FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

FEUP

Modelação Expedita de Jogos Digitais para Dinamização de Grupos

David José Miranda Mendes

Mestrado Integrado em Engenharia Informática e Computação

Orientador: António Coelho (Professor Auxiliar)

Co-orientador: Armando Sousa (Professor Auxiliar)

17 de Junho de 2011

Modelação Expedita de Jogos Digitais para Dinamização de Grupos

David José Miranda Mendes

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo júri:

Presidente: Rosaldo Rossetti (Professor Auxiliar da FEUP)
Vogal Externo: Luis Magalhães (Professor Auxiliar da UTAD)
Orientador: António Coelho (Professor Auxiliar da FEUP)

---

12 de Julho de 2011

Resumo

A integração de novas pessoas em grupos de trabalho ou em novos ambientes implica sempre uma adequação a novos processos e comportamentos, e nem sempre é fácil adquirir a confiança e o nível de comunicação necessários para a dinamização e homogenização do grupo. Isto acontece nas mais diversas situações, seja na composição de um grupo de trabalho de um projecto ou de uma empresa ou na integração de estudantes num novo nível da sua vida académica.

Tendo em conta o grande poder de motivação dos jogos de computador, juntamente com a crescente utilização dos mesmos na educação e na aprendizagem, é possível recorrer a um jogo sério de forma a potenciar a dinamização de grupos, através de mecânicas de jogo que reforcem a cooperação e coordenação entre os elementos dos grupos. Para tal, os jogadores devem poder interagir através da sua personagem com o ambiente virtual e com os restantes jogadores.

Da utilização de um mundo virtual surge um problema que diz respeito à limitação de espaço existente no ambiente virtual, ou seja, a extensão de terreno que se pode utilizar não é ilimitada e é necessário saber rentabilizar o espaço disponível. Assim, esta dissertação tem como objectivo desenvolver um mecanismo que permita criar rapidamente um nível virtual numa área de jogo limitada. Desta forma é possível reutilizar a mesma área de jogo para a construção alternada de níveis, através de um processo que permite rapidamente ter um novo nível disponível para ser utilizado quando se desejar.

Da análise das ferramentas disponíveis para construção desta plataforma, a utilização de um ambiente virtual multi-utilizador surge como a melhor alternativa, já que é possível dispôr de um ambiente virtual 3D pronto a utilizar, beneficiando das vantagens que apresenta a nível do controlo do espaço virtual, dos objectos e dos utilizadores. Assim, nesta dissertação foi desenvolvida uma plataforma de modelação expedita para os ambientes virtuais Second Life e OpenSimulator, que permite a construção de níveis de um jogo sério que reforça e melhora a dinamização de grupos, motivando os jogadores a comunicar, cooperar e competir entre si.

A metodologia para o desenvolvimento do mecanismo de modelação expedita passa pela modelação e definição das propriedades de cada objecto virtual e pela especificação prévia do *layout* a utilizar em cada nível de jogo.

No protótipo construído foram implementados dois níveis de jogo cuja mecânica requer que duas equipas compitam para atingirem os seus objectivos, através de estratégias de cooperação entre os seu elementos.

Abstract

The integration of new people in working groups or new environments always implies an adaptation to new processes and behaviors, and is not always easy to reach the confidence and the level of communication necessary to achieve a more dynamic and homogeneous group. This happens in different situations, whether the composition of a working group in a project or in a company or the integration of newcomers students into a new level of their academic life.

Considering the great power of motivation in computer games, along with the growing use of them in education and learning, it is possible to use a serious game in order to enhance the group dynamics through game mechanics that strengthens cooperation and coordination between groups elements. For such, players must be able to interact through their character with the virtual environment and with other players.

From the use of a virtual world occurs a problem that concerns with the limitation of existing space in the virtual environment, ie, the extent of land that can be used is not unlimited and it is necessary to know how to monetize the available space. Thus, this work aims to develop a mechanism to quickly create a virtual level in a limited play area. This way it is possible to reuse a play area for the alternating construction of levels through a process that allows to quickly have a new level available for use whenever is necessary.

After the analysis of the tools available to build this platform, the use of a multi-user virtual environment emerges as the best alternative, since it is possible to have a 3D virtual environment ready to use, benefiting from the advantages they offer in terms of controlling of virtual objects and users. Thus, in this dissertation was developed a expeditious modeling platform for Second Life and OpenSimulator virtual environments, which allows the construction of a serious game levels that strengthens and improves groups dynamics, motivating the players to communicate, cooperate and compete with each other.

The methodology for the development of the expeditious modeling mechanism involves the modeling and defining the properties of each virtual object and the prior specification of the layout to use in each game level.

In the constructed prototype were implemented two game levels whose mechanics requires that two teams compete with each other to achieve their objectives through cooperation strategies among their elements.

Agradecimentos

Todo o trabalho desenvolvido neste dissertação não é fruto apenas do meu esforço, mas conta também com a participação de várias pessoas que, directa ou indirectamente, contribuíram para que fosse possível a sua realização.

Gostaria antes de mais de agradecer aos meus orientadores, o Professor António Fernando Vasconcelos Cunha Castro Coelho e o Professor Armando Jorge Miranda de Sousa, pelo seu acompanhamento contínuo e pela sua orientação ao longo de todo o trabalho. Os seus ensinamentos, conselhos e opiniões tornaram possível a realização desta dissertação, ajudando-me a definir o rumo do trabalho e a tomar as decisões mais importantes. Também a dedicação e paixão com que encararam este projecto serviram de motivação para continuar a trabalhar e tentar continuamente melhorar o resultado final.

Agradeço também ao André Cruz, cujo trabalho serviu de motivação para o desenvolvimento desta dissertação. A sua disponibilidade, material fornecido, conselhos e ensinamentos foram fundamentais para que fosse possível a realização deste projecto.

Queria também agradecer à direcção da Faculdade de Engenharia da Universidade do Porto e ao departamento de Engenharia Informática e Computação por toda a disponibilidade e competência que os caracteriza. A disponibilização de um espaço de trabalho no laboratório de Computação Gráfica permitiu-me vários momentos de discussão e aprendizagem, que me ajudaram a realizar este trabalho.

Todo o meu percurso académico não seria possível sem o apoio, carinho e compreensão disponibilizados pela minha mãe, a quem deixo um especial pedido de agradecimento. Agradeço também a toda a minha família, que esteve sempre presente na minha vida, nos bons e nos maus momentos, e que permitiu que eu me tornasse na pessoa que sou hoje.

Por fim não posso deixar de agradecer a todos os meus amigos e colegas. Por todos os momentos de estudo, de trabalho e de troca de ideias, e sobretudo pelos momentos de alegria, convívio, e descontração vividos, que ajudaram a que eu conseguisse ultrapassar os momentos mais difíceis. Um grande obrigado por todo o apoio, compreensão e disponibilidade.

David Mendes

Conteúdo

Lista de Figuras

Lista de Tabelas

Abreviaturas e Símbolos

3D	Tridimensional
FEUP	Faculdade de Engenharia da Universidade do Porto
GBL	Game-Based Learning
GPC	Geração Procedimental de Conteúdos
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
LSL	Linden Scripting Language
MMOG	Massive Multiplayer Online Game
MUVE	Multi-User Virtual Environment
NMC	New Media Consortium
NPC	Non-Player Character
OOP	Object-oriented Programming
SDK	Software Development Kit
SG	Serious Game(s)
UDK	Unreal Development Kit
URL	Uniform Resource Locator
XNA	XNA's Not Acronymed

Capítulo 1

Introdução

Os jogos digitais têm um elevado poder de motivação e de imersão que potencia a sua utilização no ensino e na aprendizagem. Os seus utilizadores são cativados e seduzidos através da representação de realidades ou da encarnação de fantasias, que permitem aos jogadores fazer nos jogos o que não é possível ou permitido na vida real [MSS04]. Através do entretenimento e do desafio, os jogadores são motivados a ultrapassar barreiras e atingir objectivos, sendo estimulados a estabelecer comunicação e a cooperar com os outros jogadores. Assim, a sua utilização na dinamização de grupos permite uma grande capacidade de interacção e correspondente facilidade na integração dos elementos do grupo e na sua comunicação.

Este capítulo introdutório apresenta o enquadramento e a motivação desta dissertação e define o problema abordado, os objectivos e os resultados esperados deste trabalho.

1.1 Enquadramento e Motivação

A integração de novas pessoas em grupos de trabalho ou em novos ambientes implica sempre uma adequação a novos processos e comportamentos, e nem sempre é fácil adquirir a confiança e o nível de comunicação necessários para a dinamização e homogeneização do grupo. Isto acontece nas mais diversas situações, seja na composição de um grupo de trabalho de um projecto ou de uma empresa ou na integração de estudantes num novo nível da sua vida académica.

