliance of modifiers and/or SRT (Scale, Rotation and Translation) adjustments to the position, size and rotation of the model components (vertices, edges and polygons). This can result in a very time consuming task, even for very simple models. For this prototype Autodesk Softimage was used with XSI Sculpt plug-in for the creation of more mechanical and hard-edge type of models (Fig. 4.5). Pixologic Zbrush was also used as a sculpting tool for organic type models (Fig. 4.4). During the development, it was noticed that even with the floor blocks (3D box prim) perfectly aligned in the y axis, the crates (3d cube prims) could eventually become stuck by the physics engine in the threshold of each floor block. Zbrush was also used to soften the cube shape and the result was a smother translation of the crates inside the game engine.



**Figure 4.4.** Crate sculpting for better movement during player interaction.



**Figure 4.5.** Asset modelling in Autodesk Softimage.

Animations were developed in Avimator and applied to the games wearable object, a wristband. This allowed player characters to perform animation during certain situations, like for instance a pushing animation when the player is in contact with an object to push (Fig 4.6).



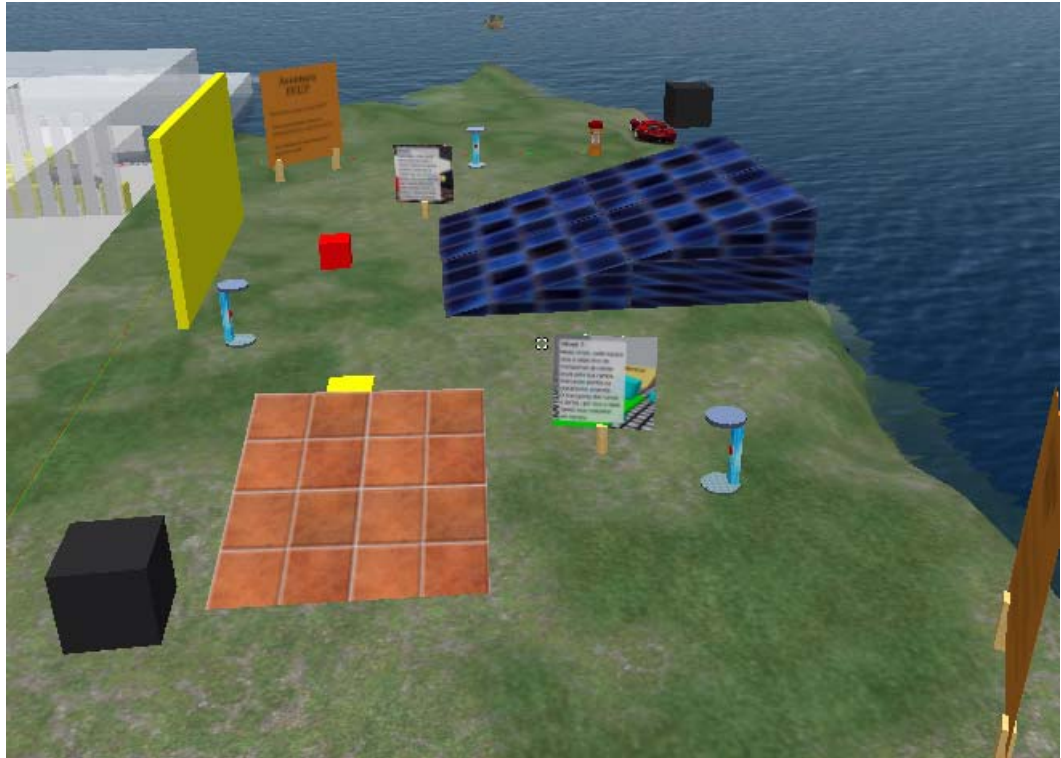
**Figure 4.6.** Avatar animation during player interaction with object

As the core of the game mechanics rely heavily on moving and pushing objects around the map level, the Second Life physics engine (Havok) was used to implement their behaviour. Some trial and error was used to figure out proper configurations for moving objects with the player characters. The physics engine calculates the mass of an object based on its size (width, height and depth) and although it's possible to define a predefined material, this only affects friction, elasticity and collision sound. There is also the possibility to program the behaviour in a non-physic object, but as setting a translation by script causes a 0.2 seconds pause over the routine, object movements can become very choppy. With the impossibility to define the mass of a physical object, some testing was conducted with size, material and shape settings to reach an acceptable result. In this case, the objective was to make the object not too hard to move around, but not too easy either, as it should be difficult enough to make the players teams work together.

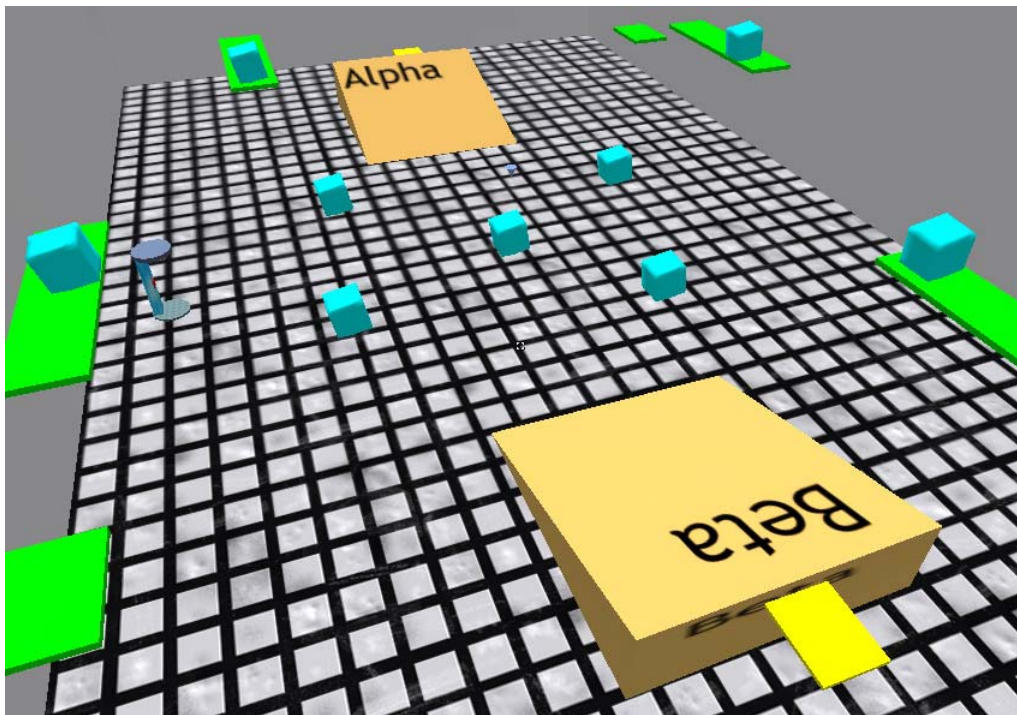
During the assembly of the third level (Jumping Platforms), it was observed that some of the conceptualized game mechanics would be very hard to implement in this platform. Object translations in the X and Z axis and rotations in the Y axis would not carry the avatar to a new position. The prototype was tested with a playable third level, but not with all the features that were initially designed.