Um caso particular deste problema diz respeito aos novos estudantes do Ensino Superior, que enfrentam alguma dificuldade inicial em se integrarem num novo sistema de ensino. Por esta razão a FEUP desenvolveu uma unidade curricular para potenciar este processo - o Projecto FEUP. Desde o ano lectivo de 2004/2005 que esta unidade curricular tem recebido os novos alunos da FEUP e integrando-os na vida académica, tendo como

principais objectivos dar a conhecer os principais serviços da faculdade, dar formação inicial nas áreas conhecidas como *Soft Skills*, alertando para a sua importância ao longo da carreira em Engenharia e discutir cientificamente um tema, de forma a resolver um projecto de dificuldade limitada [FEU10]. Desta forma é possível reunir, em cada ano, os novos estudante do Ensino Superior e estimular a interacção entre eles, garantindo que ocorra socialização e aprendizagem no primeiro contacto com a faculdade.

No entanto, é necessário considerar novas formas de promover a integração, através da utilização de um sistema que seja, simultaneamente, motivador e familiar para os estudantes, bem como ter em conta a dimensão do problema, já que entram todos os anos cerca de 900 novos alunos para a FEUP.

Neste sentido, a utilização de jogos digitais surge como uma boa aposta para ajudar na integração de novos elementos e na dinamização de grupos em geral, já que é possível garantir comunicação, cooperação e entreajuda, com a possibilidade de envolver todos os intervenientes neste processo. Com este objectivo existe o trabalho anterior em desenvolvimento denominado FEUP Adventure, que consiste na elaboração de 3 níveis virtuais no Second Life. Foi utilizado na unidade curricular Projecto FEUP e visa analisar a potencialidade da utilização de um jogo digital na integração dos novos estudantes do ensino superior. Este projecto é analisado mais pormenorizadamente na secção 3.2.1.4.

Esta situação particular surge como enquadramento para o desenvolvimento deste projecto que vai além do âmbito do projecto FEUP mas que tem nele o caso de estudo para o seu desenvolvimento.

1.2 Descrição do Problema

A utilização de jogos para o ensino e aprendizagem corresponde a uma abordagem que permite cativar os seus utilizadores de forma eficaz e produtiva, já que os jogadores adquirem os conhecimentos através do entretenimento e do desafio.

Desta forma é possível conceber um jogo digital que facilite e promova a dinamização de grupos, através de estratégias de cooperação e competição, num mundo virtual em que os jogadores interajam através da sua personagem. Para tal, é fundamental recorrer a mecânicas simples de interacção entre os jogadores e os elementos de jogo, de forma a que seja valorizada a cooperação entre os jogadores.

É necessário garantir que o jogo seja de aprendizagem fácil, para que a interacção seja rápida e eficaz,de forma a não comprometer a experiência de cooperação e de competição. É também importante considerar a sua expansibilidade, de forma a ser possível a criação e suporte de novos níveis de jogo.

Além disso, a criação e utilização de um espaço virtual 3D está limitada à extensão de terreno disponível, e é necessário pensar em métodos para rentabilizar o espaço disponível. Os diferentes ambientes virtuais dos jogos digitais são habitualmente concebidos

individualmente sem recorrer à reutilização do espaço para reproduzir diferentes ambientes. Isto requer, por um lado a utilização de uma grande extensão de terreno virtual para a construção de todo os cenários pretendidos, e por outro lado que as personagens do jogo se desloquem no ambiente virtual para terem acesso aos diferentes níveis de jogo.

1.3 Objectivos

O objectivo principal deste projecto consiste em desenvolver um mecanismo de modelação expedita de níveis de um jogo sério que possibilite e promova a dinamização de grupos. O jogo deve promover estratégias de cooperação e de competição entre os diversos jogadores, nos diversos âmbitos em que estejam inseridos.

Para tal, é necessário efectuar uma pesquisa bibliográfica e investigar o estado da arte no que diz respeito à criação de jogos digitais, tendo em conta jogos sérios utilizados na educação e também relativamente a ambientes virtuais multi-utilizadores.

Os níveis de jogo devem ser desenvolvidos como um mundo virtual, em que cada jogador terá uma personagem que pode controlar e com ela interagir com os objectos e com os outros jogadores.

Deve permitir a construção de níveis simples, que requeiram a envolvência de todo o grupo, no sentido de cooperarem e se coordenarem de forma a atingir os objectivos. Estes objectivos podem ir desde a deslocação de peças no cenário, até à resolução de puzzles e enigmas.

Tendo a unidade curricular Projecto FEUP e o processo de integração dos estudantes no Ensino Superior como exemplo, deve-se desenvolver níveis protótipo que tenham em consideração a distribuição dos estudantes nas sessões de trabalho, de forma a ser possível providenciar facilmente as diferentes experiências de cooperação e competição de cada nível.

É também objectivo deste projecto agilizar o processo de criação dos níveis, através da modelação expedita dos elementos virtuais a inserir no espaço 3D, que permita que a mesma área de jogo virtual seja reutilizada para construção de vários níveis, ao mesmo tempo que torna esta construção rápida e fácil, de forma a estar acessível para ser utilizada rapidamente.

1.4 Resultados Esperados

É esperado que desta dissertação resulte uma plataforma para a construção de um jogo sério que ajude a promover o processo de dinamização de grupos de trabalho, através da interacção entre os seus constituintes com o jogo e com os outros participantes. Assim, é esperado que seja desenvolvido um mecanismo para a construção de níveis de forma

rápida e eficaz, com a definição dos objectos presentes e das suas propriedades e se desenvolva a interacção entre os diversos elementos do jogo.

Deve permitir que vários participantes joguem simultaneamente, fomentando entre eles o espírito de cooperação e competição. Para tal, é necessário seleccionar uma tecnologia e conceber uma arquitectura para o desenvolvimento do jogo.

O resultado final esperado consiste num protótipo que permita verificar o funcionamento da criação de níveis através da modelação expedita dos seus elementos, que esteja pronto a utilizar com níveis de teste em que os jogadores são alocados em equipas e podem jogar.

1.5 Metodologia

Para o desenvolvimento deste projecto é necessário realizar uma pesquisa bibliográfica acerca do estado da arte, no que diz respeito ao desenvolvimento de jogos digitais, jogos sérios, jogos para a educação (*Edutainment*, *Game-based Learning*), bem como acerca de ambientes virtuais, *Multi-users Virtual Environments* e geração procedimental de conteúdo.

É também necessário perceber, antecipadamente, quais as preferências, qual o nível de familiaridade com jogos digitais e qual a aceitação que o jogo pode ter. O desenvolvimento do protótipo deve ser realizado em fases incrementais, com níveis intermédios de teste, de forma a ser possível a realização de experiências de utilização. Desta forma é possível avaliar a interacção e a aceitação do jogo e estudar as alterações e evoluções necessárias.

1.6 Estrutura do Documento

Este documento encontra-se organizado em 7 capítulos, complementados com referências e anexos técnicos. O primeiro e presente capítulo consiste num capítulo introdutório, onde é apresentado o enquadramento e motivação desta dissertação, os objectivos e os resultados que se pretendem atingir com este projecto.

O segundo capítulo apresenta uma revisão ao estado da arte relativo ao âmbito desta dissertação. Assim, é abordada a criação de jogos digitais, sendo feita uma revisão tecnológica às ferramentas existentes, bem como aos jogos sérios, analisando os conceitos de *Edutainment* e *Game-based Learning*. Também são abordados os ambientes virtuais multi-utilizadores e a geração procedimental de conteúdos.

No capítulo 3 são apresentados trabalho e projectos relacionados com as áreas de estudo abordadas no capítulo 2 e cuja análise é importante para a realização desta dissertação.

O quarto capítulo apresenta a metodologia utilizada para o desenvolvimento deste projecto, referindo os processos teóricos para o desenvolvimento de um mecanismo de modelação expedita de níveis de jogo num ambiente virtual multi-utilizador. Assim, é apresentada a metodologia de utilização do espaço virtual, a mecânica de jogo e o processo de construção dos níveis de jogo.

No quinto capítulo é apresentado e detalhado o trabalho necessário a nível de implementação, com especificação das decisões e implementações técnicas efectuadas. Nele é apresentada a arquitectura da solução e são analisadas as ferramentas de desenvolvimento utilizadas. É também especificado o processo de criação dos níveis de jogo a um nível mais técnico, relativamente ao efectuado no capítulo 4, bem como são referidos os métodos de comunicação utilizados entre os vários elementos de jogo.

O capítulo 6 descreve os resultados obtidos na construção e utilização do protótipo desenvolvido. Para tal nele se apresenta o processo de criação de um nível e é feita uma avaliação da performance da plataforma. São também analisados os resultados obtidos na reprodução de um caso de estudo e é feita uma análise às ferramentas utilizadas para utilização do ambiente virtual.

No sétimo e último capítulo é feito um resumo geral do trabalho realizado nesta dissertação e apresenta uma conclusão ao resultado obtido neste projecto, com verificação do cumprimento e do nível de satisfação dos objectivos. São também referidas perspectivas de trabalho futuro que visam melhorar o comportamento e a aplicabilidade deste trabalho.

Capítulo 2

Estado da Arte

Neste capítulo é descrito o estado da arte, de forma a perceber qual o domínio em que este projecto se encontra inserido. Assim, são abordados os jogos de computador, efectuando uma revisão tecnológica às ferramentas existentes, os jogos sérios (*serious games*) destinados à educação (*edutainment* e *game-based learning*) e os ambientes virtuais multi-utilizadores, bem como a utilização de geração procedimental de conteúdos.

2.1 Jogos de Computador

Uma das razões para a utilização de jogos de computador no ensino deve-se ao facto de estes cativarem os utilizadores. Tal acontece porque os jogos são sedutores, utilizando tecnologia para representar realidades ou encarnar fantasias. Eles também motivam através do entretenimento e do desafio, já que são jogados para se ganhar ou atingir um objectivo [MSS04].

Além das escolas que utilizam jogos no seu processo de ensino, muitas empresas fazem pleno uso de jogos digitais para questões relacionadas com *marketing*, recrutamento e treino, através de *websites*, ambientes virtuais ou telemóveis, entre outros [Raf08].

Assim, é necessário apurar o que é necessário para o desenvolvimento de um bom jogo, no sentido de cativar e manter os jogadores motivados [Tee10]:

- **Desafio contínuo:** um bom jogo apresenta continuamente desafios, que levam sempre a novos desafios, de forma a manter os jogadores "agarrados" e com vontade de querer continuar a jogar.
- **Guião interessante:** a história não é essencial em todo o tipo de jogos, especialmente quando os jogadores competem uns com os outros - nestes casos o entusiasmo da competição é normalmente suficiente para os cativar. No entanto, um bom guião pode animar a competição ainda mais.

- **Flexibilidade:** é necessário assegurar que existem várias formas diferentes de realizar cada objectivo. Deve-se deixar que cada jogador (ou equipa) elabore a sua própria estratégia para atingir os objectivos, mantendo os desafios e alcançando os conhecimentos pretendidos.
- **Recompensas úteis imediatas:** em vez de apenas aproximar para a vitória, pode-se recompensar os jogadores bem-sucedidos com novos recursos, informação adicional ou até mesmo uma nova tarefa para explorar. Tais recompensas são motivadoras, já que é objectivo do jogo não apenas vencer, mas continuar a jogar.
- **Combinar diversão e realismo:** demasiado realismo pode ser entediante, pelo que se deve saber incorporar premissas geralmente incorrectas ou recompensar comportamentos irrealistas, que não seriam habitualmente possíveis no mundo real.

Um passo também fundamental na realização de um jogo digital tem a ver com a análise de ferramentas existentes para a construção do protótipo. Devido à enorme variedade de plataformas e motores de jogo disponíveis, a escolha de qual utilizar deve ser baseada nos objectivos que se pretendem atingir e nos requisitos técnicos necessários para o contexto e utilização pretendidos [PDDFP10]. Devido ao carácter *multiplayer* e à necessidade de utilização em rede, deve-se ter em conta *game engines* que permitam tais características. Simultaneamente, deve ser possível obter informação e documentação facilmente.

Existem várias opções disponíveis, mas tendo em conta a sua utilização crescente na actualidade e as avaliações existentes [Fre09], foram seleccionados os motores Unity3d, UDK, Torque3d e CryEngine 3, bem como a plataforma Second Life. Apesar de esta última opção não ser um jogo, é um ambiente virtual 3D multi-utilizador que pode ser utilizada como plataforma para desenvolvimento de jogos. Embora apresente algumas limitações neste sentido (como a utilização de uma linguagem de *scripting* não orientada a objectos), possui um sólido sistema em rede e um *multi-plataform web client*, que permitem uma grande flexibilidade e estabilidade no contexto de simulação virtual e de jogos *multiplayer* [CSC10]. Também é possível utilizar a plataforma *opensource* OpenSimulator que corresponde a uma hospedagem de mundos virtuais 3D similar ao Second Life.

Relativamente aos motores de jogo referidos, todos eles se encontram na lista das 10 melhores *game engines* de [Fre09]. UDK é um motor de jogos poderoso e flexível, com resultados satisfatórios a nível de características, estabilidade e documentação [Dev10b], sendo, no entanto, recomendado para utilizadores experientes. Unity 3d é um motor de jogo flexível, com uma reduzida curva de aprendizagem, que conta com uma vasta comunidade de utilizadores e boa colecção de documentação de apoio ao desenvolvimento [Dev10a]. O motor Torque 3d é caracterizado pela sua poderosa arquitectura Cliente/Servidor, tendo já sido premiado pela sua rede *multiplayer*. Apesar de possuir uma licença

paga, é fornecido todo o código fonte C++, de forma a ser possível realizar qualquer adição necessária ao jogo [Dev07]. CryEngine 3 permite desenvolvimento e edição em tempo real e dá acesso a todo o código fonte e documentação. A licença Educacional é disponibilizada gratuitamente a educadores, para uso não comercial em cursos ou projectos de investigação [Gmb11].

Tabela 2.1: Comparação entre *game engines* [PDDFP10] [Dev04] [Dev07] [Dev10b] [Dev10a] [Dev08].

		UDK	Unity3D	Torque	CryEngine 3	Second Life
Fidelidade audiovisual	*Rendering*	*Stereo Rendering, Deferred Shading, Render-to-Texture, Raytracing, Raycasting, Fonts,* GUI	*Render-to-Texture, Fonts*	*Fixed-function, Render-to-Texture, Fonts,* GUI	*Fixed-function, Render-to-Texture*	*Fonts,* GUI
	Animações	*Inverse Kinematics, Forward Kinematics, Keyframe Animation, Skeletal Animation, Facial Animation, Animation Blending, Morphing*	*Keyframe Animation*	*Inverse Kinematics, Skeletal Animation, Animation Blending*	*Inverse Kinematics, Skeletal Animation, Animation Blending*	*Keyframe Animation*
	Som	2D, 3D, *Streaming Sound*	2D, 3D, *Streaming Sound*	2D, 3D, *Streaming Sound*	2D, 3D	3D, *Streaming Sound*
Funcionalidades	*Scripting*	UnrealScript	JavaScript, C#.NET	*C++-based*	C++, LUA-*based*	LSL
	Técnicas de Suporte de IA	*Path Finding, Decision Making, Finite State Machines, Scripted, Neural Networks*	*Collision Detection, Path Finding, Decision Making, Scripts*	*Finite State Machines, Scripts*	*Path Finding, Decision Making, Scripts*	*Scripted:* LSL
	Física	*Basic Physics, Collision Detection, Rigid Body, Vehicle Physics*	*Basic Physics, Collision Detection, Rigid Body, Vehicle Physics*	*Basic Physics, Collision Detection, Rigid Body, Vehicle Physics*	*Basic Physics, Collision Detection, Rigid Body, Vehicle Physics*	*Basic Physics, Collision Detection, Rigid Body, Vehicle Physics*
Agregação	Importação /Exportação	3ds max, maya	3ds max, maya	3ds max, maya	3ds max, maya	*Sculpt Textures*
Acessibilidade	Curva de aprendizagem	Elevada	Baixa	Média	Média	Baixa
	Documentação e tutoriais	Sim	Sim	Sim	Sim	Sim
	Licenciamento	Grátis (para uso não comercial)	Grátis (sem ser versão pro)	$99	Grátis (licença educacional)	Grátis (com custos na manutenção de regiões virtuais)
Rede		Cliente-Servidor, P2P, *Master Server*	Cliente-Servidor	Cliente-Servidor	Cliente-Servidor	*Master Server: Simulators (sims)*
Multiplataformas		Windows, iOS	Windows, MacOS, Nintendo Wii, Browser-based, iOS	Windows, Linux, MacOS	Windows, Xbox, Paystation, GameCube	Windows, Linux, MacOS

Na tabela 2.1 é feita uma breve comparação entre as *game engines* referidas, tendo em conta parâmetros fundamentais na escolha final. Apesar de o Second Life não corresponder a uma *game engine* comercial, as suas especificações são importantes para perceber

as funcionalidades disponíveis na utilização do ambiente virtual.