In the end, four stages were assembled inside the university island, where the training area is of public access at the ground level and the other levels were created inside a closed "box" room on a sky high location, only accessible by teleportation (Fig. 4.7 to 4.11). The start of each level, except the training area, must be activated by a monitor teacher after each team states as ready. This action will essentially activate the stage timer and instantiate all objects needed for the game workflow.

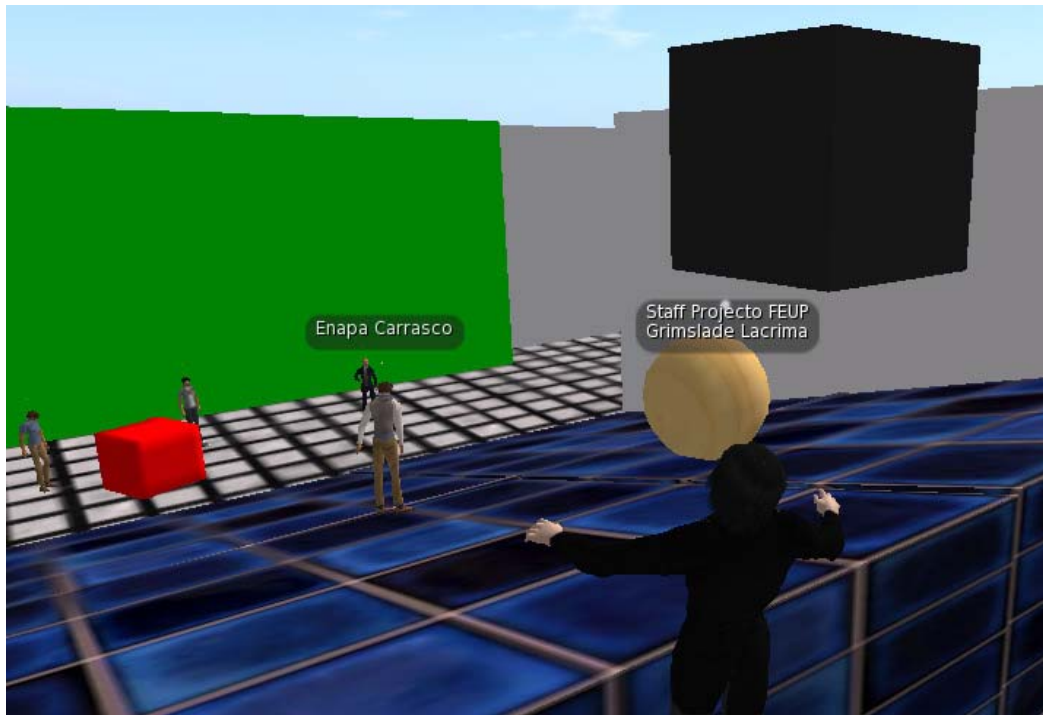




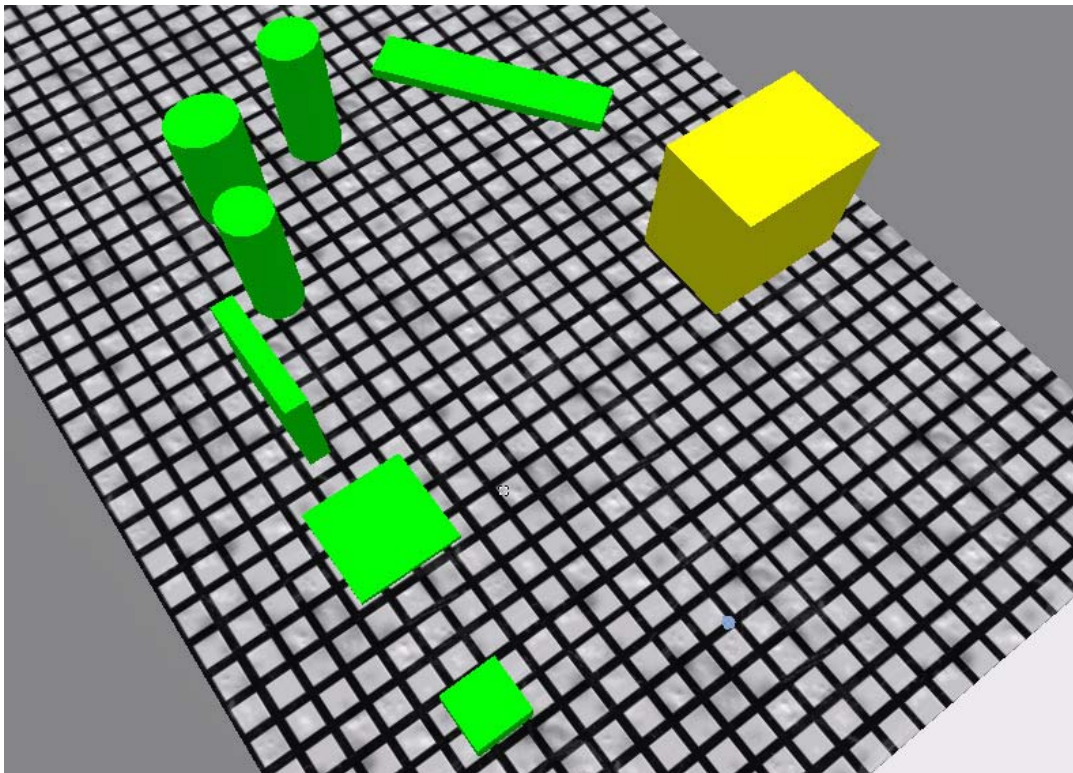
**Figure 4.7.** Practice area in the FEUP Adventure game.