2.2 Massively Multiplayer Online Game

Massively Multiplayer Online Games (MMOG) são um género de jogos de computador *online* cujo principal foco consiste na criação de um mundo virtual que vários jogadores possam explorar em simultâneo, interagindo entre si e com o ambiente, competindo e cooperando em larga escala. A utilização deste tipo de jogos tem crescido extraordinariamente desde o início do século XXI, com mais de 11 milhões de utilizadores com subscrições pagas em todo o mundo [ZKD08].

O termo MMOG é utilizado genericamente para caracterizar os vários géneros de jogos online dirigidos para vários utilizadores. Assim, existem diversos tipos de *massively multiplayer online games*:

- *MMO role-playing game*
- *MMO first-person shooter*
- *MMO real-time strategy*
- *MMO sports game*
- *MMO racing*
- *MMO social game*

Estes tipos de jogos caracterizam-se por neles existir um mundo persistente, onde o tempo de jogo é contínuo, haja ou não interacção. Também se distinguem pela sua jogabilidade direccionada para a vertente *multiplayer*, dando maior relevância às actividades em grupo, pelo que os jogadores não conseguem terminar o jogo, no sentido típico dos jogos *singleplayer* [Wik11].

2.3 Serious Games

Serious Games é uma categoria de jogos cujo objectivo primário não é entreter o utilizador, mas sim educar, sensibilizar ou dar a conhecer algo. São desenvolvidos com a intenção de melhorar algum aspecto da aprendizagem, sendo utilizados na formação de serviços de emergência, em formação militar, em educação corporativa, nos cuidados de saúde, e em muitos outros sectores da sociedade. São também encontrados em todos os níveis da educação, em todos os tipos de escolas e universidades [Der07].

Conceitos como *Edutainment* e *Game-based Learning* estão fortemente relacionados com os *Serious Games*, vistos numa vertente orientada para a educação.

2.3.1 Edutainment

Edutainment, que pode ser traduzido como entretenimento educacional, é uma forma de entretenimento que tem como objectivo, não só divertir, mas também educar. O *edutainment* tipicamente tenta instruir ou sensibilizar a sua audiência através de lições transmitidas sobre várias formas de entretenimento (programas de televisão, filmes, músicas, etc.).

Neste sentido, é óbvia a utilização de jogos digitais como forma de *edutainment*, já que estes são uma forma natural de entretenimento, onde é possível desenvolver conhecimentos e proficiências. Jogos de computador educativos combinam educação e entretenimento, e podem ser definidos como um meio electrónico com todas as características de um ambiente de jogo, que tem objectivos educacionais direccionados especificamente a determinados grupos de alunos.

2.3.2 Game-based Learning

Também no âmbito de entretenimento como educação surge o *Game-based learning* (GBL), que é um ramo dos *Serious Games* que lida com aplicações que têm como objectivo obter resultados de aprendizagem nos utilizadores. Geralmente são desenhados de forma a equilibrar o tema e os objectivos do jogo com a jogabilidade e a capacidade do jogador em reter e aplicar esses objectivos.

Há vários elementos que definem uma actividade como sendo um GBL [Tee08]:

- Competição: através da manutenção de pontuações e/ou do alcance da condição de vitória. Não implica que os jogadores compitam uns contra os outros, sendo que em inúmeros jogos os jogadores trabalham em equipa para superar os obstáculos ou adversários.
- Empenho: que surge na vontade de não parar até conseguir atingir todos os objectivos de um jogo.
- Recompensas imediatas: já que os jogadores recebem pontos ou prémios pelas vitórias ou algum tipo de *feedback* descritivo, conforme as metas vão sendo cumpridas.

Estas características são em tudo semelhantes às de um plano de uma boa lição:

- Realização: cada aula ou actividade é baseada em objectivos que os estudantes devem alcançar, adquirindo novos conhecimentos e proficiências. Os estudantes devem ser desafiados a por em teste as suas capacidades.
- Motivação: os temas abordados são fundamentalmente interessantes, motivando os alunos através do estabelecimento de conexões entre os conteúdos aprendidos e a resolução de problemas que podem surgir no quotidiano e nas suas vidas.

- Avaliação: Em rigor, a recompensa pelo bom desempenho na escola é a maior compreensão e novos conhecimentos. Estes são representados em termos de notas e avaliações.

Neste sentido, a aplicação de GBL neste projecto surge do facto de se pretender que os utilizadores consigam, através da utilização do jogo, assimilar e praticar os melhores processos para se integrarem no novo grupo de trabalho. Esta analogia é realizada através da cooperação com os elementos da sua equipa, na realização dos objectivos em conjunto, e da competição com equipas adversárias.

2.4 Multi-User Virtual Environments

Multi-user virtual environments (MUVEs) são ambientes virtuais *online*, destinados a ser utilizados por vários utilizadores. Todos os MUVEs permitem que múltiplos participantes em simultâneo acedam ao contexto virtual pretendido, interajam com objectos virtuais, se representem através de avatares (em alguns casos visuais e noutros baseados em texto), comuniquem com outros participantes e façam parte de experiências em que é necessário seguirem determinado modelo e orientarem-se na resolução de problemas semelhantes aos existentes em contextos do mundo real [DC07].

Com estas características, é evidente a potencialidade dos MUVEs no apoio ao ensino e à integração, e vários grupos de pesquisa têm criado e explorado ambientes virtuais e investigado a sua eficácia [CIT]. MUVEs educacionais são desenhados para conter aprendizagem baseada na pesquisa e na compreensão, onde usualmente não existe uma única forma correcta de realizar as tarefas, mas várias soluções, que, como na vida real, podem ser melhor umas em relação a outras. Desta forma, é focada a lógica do raciocínio dos utilizadores. Ao contrário de MUVEs desenhados exclusivamente para o entretenimento, utilizadores de MUVEs educacionais precisam frequentemente de reunir informação *offline* ou interagir com outros participantes de uma forma mais pessoal, de forma a realizar as tarefas através do conhecimento e da colaboração [CIT].

É possível ver alguns exemplos de *multi-user virtual environments* na secção 3.2.

2.5 Geração Procedimental de Conteúdos

A geração procedimental de conteúdos permite que o conteúdo seja criado e implantado pelo próprio programa de computador em si, através de algoritmos bem definidos e/ou conjuntos de regras [Rud09]. Isso permite que grande quantidade de conteúdo seja produzida automaticamente, gastando-se muito menos tempo e dinheiro. Por outro lado fornece uma abordagem simples de construir ambientes 3D, mesmo para pessoas sem conhecimento técnicos.

Quanto aos jogos digitais, o conteúdo pode ser considerado como o material de jogo que é gerado durante a execução do jogo, como os níveis, a inteligência artificial, sons, objectos ou personagens. Tendo em conta a geração de ambientes de jogo em geral, existem quatro abordagens principais: além da geração procedimental, há os espaços criados pelos *designers*, os criados aleatoriamente e os criados pelos utilizadores. A maioria dos jogos dependem de elementos pré-fabricados, sem o uso da geração procedimental de conteúdo [NAH+06], através de espaços 3D criados por *designers* que estão previamente definidos e que não podem ser alterados no decorrer do jogo. Este tipo de conteúdo é considerado como material *handcrafted*, já que cada objecto modelado requer atenção individual por parte do *designer*, no seu processo de modelação e na definição das suas propriedades. Assim, quanto mais material *handcrafted* for construído, maiores os custos (figura 2.1).

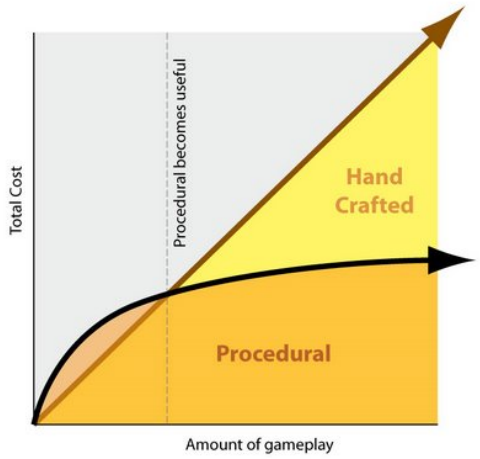


Figura 2.1: Relação entre a quantidade de conteúdo e o seu custo, de conteúdo gerado procedimentalmente e *handcrafted* [Dan07].

A utilização de geração procedimental de conteúdo tem a vantagem de, apesar do custo de arranque inicial, é possível obter uma enorme quantidade de material após a fase inicial, por pouco ou nenhum custo adicional. Também tem o benefício de permitir novas abordagens facilmente, podendo o conteúdo ser refeito, testado e envolvido num processo interactivo que permite a quem desenvolve testar novas ideias rapidamente [Dan07].

Neste projecto é realizada uma abordagem baseada na geração procedimental de conteúdos, em que os objectos 3D são criados procedimentalmente no ambiente virtual, seguindo uma especificação prévia que define a sua posição e orientação, o que permite que a modelação dos níveis seja rápida e simples.

2.6 Conclusão

Os jogos de computador têm grande potencial para serem utilizados na aprendizagem e na educação, já que os jogadores ficam imersos no ambiente oferecido pelo jogo, existindo grande envolvimento e concentração no cumprimento das tarefas e dos objectivos. Como tal, é importante assegurar que o jogo seja motivador e cativante.

Na escolha das ferramentas a utilizar é importante verificar e analisar as suas principais características, nomeadamente na escolha do motor de jogo a utilizar como base para a construção da plataforma. Todos os motores analisados são potencialmente boas opções, apresentando vantagens e desvantagens. No entanto, a utilização de um ambiente virtual 3D multi-utilizador como o Second Life ou o OpenSimulator surge como a melhor opção, já que permite tirar partido de todo o ambiente 3D já existente e de todo o sistema de comunicação cliente/servidor.

A tipologia *Massively Multiplayer Online Games* é analisada, já que tem como característica a possibilidade de exploração de um ambiente virtual por diversos jogadores em simultâneo, trabalhando e competindo em equipa com outros utilizadores.

A utilização de um *Serious Game* é aplicada no sentido de desenvolver um jogo que ajude a promover e a melhorar os processos de integração e de dinamização de grupos de utilizadores. Conceitos como *Edutainment* e *Game-based Learning* são tidos em conta na medida em que se pretende conciliar educação com entretenimento, garantindo que exista aprendizagem por parte dos utilizadores.

Também os *Multi-user Virtual Environments* potenciam o processo de aprendizagem e de cooperação e competição, na medida em que dão liberdade aos seus utilizadores para descobrir, investigar e interagir com o ambiente, ao mesmo tempo que motivam para a resolução de problemas e tarefas.

Considerando a necessidade de rapidamente providenciar um ambiente de jogo para ser utilizado com os grupos de trabalho, a modelação expedita é realizada tendo como base a geração procedimental de conteúdo, tirando partido da facilidade de geração de material 3D e da rapidez e simplicidade com que é possível criar os níveis de jogo.

Capítulo 3

Trabalhos Relacionados

Neste capítulo são apresentados trabalhos, projectos e exemplos relacionados com as áreas de estudo apresentadas no capítulo anterior.

3.1 Serious Games

Existem inúmeros projectos relativos a *serious games* (SG), direccionados para as mais diversas áreas. Nas subsecções seguintes serão focados SG voltados para a educação e para a aprendizagem, de forma a perceber as potencialidades da sua utilização nestas áreas.

3.1.1 MindRover

MindRover foi desenvolvido pela CogniToy para ajudar os jogadores a aprender a programar. Especificamente, os jogadores codificam inteligência artificial para veículos robóticos para os ajudar a navegar por pistas de obstáculos e a superar outros desafios. A interface de programação envolve arrastar peças lógicas da mecânica dos robots e definir parâmetros para cada uma, não existindo linhas de código. O jogo encoraja uma aproximação exploratória para a aprendizagem de programação, ajudando os jogadores a ter os seus programas em execução rapidamente, de forma a poderem experimentar e interagir [KOS09].

3.1.2 Astroengineer: Moon Rover

É um projecto desenhado para ensinar estudantes do ensino secundário acerca de temas como a ciência, tecnologia, engenharia e matemática. O jogo passa-se no ano de 2040 e a humanidade criou várias bases na superfície lunar e na sua órbita, que trabalham

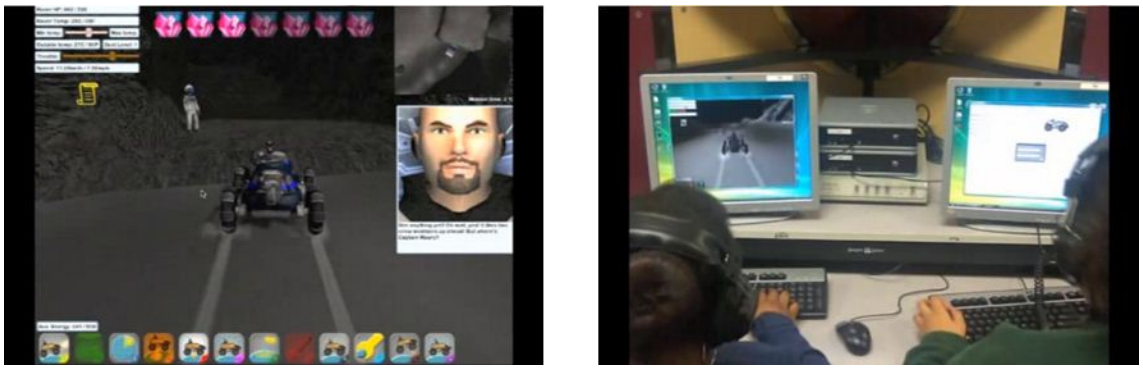


Figura 3.1: Interface e utilização de Astroengineer: Moon Rover [Wis10].

em conjunto para recolher recursos, explorar e desenvolver uma base de operações para futuras missões a outros planetas. O jogo ajuda os alunos a desenvolver e testar soluções para problemas com determinados critérios e restrições. É desenvolvido pela WisdomTools, Inc, e resulta de várias parcerias [Wis10]. É possível ver na figura 3.1 a sua interface e utilização.

3.1.3 Time Engineers

Time engineers é um jogo que ensina engenharia, ciência e matemática ao mesmo tempo que combina a diversão de um jogo de computador. Os alunos viajam no tempo para diferentes períodos, onde encontram diversos problemas de engenharias típicos, que devem ser resolvidos de forma a construir pirâmides, irrigar quintas, comandar submarinos, entre outras tarefas (figura 3.2). Desenvolvido pela Software Kids em conjunto com a Faculdade de Engenharia da Universidade de Valparaiso (Chile), foi projectado para ajudar alunos do ensino básico e secundário [Kid10].

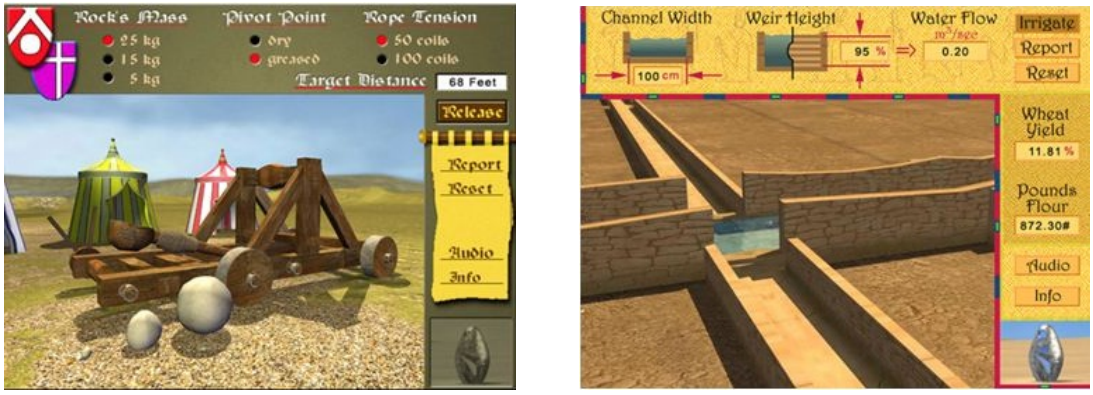


Figura 3.2: Interface de Time Engineers [Kid10].

3.1.4 Supercharged!

Supercharged! é um jogo para ensinar conceitos de alto-nível de física. O jogo permite aos utilizadores pilotar uma nave espacial num ambiente tridimensional utilizando a carga eléctrica da nave e partículas com carga, presentes no espaço (figura 3.3). Os alunos planeiam a sua trajectória ao longo dos níveis traçando as linhas do campo proveniente de cada objecto com carga. Através deste jogo os alunos desenvolvem um conhecimento mais práctico de como as partículas com cargas se comportam [DF06].