**Figure 4.8.** Assemblage of Level 1 – Crate Carrier



**Figure 4.9.** Level 2 – Sliding Spheres screenshot during gameplay



**Figure 4.11.** Level 3 – Jumping Platforms during conception

## Programming

Second Life provides scripting capabilities through Linden Scripting Language (LSL). LSL is a state based and event driven programming language. Usually a script is attached to an object content folder and used to program that object behaviour in the virtual world.

A LSL file script support variables, functions and states, but LSL is not an object oriented language. Classes and interfaces do not exist and concepts like inheritance and encapsulation are not supported. This is effectively a possible limitation for game programmers, as the majority of games use the OOP paradigm in its conception. Code implementation, testing and debugging could not be the most flexible either, eventually resulting into an increase of development time. Nevertheless, LSL has more than three hundred functions and over thirty two events.

Many of the common operations used in game programming like collision detection, object instantiation, timers, etc, are present. Objects can be linked with LSL scripts, defining therefore their states, functions and listeners accordingly with event occurrences. LSL also supports server side communication through XML-RPC, although might be considered limited for certain tasks [GIA10], and HTTP protocols. HTTP allows more flexibility and scalability in comparison with its XML-RPC counterpart. It also permits a fast secure object to object communication, enabling quick and easy communication between objects. In the context of game development, this is relevant as it allows instance data exchanges and game data storage.

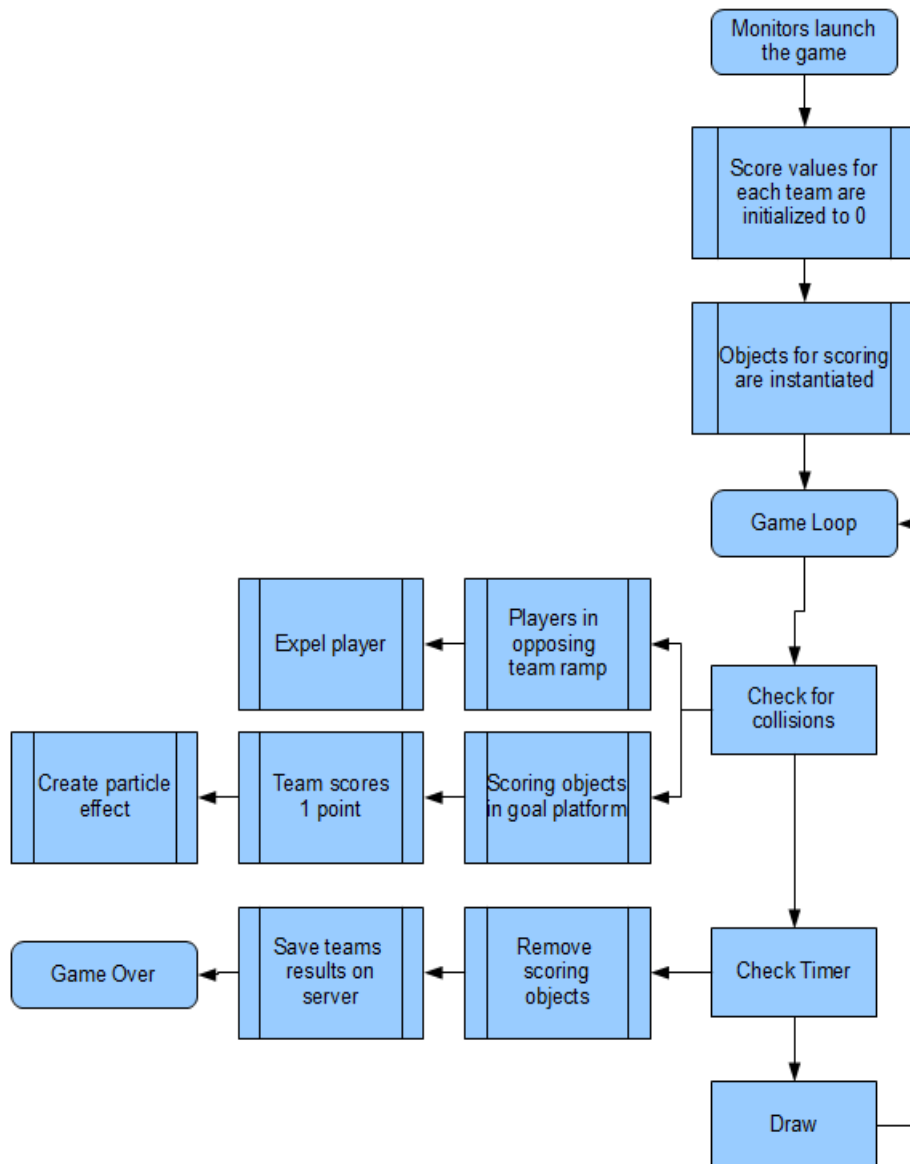
In this project implementation, scripts are stored inside objects and are activated accordingly to events or states. There is a master object that controls the start of each level and its respective asset instantiations. From there, each instance has its specific behaviour, and in this case, normally associated with collisions with other objects and translation in world space. The wearable object scripts, a common technique used in Second Life games as stated above, is in this case a wristband object that each player wears before the game start. This object allows more control over actions performed by players and in this case is used principally for animation purposes.