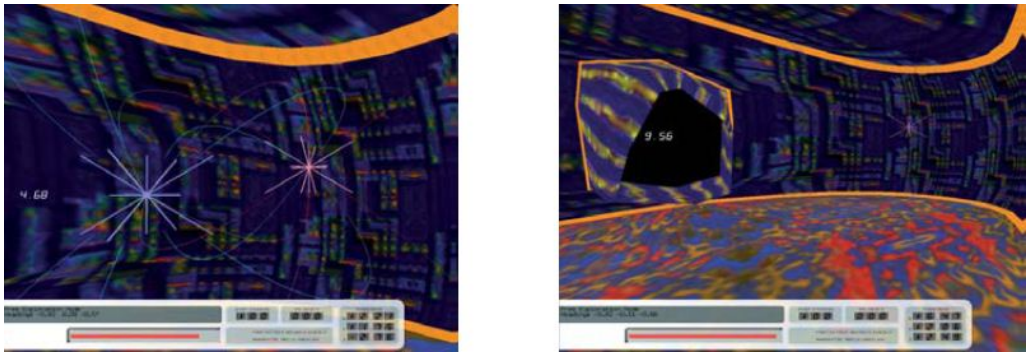


Figura 3.3: Interface de Supercharged! [DF06]

Em [SBGH04] concluiu-se que estudantes que participaram num módulo que utilizou o jogo obteve melhores resultados que os estudantes que aprenderam física através de experiências prácticas, demonstrações e visualização de simulações [DF06].

3.1.5 Palmagotchi

Palmagotchi é um jogo para telemóveis desenvolvido no MIT Scheller Teacher Education. Combina as ideias de animais de estimação virtuais (como o popular Tamagotchi) com a história evolutiva do percurso de Darwin nas ilhas Galápagos. Os jogadores mantêm famílias de aves que devem monitorizar e alimentar e ilhas de flores para cuidar.

O jogo é desenhado para ser *school-friendly*, por isso os alunos são estimulados a interagir com o jogo a cada 3 ou 4 horas, de forma a não perturbar as aulas e ao mesmo tempo criar sentido de responsabilidade, pois precisam de estar atentos de forma a manter os seus organismos vivos e saudáveis [KOS09].

Este jogo demonstra novas abordagens, comparando com os outros jogos referidos neste capítulo, relativamente à integração de jogos nas escolas, sem utilizar tempo reservado para as aulas [KOS09].

3.1.6 Racing Academy

Desenvolvido pela FutureLabs no Reino Unido, em colaboração com a Lateral Visions, o UK Higher and Further Education Joint Information Services Council e o Departa-

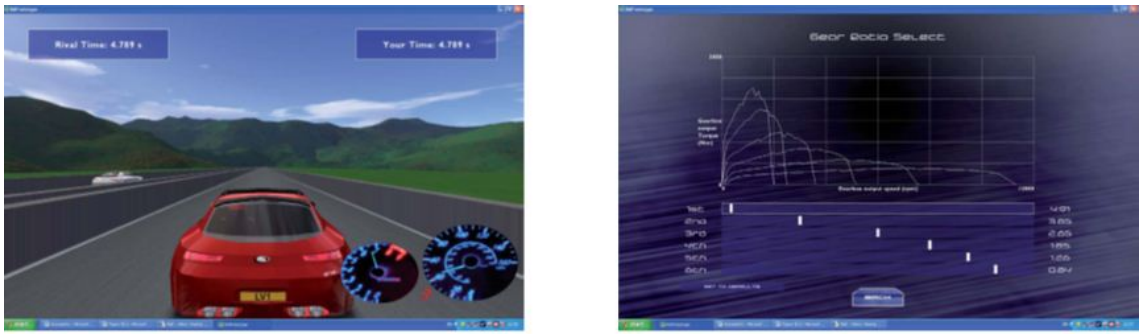


Figura 3.4: Interface de Racing Academy [DF06].

mento de Psicologia da Universidade de Bath, este jogo permite aos estudantes o acesso a modelos precisos e em tempo real de como os carros funcionam, inseridos no contexto de um jogo de corridas. Os estudantes constroem, fazem a manutenção e competem com os seus veículos e, para conseguir vencer, devem monitorizar e analizar o seu desempenho, através de variados dados e medições fornecidas (figura 3.4). É possível aos estudantes participar em sessões prácticas, onde devem tomar decisões complexas de forma colaborativa, manipulando mais de 1000 parametros acerca dos seus veículos [KOS09].

3.2 Multi-User Virtual Environments

Também a nível de MUVEs existem diversos projectos e aplicações, nomeadamente com utilização em escolas e universidades. Nas subsecções seguintes são apresentados alguns destes ambientes virtuais.

3.2.1 Second Life

Second Life é um mundo virtual 3D, desenvolvido pela Linden Labs, disponível gratuitamente *online*, onde os utilizadores navegam e interagem com o ambiente através dos seus avatares. Actualmente existem milhares de utilizadores em todo o mundo que acedem a este ambiente através da internet. Possui recursos tanto para a construção num ambiente 3D como para a interacção interpessoal síncrona, pelo que é potencial para planear e desenvolver ambientes para usos específicos [HC09] nomeadamente para fins de diversão, aprendizagem e educacionais.

É usado por professores e educadores para criar experiências de aprendizagem através da criação de ambientes específicos, onde a simulação está presente. No Second Life, o envolvimento é conseguido através de um ambiente imersivo, onde os utilizadores interagem com os objectos e entre si e constroem conhecimento. As simulações são consideradas complexas, interactivas e experiências sociais, e, como tal, são baseadas em noções de aprendizagem colaborativa e construtiva [HC09].

Figura 3.5: Second Life [Hob06] [Val08].

São vários os projectos e ilhas (espaços do mundo Second Life) existentes, bem como campus virtuais, que representam escolas e universidades reais. É possível ver na figura 3.5 as possibilidades que o ambiente virtual fornece para realização de reuniões e sessões educativas.

3.2.1.1 New Media Consortium

New Media Consortium (NMC) é uma comunidade de centenas de universidades, faculdades, museus e centros de pesquisa. Tem como objectivo estimular e promover a exploração e o uso de novos meios e tecnologias para a aprendizagem e expressão criativa [NMC06]. O Campus virtual do NMC é a maior estrutura educacional no Second Life e realiza eventos, aulas, demonstrações e exibições. Já acolheu inúmeros oradores, eventos e conferências *online* e apoia uma vasta gama de projectos [WM07].

Figura 3.6: New Media Consortium [WM07].

3.2.1.2 Schome Park

Schome Park é um projecto em desenvolvimento realizado na Universidade Aberta do Reino Unido que visa aplicar e analisar prácticas de liderança numa comunidade virtual.

Foi a primeira ilha fechada (i.e. protegida) da Europa no Teen Second Life (versão do Second Life reservada a adolescentes) [PGF08]. É uma iniciativa de investigação que tenta desenvolver uma adequação ao sistema de educação para a sociedade de hoje, tentando apoiar a aprendizagem das pessoas ao longo de toda a sua vida.

3.2.1.3 Sloodle

Sloodle (*Simulation Linked Object Oriented Dynamic Learning Environment*) é um projecto *open-source* que integra o ambiente virtual do Second Life com o sistema de gestão de aprendizagem Moodle (figura 3.7). Oferece uma gama de ferramentas de apoio à aprendizagem e ensino para o mundo virtual, ferramentas que são totalmente Integradas com um sistema de gestão baseado na *web*, testado e utilizado por centenas de milhares de educadores e estudantes por todo o mundo (Moodle). Os alunos podem fazer testes e pesquisas, apresentar trabalhos, gravar conversas e acompanhar o seu progresso através de um sistema de pontos visível através do Second Life [SLO11].

Figura 3.7: Utilização do Sloodle [McK10].

3.2.1.4 FEUP Adventure

FEUP Adventure é um projecto que visa analisar a potencialidade da utilização de um jogo digital na integração dos novos alunos, realizado por André Cruz no âmbito da sua dissertação no Mestrado Multimédia da FEUP [CSC10]. Consiste num conjunto de níveis virtuais no Second Life, utilizados na unidade curricular Projecto FEUP.

Os níveis construídos envolvem a coordenação entre os alunos, de forma a que cada equipa consiga pontuar e vencer. Existem 3 níveis criados com objectivos próprios: no primeiro nível cada equipa deve deslocar as caixas presentes para a sua plataforma, enquanto simultaneamente tenta evitar que a equipa adversária seja bem-sucedida; o segundo, tal como o primeiro, envolve a cooperação entre os elementos das equipas, já que

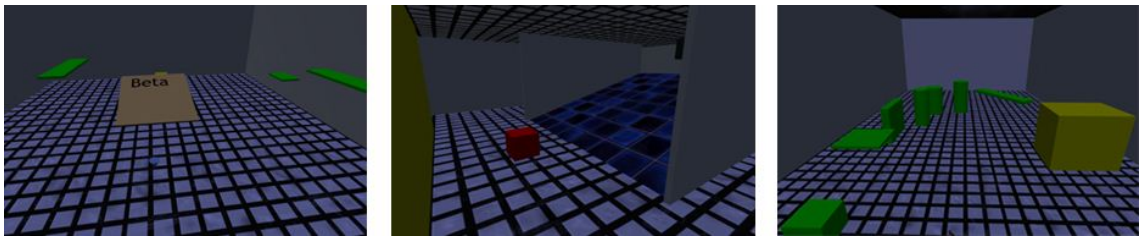


Figura 3.8: FEUP Adventure - níveis 1, 2 e 3.

consiste no deslocamento de esferas para a área correspondente a cada equipa; no terceiro nível cada estudante tem que atingir individualmente uma plataforma, percorrendo um percurso definido.

Num teste realizado com alunos do Projecto FEUP [CSC11] observou-se que a sua utilização promoveu o trabalho em equipa, verificando-se cooperação e comunicação crescentes no decorrer dos níveis, tanto a nível de interajuda no mundo virtual como a nível de comunicação directa na sala de aula.

3.2.2 Whyville

Whyville é um MUVE gráfico desenhado para crianças e adolescentes. Através de uma interface baseada na *web*, os utilizadores podem comunicar com amigos e conhecidos através de chat e através do Whyville-Times, um jornal com artigos criados pelos utilizadores, aprender matemática, ciência e história através de actividades interactivas e construir uma identidade virtual [DC07]. Whyville foi dos primeiros mundos virtuais direccionados para crianças, bem como é dos poucos MUVEs cujo principal objectivo é educacional.

3.2.3 The River City Project

River City é um MUVE com aplicações educativas, onde alunos do secundário aprendem acerca de investigação científica e aprendem competências em diversas áreas [Uni09a]. O Mundo virtual de River City consiste numa cidade do século XIX atravessada por um rio, com diferentes formas de terreno que influenciam o fluxo da água, vários vizinhos, indústrias e instituições como hospitais e universidades. Os próprios estudantes populam a cidade, juntamente com agentes *computer-based*, objectos digitais que podem incluir clips de áudio ou vídeo, e os avatares dos instrutores [KDCN06]. É possível ver a interface do River City na figura 3.9.

Os estudantes empenham-se na aquisição de competências, principalmente através dos seguintes comportamentos [KDCN06]:

- Observação da cidade virtual;

Figura 3.9: Interface de River City [Uni09b].

- Realização de questões aos residentes virtuais, reunindo informação que por vezes oferece pistas para os problemas;
- Análise de livros e outras formas de informação;
- Utilização de ferramentas para reunir, analisar e interpretar dados;
- Planeamento de experiências científicas - os estudantes são guiados através de um processo generalizado do método científico, que culmina na criação de uma experiência para testar as suas hipóteses acerca dos problemas;
- Propostas de respostas, explicações e previsões sobre os problemas;
- Comunicação dos resultados.

Num estudo realizado em [KK05] foi examinada a percepção obtida por alunos que utilizaram os MUVEs River City e Whyville. A sua opinião foi genericamente positiva acerca da utilização de ambos, como ferramentas na sala de aulas. Enquanto o River City era considerado mais educacional que o Whyville, muitos estudantes viram isso como uma característica positiva e rapidamente se verificou que, apesar de o Whyville ser divertido, o chat e a criação de avatares prejudicavam a atenção para a ciência [KK05].

3.3 Conclusão

A indústria de *Serious Games* encontra-se também em rápida expansão, e existem inúmeras empresas a apostar nesta área [Des11]. Organizações militares, corporativas, educativas e de saúde de todo o mundo têm também aproveitado os efeitos positivos que a implementação de SG trás às suas necessidades educativas [Der07].

De forma similar a utilização de MUVEs como ambientes para promover a aprendizagem é vista actualmente como uma boa aposta e cada vez mais surgem projectos nesse

sentido, envolvendo universidades, professores, empresas e muitas outras entidades. O estudo realizado em [KDCN06] mostra que através de um MUVE é possível ensinar conteúdos padrão (biologia neste estudo) aliados a capacidades complexas de investigação, melhor que as abordagens tradicionais. Os resultados do estudo mostram que os alunos aprenderam o conteúdo pretendido, que houve grande envolvimento por parte de alunos e professores, que a assiduidade dos alunos melhorou, e, sobretudo, que a utilização deste tipo de tecnologia numa sala de aula pode facilitar uma boa aprendizagem e investigação.

Capítulo 4

Modelação Expedita de um Jogo Sério em Ambiente Virtual

Após a análise das ferramentas para a criação de jogos, e tendo em conta o âmbito e as necessidades deste projecto, cujo objectivo requer que a interacção seja simples e de aprendizagem rápida, com uma forte capacidade de suporte ao aspecto *multiplayer*, a utilização de um ambiente virtual 3D multi-utilizador como o Second Life ou o OpenSimulator surge como a melhor hipótese para o desenvolvimento da plataforma. Permite, desde logo, ter toda a estrutura cliente/servidor disponível e pronta a utilizar, com todo o processo de criação de avatares e personalização das personagens já existente. Além disso, o seu suporte ao controlo da física no ambiente virtual, apesar de simples e limitado, é suficiente tendo em conta a mecânica pretendida para o desenvolvimento dos níveis de jogo.

Assim, nesta dissertação pretende-se desenvolver uma plataforma que permita construir níveis de jogo num ambiente virtual, com possibilidade de facilmente recriar a mecânica existente no projecto FEUP Adventure, já que este representa um trabalho que comprova a possibilidade de utilizar um mundo virtual 3D como o Second Life na promoção do processo de integração de novos estudantes no ensino superior. Tendo em conta a necessidade de alocar centenas de utilizadores a intervalos de tempo e espaço virtual limitado, este projecto deve ser construído de forma a controlar o processo de geração de níveis e o decorrer do jogo.

Neste capítulo é apresentada a metodologia utilizada para solucionar o problema enunciado nesta dissertação, descrevendo os processos necessários para a criação de um mecanismo de modelação expedita de níveis de um jogo sério num ambiente virtual multi-utilizador.

4.1 Utilização do Espaço Virtual

O mundo virtual, assim como o real, tem limitações a nível de espaço e extensão de terreno e, considerando que a utilização do Second Life e a respectiva necessidade de manter uma ilha virtual representa custos monetários, é necessário considerar o desenvolvimento de uma plataforma que permita rentabilizar e reutilizar o espaço disponível. No projecto FEUP Adventure, os diferentes níveis foram criados em "camadas", distribuindo verticalmente os diferentes pisos em que os níveis se encontram construídos (figura 4.1). Cada nível do jogo foi construído no seu próprio espaço e, para a sua utilização, todos os seus elementos (plataformas, rampas, objectos, etc) devem ser permanentemente mantidos.

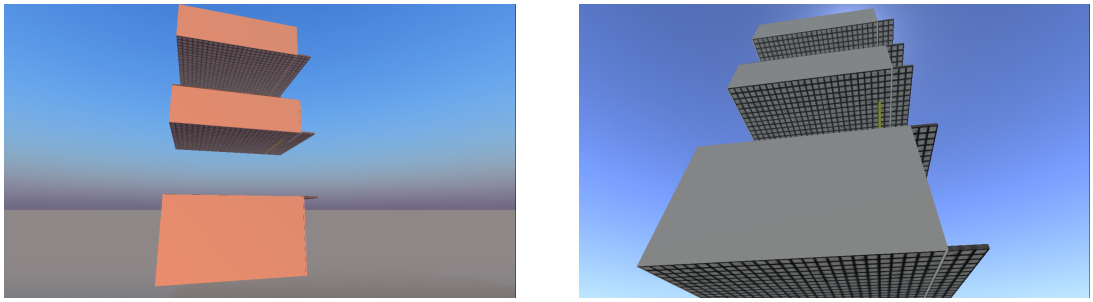


Figura 4.1: Distribuição dos níveis no Second Life no projecto FEUP Adventure.

Com esta abordagem é possível rentabilizar o espaço existente, no entanto fica limitada à altitude máxima que o ambiente virtual fornece. Cada região tem uma altura máxima de 4096 metros no Second Life e de 10000 metros no openSimulator [VCM+10], e tendo em conta que cada nível pode necessitar de uma altura elevada para hospedar correctamente os diferentes elementos do jogo, o número máximo de níveis que é possível dispor verticalmente é limitado.