Although there isn't properly an update and draw methods like in most game engines, the common workflow of each level is essentially described like in the following flowchart diagram (see Fig. 4.12). Here is the logic of the most important scripts, developed for the game levels:

**Start the game with verification:** In this case, any level starts by the activation of the master object (a simple click event) from a monitor, whom must be previously assigned to the monitors group in Second Life. The master object, verifies if the avatar that cliqued belongs to the monitor group through the internal function `llDetectGroup`. In the case of being true, the game will start.

**Initializations (Timers and Scores):** Also in all levels, a timer starts counting, defining the time limit of each level. This process is achieved with the function `llSetTimerEvent` an one second interval, the time is updated and showed (in text) in the middle of the stage.

Also, the master object communicates with all objects needed to define additional initializations, like object positions or score points. This communication is implemented with the listen event through a proper channel. The score points of each team are showed on the prim box that serves as the goal platform.



**Figure 4.12.** Flowchart diagram with the main procedures

**Object Instantiation:** The master object also stores other non-script objects like models, sounds or images. Therefore, and with the use of the function `llRezObject`, instances can be created and positioned in space. In other hand, these instances also contain scripts that define their behaviour and can also instantiate other objects. This was the approach used to instantiate objects over 10 meters from the master object (Second Life space limit for instantiation). This method can become a very complex process of nesting objects. A better option can be moving the master object around the level to create the instances [Men10]. Nevertheless, the approach used in this work guaranties that all the instances are created at the same time.

This instantiated objects have the precise duration of the game level, being removed when the timer ends (`llDie` function).

**Disruption prevention:** One of the most important aspects during game design was the constant awareness needed to avoid ruining the experience for players, due to unfair play possibilities made available to the players. As mentioned the state of the art chapter, the player's attempts to disrupt the normal functioning of a game was common. In this case, and knowing



that a single player could easily prevent a team from scoring if he was in the opponent's ramp deviating the scoring objects, it was used the `llDetectGroup` to verify if an avatar belongs to that team ramp or not. This action was built inside the collision test event between avatars and the ramp (`collision_start`). When the condition verifies that the avatar is an opponent, the `llPushObject` is used to expel the avatar, projecting him backwards a long distance.

**Scoring:** The scoring system also uses the `collision_start` method, but in this case between the goal platforms and the scoring instances (crates or spheres accordingly to the level). When this happens, the scoring instance is removed from game and added one point to the score of that team (updated in the goal platform scoring text).

Is also created a particle emission of "sprites" (`llParticleSystem`), presenting an additional visual effect during team scoring.

**Ending the game:** When the timer reaches zero, all remaining instances are removed and the scores recorded on server through the `llHTTPRequest` function.

## Summary and Conclusions

The experience gained from the development of the three level game "FEUP Adventure", led to the following strategies for development of similar type of gaming applications:

- It was found that ramps and balls are objects where it is safe to have interesting physics.
- It is also interesting to define carefully the sizes of the objects involved (only a limited number of object materials are available – materials define density, elasticity, friction and default sound).
- Creation of scripts for normal mechanics of the game (example – pushing the crate animation).
- Creation of the scripts for normal dynamics of the environment (example moving platforms, etc).
- Creation of other illusory animations (client side animations for visual interest only).
- Creation of special scripts (example for game creation / initialization / instantiation / timings / collisions / special game rules – fouls, etc); in our game, the game initialization script was activated by the event associated with an administrator flipping a start switch.
- Our strategy for team formation was to create the groups as available in second life platform which turned out to be lengthy; another probably better solution would be to allow wearable objects to define teams.
- Scoring data is associated with a specific object like, in our case, our starter switch; storing data (scores, etc) was achieved by communication with the server and storing data in the server itself (server side programming needed).
- Deployment of the objects was firstly tested by editing and finding best parameters for all game elements.

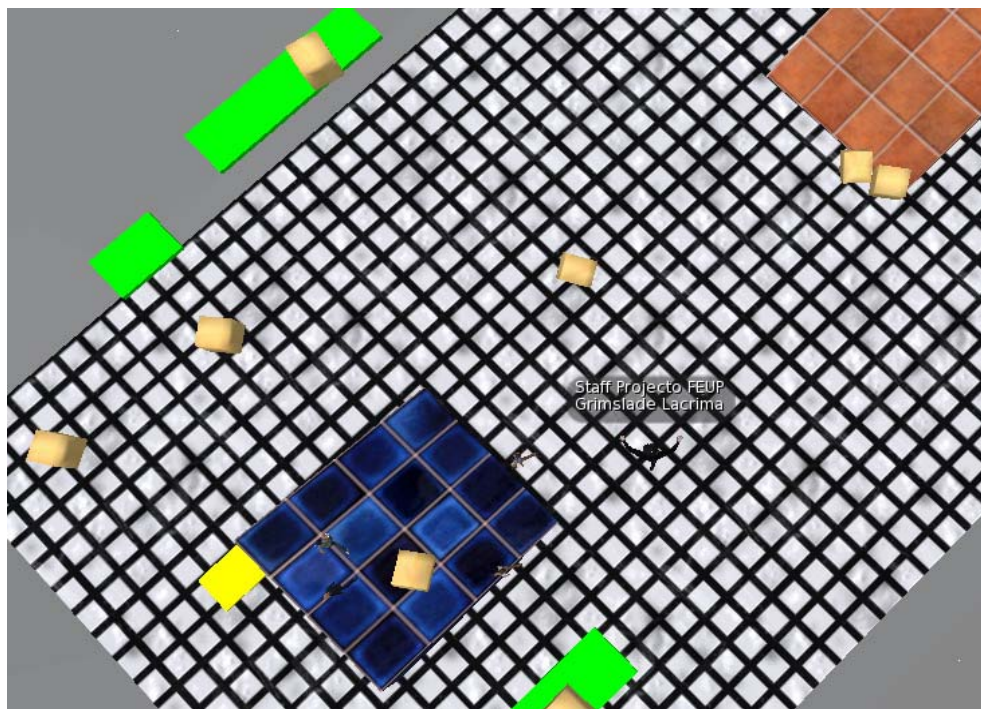
- Complex platform game genre could be hard to implement in the Second Life virtual world.
- It is possible to implement some features by using resources based on the server but networking latency will likely generate slow responses in world interaction, something not interesting for the “FEUP Adventure” game.

The presented strategies are not the only ones possible as the development platform is very flexible. The presented strategies were, however, found to be useful and interesting.

# Prototype Evaluation

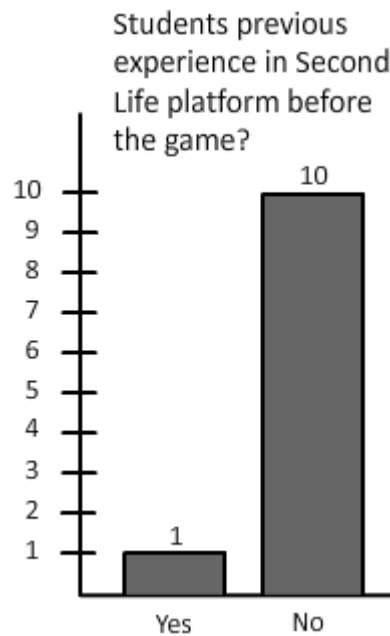
The prototype evaluation was made with a sample group of 12 students during the initial weeks of their first lecture year. No grades were associated with any gameplay. The operations were implemented in classroom with the supervision of assistants; each one had the task of monitoring and collecting student's behaviour during gameplay. Students were organized in two teams for testing the idealized game levels in practice (Fig 5.1) and in the end a final inquiry was submitted by the 11 students.

This was a preliminary evaluation. The teams were formed by the available students at the moment and there was no attempt to ensure its significance accordingly to the curricular activity.

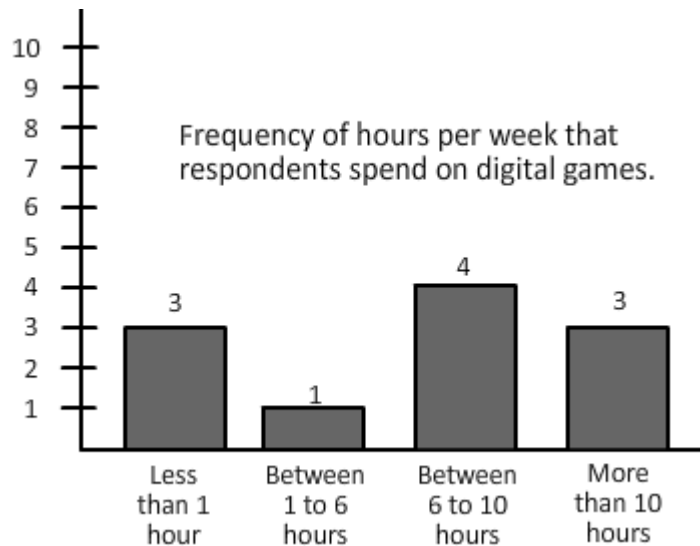


**Figure 5.1.** Top view screenshot during gameplay in Level 1.

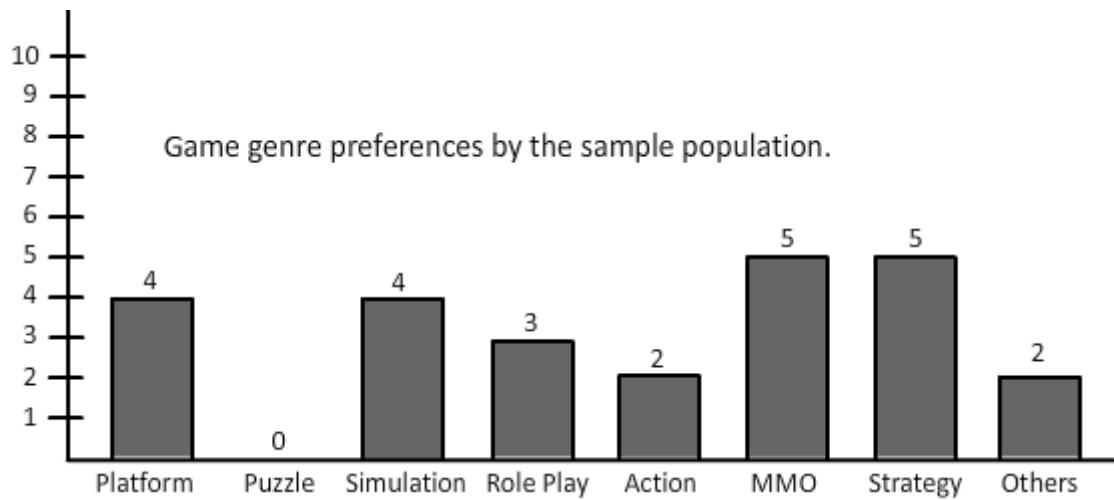
The previous experience in games and virtual worlds by the sample population of this study was very important for data analyses and eventual conclusions. As showed in figures 5.2, 5.3 and 5.4, although the great majority of students had no previous experience under the SL platform, a significant percentage play games in general and game genre preferences by the sample population were broader.



**Figure 5.2.** Student's previous experience in Second Life platform before the game.



**Figure 5.3.** Frequency of hours per week that respondents spend on digital games.

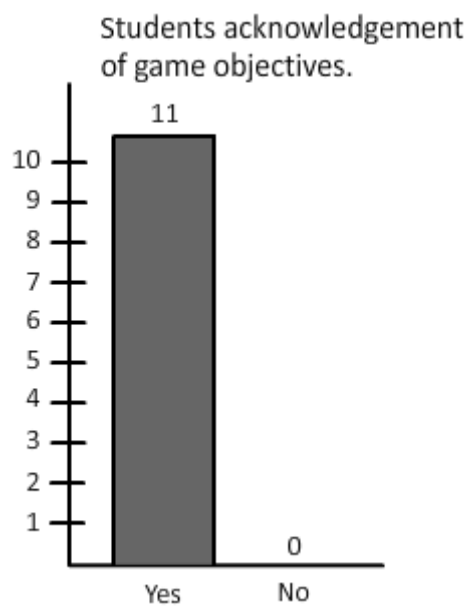


**Figure 5.4.** Game genre preferences by the sample population (in percentage).

The first process involved the student's adaptation to the Second Life virtual world, before starting the game itself. Although students had low overall experience in Second Life platform (Fig. 5.2), the observed platform interaction and student feedback provided the idea that, in general, this adaptation was easy and fast.

The following step consisted in distributing the students in two teams (Alpha and Beta) and that process, under the pre-established method, consisted in the use of Second Life Groups. This approach revealed to be lengthy, resulting in about thirty minutes to setting all up. Probably a better solution would evolve the use of wearable objects, thus reducing the team setting up duration and eventually differentiate better the members of each team during gameplay.

Before the games begin, the students were able to practice and understand the games rules and objectives in the training area. This stage occurred normally and students didn't present major questions or problems at this point. The inquiry proved that fact, as student feedback indicated a good understanding of rules and game objectives (Fig. 5.5).

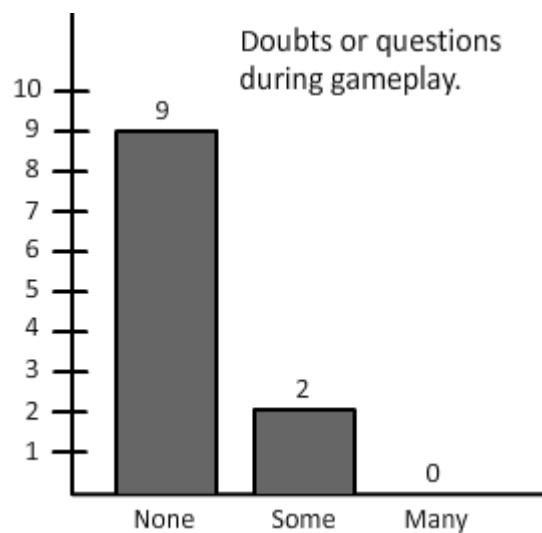


**Figure 5.5.** Student's acknowledgement of game objectives.

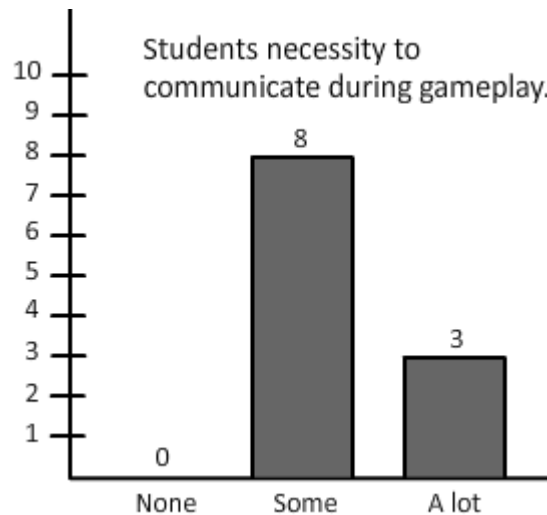
During the course of the game some observations were noticed:

- As expected, there were clearly initial attempts of disrupting the opposing team game. The foreseen solution of expelling players from the opposing team area worked and players give up those strategies. Although in terms of game design this could be a possible option for a team strategy, we believed that operations of that nature could eventually ruin the in-game experience, preventing an already hard objective of being ever accomplished and therefore make the majority of the team efforts useless and frustrating.
- There was observed teamwork and communication based on the recorded data by mediators which showed that occurred 109 attempts of communication between teammates during the first two levels and 54 were registered by students themselves on the third level (as a parallel activity).
- Only 8 occurrences of communication between opponent team members were registered.
- External signs of fun and frustration were detected, where in total 90 occurrences of fun and 50 of frustration were recorded. In this case, 88% of the frustration signs happened in level 3 (Jumping Platforms Level).
- Only a single student didn't reach the third level finish line and there were no objective attempts by their teammates to aid in somehow. All other students that had finished the course were distracted or doing other activities.

Although the students understood the game objectives, a few questions emerged during gameplay (Fig. 5.6). Good indicators were observed in the inquiry as it shows that students felt an overall need to communicate during the game (Fig. 5.7), reinforcing the notion that these game design approaches could promote communication between players.

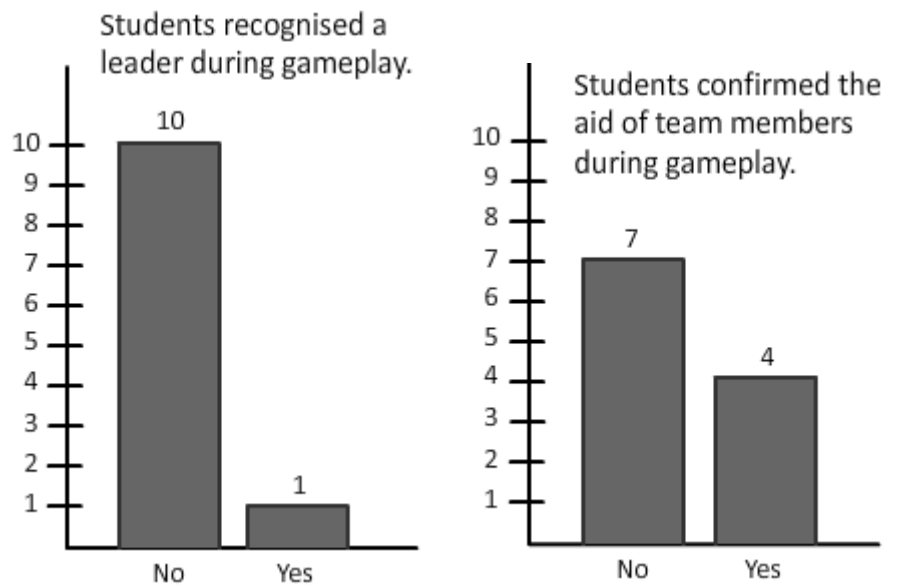


**Figure 5.6.** Doubts or questions during gameplay.



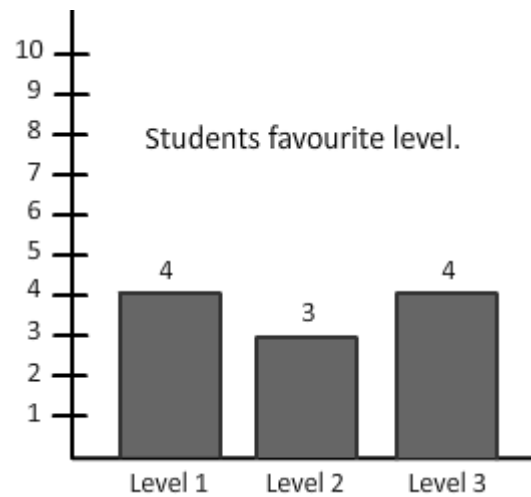
**Figure 5.7.** Students necessity to communicate during gameplay.

On the other hand, the majority of students didn't acknowledge the help of teammates (Fig 5.8) and only a single student recognised the existence of team leader during the game (and was incapable of identify him afterwards).



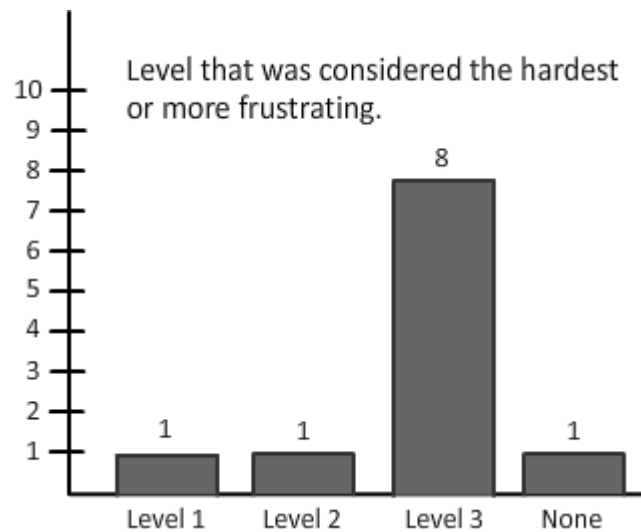
**Figure 5.8.** Students reports on leader recognition and team support.

Although the preference in terms of level was well distributed (Fig. 5.9), the third level (Jumping Platforms) was considered by the vast majority as the hardest or as the most frustrating (Fig. 5.10). As we ponder a little more on this matter, some factors may have contributed to this result. First of all, the end result of the level was far away from the initial concept and the objective was more centred in the skill of each individual than in the efforts of teamwork. Besides that, there was a need for more precision in the jumps, and the used engine may not offer an adequate performance in that interaction process.



**Figure 5.9.** Students reports on leader recognition and team support.

During observation it has become clear that the system response to a jump was significantly affected by latency and further analysis may be needed to determine if the cause derives from network performance, processing performance, the engine predefined action or a combination of the three. Nevertheless, the third level was still the favourite for 36% of the students, demonstrating that being more difficult also could mean being more challenging and appealing.



**Figure 5.10.** Level that was considered the hardest or the most frustrating for the students.



It was also conducted a Likert type scale with concordance values between one and five. The students mean average response to the questions is showed in the Table 5.1:

**Table 5.1 – Likert results from students inquires**

<b>Questions</b>	<b>Mean Average</b>
“Did you like to play this game?”	3.73
“Were you able to play it well?”	3.69
“Was the end result satisfactory?”	3.77

With all item scores above three points, these results showed eventually that the used game level concepts could provide a pleasant experience and an acceptable playability. Notwithstanding, all scores were also below four points, indicating that a lot of work can be done to improve the user experience.

## Summary and Conclusions

The observations and inquires performed during the prototype evaluation presented invaluable information about this project results.

As expected, the previous experience of students using digital games can influence their overall performance, but as in this concept a team is composed by a large amount of elements that need to work collaboratively, this aspect could be more diluted than a more single player approach.

Disruptive actions were attempted by players in the beginning of the first level, but quickly discouraged due to measures implemented. However, only with further testing will be possible to verify the existence of unforeseen disruptive actions.

The objective of promoting communication during gameplay seems to be accomplished, but colleague recognition (either by aid or leadership) fell below in the expectations. Probably other game concept or mechanics could work better for this, but we also believe that a more adequate reward system could be of some importance to the commitment taken in the activity.

# Conclusions and Future Work

This work proposes virtual games as a helper tool for engineering students' integration at FEUP. Requirements were presented and the choice of the Second Life platform was discussed. The presented game called "FEUP Adventure" will hopefully promote student integration in an amicable virtual environment and construct real experiences in a controlled environment that can provide a starting point for other integration activities.

The proposed game mechanics appeared to be a plausible option for promoting communication between team players. This is an important factor in the project objectives, as we advocate that communication is a key element in social interaction, relationships and therefore student integration in universities.

Observations related with teamwork tactics and teammate support fell short to expectations. Probably different and rethinking strategies need to be used to improve this result. It was obvious for us that students were not very focused in the tasks ahead (even with the activity inserted in the context of a curricular unit) and players' indifference to a teammate with difficulties should be prevented. We believe that a better objective reward system could be useful to this situation and force a higher level of commitment during the game, when either to help during the game tasks or by simply offer moral support (very significant for the project objectives).

Independently of whether or not is important to promote leadership within the integration process, the proposed game design proved to be very inadequate. In this case, we believe that a game design more centred in players with different roles/abilities should be more suited for this goal, instead of having players' complete equal in their traits.

The student sample also demonstrated an easy adaptation to the platform and the game levels, but this could not be the case for students with less experience in playing games.

One immediate conclusion of developing games in virtual worlds is that is that it can be very different in comparison with a common game SDK. Independently of technical limitations and arguably distinct workflows, the option of a virtual world as a delivery platform can be a valid option, especially when the game goal and design is defined to take advantage of the persistent 3D world with user selected avatars. Although many kinds of implementation can be used, it probably offers more advantages and opportunities to serious games due to the platform concept nature, despite the identified drawbacks.

The popularity of Second Life may have contributed to be an attractive target platform for game development, but building large and complex games might end in very long development times. The overall system performance must also be taken in account. Fast paced action games could not be well suited under this platform and precise physic simulations may be very hard to implement (if not altogether impossible).

But probably some of the major Second Life limitations into game development concerns are its lack of flexibility to create and edit game assets, the use of a non-object-oriented programming language and very restricted processes of interaction for the user.

Other very common practices in game development like cinematic scenes and non-player character animation are also very limited in this pre-established environment. Therefore, NPCs tend to be static objects that open message boxes when players interact with them.

Target end-users may be actual residents, but for new users, the new virtual world might present a lengthier learning curve, as they must become familiar with the Second Life interface first. External conditions may also affect the game experience, as for example, network congestion or excess of users inside the same region might slowdown performance and provoke latency, thus degrading the quality of the experience of the user.

Second life can be a very attractive platform, from the developer perspective, to produce some types of games due to its easy of use, built-in functionalities and accessibility. These characteristics can be the main factor, particularly in serious games development, where teams are usually composed by few elements or exists short development deadlines. The most important factors that can be decisive for this kind of platform selection will depend of the goals, concept and mechanics of the game itself.

## Future Work

Several possibilities can be conducted to improve the overall result of “FEUP Adventure” in the context of “Projecto FEUP” activities. One of them, and already conducted in [Men10], is the possibility of generate procedural game content more dynamically. In this prototype game, it was implemented only three levels, but it could be composed by many more.

Other platform options could also be interesting for further analysis, inclusive other virtual environments like OpenSim due to their specific characteristics and availability.

The addition of more puzzle genre concepts could also be beneficial to these project objectives. Instead of focusing in more action oriented levels, puzzles could reinforce the areas where the actual approach fell short, especially in the peer recognition and leadership factors.



# References

- [AVI11] Avimator - Second Life Avatar Animation Editor. Available online at: <http://www.avimator.com/> (last consulted on April 2011).
- [Bet03] E. Bethke, "Game Delopment And Production" Wordware Publishing Inc., pp. 39–62, 2003.
- [CCL08] T. Couper, S. Carroll, C. Liu, T. Franklin and D. Chelber, "Using the Virtual World of Second Life to Create Educational Games for Real World Middle School Science Classrooms", 2008.
- [Cir09] M. Ciragan, "Realistic Human Motion Animation with .bvh Files", CSCI 6350 Topics in Computer Graphics, 2009.
- [DGP06] The Developers Guide to Pwning Second Life (2006). [http://nwn.blogs.com/nwn/2006/07/the\\_game\\_develo.html](http://nwn.blogs.com/nwn/2006/07/the_game_develo.html) - visited august 2010.
- [DL03] DarkLife (2003). [http://slurl.com/secondlife/Navora/23/46/22/?img=http%3A//farm1.static.flickr.com/210/550701432\\_c6df7e2569\\_m.jpg&title=DarkLife%20Navora&msg=DarkLife%20Navora](http://slurl.com/secondlife/Navora/23/46/22/?img=http%3A//farm1.static.flickr.com/210/550701432_c6df7e2569_m.jpg&title=DarkLife%20Navora&msg=DarkLife%20Navora) - visited august 2010.
- [ET04] EduTeams (2004). TPLD. Dundee, Scotland. <http://www.eduteams.com/> - visited August 2010.
- [GIA10] J. García-Zubia, J. Irurzun, I. Angulo, P. Orduña, J. Ruiz-de-Garibay, U. Hernández and M. Castro, "Developing a Second-Life-based Remote Lab over the WebLab-Deusto architecture", 2010.
- [Hea07] J. Heaton, "Introduction to Textures, Animation Audio and Sculpting in Second Life", Heaton Research Inc, 2007.
- [IB10a] N. Iuppa and T. Borst, "End-to-End Game Development" Focal Press, pp. 11, 2010.
- [IB10b] N. Iuppa and T. Borst, "End-to-End Game Development" Focal Press, pp. 225–229, 2010.
- [IB10c] N. Iuppa and T. Borst, "End-to-End Game Development" Focal Press, pp. 80–83, 2010.
- [Inm10] C. Inman, "Use of Second Life in K-12 and Higher Education: A Review of Research", [Journal of Interactive Online Learning], pp. 44-63, Spring 2010.
- [Mak10] J. Makar, Actionsript for Multiplayer Games And Virtual Worlds, New Riders, pp. 3, 2010.

- [Man02] T. Manninen, Towards Communicative, Collaborative and Constructive Multi-player Games. Computer Games and Digital Cultures Conference. Tampere, Finland.
- [Men10] D. Mendes, Modelação Expedita de Jogos Digitais para Dinamização de Grupos, FEUP, 2010.
- [MK05] T. Manninen & T. Korva, Designing Puzzles for Collaborative Gaming Experience - University of Oulu, 2005.
- [Nov05a] Jeannie Novak, Game Development Essentials, Thomson Delmar Learning, pp. 65–72, 2005.
- [Nov05b] Jeannie Novak, Game Development Essentials, Thomson Delmar Learning, pp. 83–85, 2005.
- [PAC10] <http://www.packtpub.com/article/papervision3d-external-models-part1> visited August 2010.
- [PHA05] The Pot Healer Adventure (2005). <http://slurl.com/secondlife/Numbakulla/214/17/2> - visited august 2010.
- [REV05] Revolution – Colonial Williamsburg Online (2005), <http://www.educationarcade.org/node/357> - visited August 2010
- [SHS05] D. Shaffer, R. Halverson, K. Squire, Video Games and the Future of Learning. WCER Working Paper N°. 2005-4, University of Wisconsin, 2005.
- [SM11] Sculpty Maker. Available online at: <http://www.shiny-life.com/sculpty-maker/> (last consulted on April 2011)