É também necessário considerar a distância mínima necessária entre os diferentes níveis, de forma a, por um lado evitar que existam demasiados objectos criados na mesma cena a ser renderizados nos programas cliente, e por outro lado evitar que os estudantes e todos os demais utilizadores da plataforma consigam facilmente pré-visualizar os níveis. Tal situação pode acontecer devido à possibilidade de controlar a câmara por parte de cada utilizador, cujo alcance pode ser definido entre 64 e 512 metros, relativamente à posição do avatar [VCM+10].

Abordagem: Ambiente Virtual Único

A construção de uma plataforma única de modelação de jogos permite recorrer apenas a um piso, onde todos os níveis de jogo são construídos, utilizados e apagados. Através da geração dos objectos virtuais de forma expedita e imediata é possível garantir que

apenas num piso se reproduzem todos os níveis, não havendo necessidade de manter os elementos de jogo persistentes e inalterados.

Assim, a abordagem a adoptar consiste na criação de cada nível individualmente, conforme a utilização pretendida em cada sessão de trabalho com os utilizadores. Todos os elementos e objectos de cada nível são criados procedimentalmente em poucos segundos, de acordo com a definição do nível que esteja a ser utilizada nesse momento. O processo de definição dos níveis a utilizar em cada sessão é efectuado através de um ficheiro externo ao servidor do mundo virtual. Após a criação do nível, todos os elementos estão prontos a ser utilizados. É possível alocar cada utilizador à sua equipa e imediatamente começar o jogo. Após o final do jogo, pode-se facilmente apagar todos os objectos virtuais presentes, redefinir o novo nível de jogo que se pretende utilizar e criar de forma expedita o novo nível.

Tudo isto ocorre no mesmo espaço virtual onde os jogadores se encontram: nesse espaço eles apenas jogam e esperam pela redefinição dos níveis virtuais. Visto que a criação de cada nível é um processo rápido, é possível garantir que os utilizadores não ficam entediados nem distraídos enquanto esperam para jogar. Por outro lado, o facto de todos os avatares se encontrarem no mesmo espaço virtual garante um maior controlo por parte do supervisor presente na sessão, já que não é necessário proceder a deslocações entre diferentes áreas.

4.2 Mecânica de Jogo

Na definição da mecânica de jogo, e considerando a possibilidade de integração deste projecto com as turmas do Projecto FEUP, é necessário ter em conta os seus horários, sendo que em cada sessão se reúnem dois grupos de estudantes, que correspondem a duas equipas no jogo (diagrama da figura 4.2). As duas equipas competem, ao estilo dos Jogos sem Fronteiras[1], e em cada nível apenas uma pode vencer. Cada jogador tem o correspondente avatar que lhe permite jogar o nível e, dentro de cada equipa, é necessário existir cooperação e comunicação para que esta consiga vencer.

Em cada sessão existe um supervisor que tem acesso ao controlo dos diferentes níveis. Ele controla os acessos por parte dos jogadores, de forma a verificar se todos os estudantes estão correctamente autenticados e inseridos como jogador no nível e na correspondente equipa. Também compete ao supervisor definir o esquema dos níveis da sessão, tomando decisões acerca da dificuldade do nível, número ou tipo de objectos presentes, objectivos e pontuações, de acordo com os objectos virtuais existentes e possíveis de ser utilizados. Esta definição ocorre fora do mundo virtual.

[1] Os Jogos Sem Fronteiras eram um programa televisivo onde equipas de diferentes países competiam numa espécie de Jogos Olímpicos, participando em variados jogos com tarefas simultaneamente exigentes e divertidas.

Figura 4.2: Diagrama da mecânica de jogo.

Da concretização dos níveis surgem os resultados das prestações das equipas, que permitem concluir e analisar as estratégias utilizadas e a cooperação existente entre os jogadores de cada equipa. Simultaneamente é estudado o comportamento e a interacção de cada estudante, entre si e com o ambiente virtual. Estes resultados, em conjunto com o *feedback* de cada estudante, permitem avaliar e analisar a eficácia do jogo e concluir qual o nível de cumprimento dos seus objectivos, isto é, verificar se existe aumento e melhoria na comunicação entre os estudantes e uma maior capacidade de integração no grupo e no novo ambiente.

4.3 Construção dos Níveis

Para ser possível gerar e recriar objectos 3D nos ambientes virtuais Second Life e OpenSimulator e ser possível a modelação dos níveis de jogo, é necessário que esses objectos se encontrem previamente construídos e guardados no inventário de um objecto inicial, que irá construir toda a cena. Isto pode representar a maior limitação à utilização de modelação expedita, já que é necessário que cada elemento a utilizar seja criado e as suas propriedades sejam definidas inicialmente. No entanto, uma vez completa a fase inicial de criação e definição, todos os elementos estão prontos a ser inseridos no mundo virtual e utilizados no jogo. Além disso, tendo em conta a mecânica simples e específica pretendida, é possível especificar todos os tipos de objectos que poderão ser necessários

para o decorrer dos níveis, pelo que a sua criação inicial não corresponde a um grande impedimento para a definição dos diferentes níveis.

Os diferente objectos utilizados podem ser, entre outros:

- **Rampas**, que podem pertencer a uma equipa, sem que a outra equipa a possa utilizar, ou podem ser comuns, e todos os jogadores podem utilizar;
- **Blocos**, que podem ser fixos, usados como obstáculo ou com alguma interacção com outros elementos, ou móveis, que os jogadores devem movimentar;
- **Esferas**, para os jogadores deslocarem de forma a cumprir os objectivos dos níveis;
- **Plataformas**, que podem ser fixas ou móveis, e podem pertencer a uma equipa ou ser um elemento comum a todos os jogadores;
- **Paredes**, para limitar o espaço de jogo ou para representar as zonas de cada equipa.

Como todos os objectos necessários são na sua maioria representações de sólidos simples, permitem facilmente interagir e controlar os movimentos de jogo, mesmo através da física simples e limitada de que se dispõe no Second Life e no OpemSimulator.

4.3.1 Controlo do Jogo

É necessário também ter em conta elementos de jogo que são necessários, independentemente do nível de jogo que se esteja a utilizar. Deve existir um objecto de controlo de jogo, através do qual se dá início ao jogo e com o qual se controla a duração do mesmo. Durante o decorrer do jogo deve informar constantemente o tempo restante até ao final da sessão para que seja possível ao supervisor e aos jogadores estarem a par da duração do jogo e do tempo disponível até terminar. O objecto de controlo de jogo é também responsável por receber, gerir e guardar toda a informação originário do decorrer do jogo, como pontuações, tempos e composições das equipas.

4.3.2 Identificação dos Jogadores

Existem também elementos que devem permitir identificar os jogadores de cada equipa: correspondem a objectos que os jogadores devem "vestir" nos seus avatares, ficando visíveis de forma a ser possível identificar a cor da sua equipa. A criação das equipas deve ser realizada antes do início do jogo, através da distribuição da identificação pelos jogadores. Quando um objecto de identificação é atribuído a um jogador, este fica colocado e visível na mão do seu avatar, com a indicação do estado de jogo - se está a decorrer ou parado. A informação de cada jogador que entra ou sai de uma equipa deve ser registada e guardada na informação do jogo, além de ser indicada no chat público, de forma a garantir que o

supervisor esteja a par de eventuais saídas de equipas durante o decorrer do jogo por parte dos jogadores.

4.3.3 Semente de Construção

Como já foi referido, todos os objectos e elementos necessários devem ser inicialmente criados, definidos e guardados. Devem estar guardados no inventário de um outro objecto de construção, a semente, que os irá recriar no mundo virtual. Desta forma, a semente tem acesso a todos os objectos virtuais, sempre que for requisitada a construção de um novo nível. Tal elemento constitui um objecto presente no mundo virtual, através do qual é feita a interacção de construção e destruição dos diferentes níveis de jogo.

Corresponde ao único objecto que está presente inicialmente no espaço virtual onde irá decorrer o jogo. Através dele, o supervisor de cada sessão solicita a criação de um novo nível, que deve estar definido previamente. A semente procede à criação do nível, através da recriação de cada elemento de jogo presente no seu inventário, com as propriedades definidas. A colocação de cada elemento de jogo no cenário é feita através da deslocação da semente para a posição definida correspondente ao elemento. Estando na posição correcta, a semente de construção cria o elemento e desloca-se até à posição do objecto seguinte, sucessivamente, até que todo o nível esteja construído.

Após o fim do jogo, o supervisor pode requerer a destruição do nível através da semente, que se encarrega de eliminar todos os elementos desse nível, deixando novamente o espaço virtual vazio.

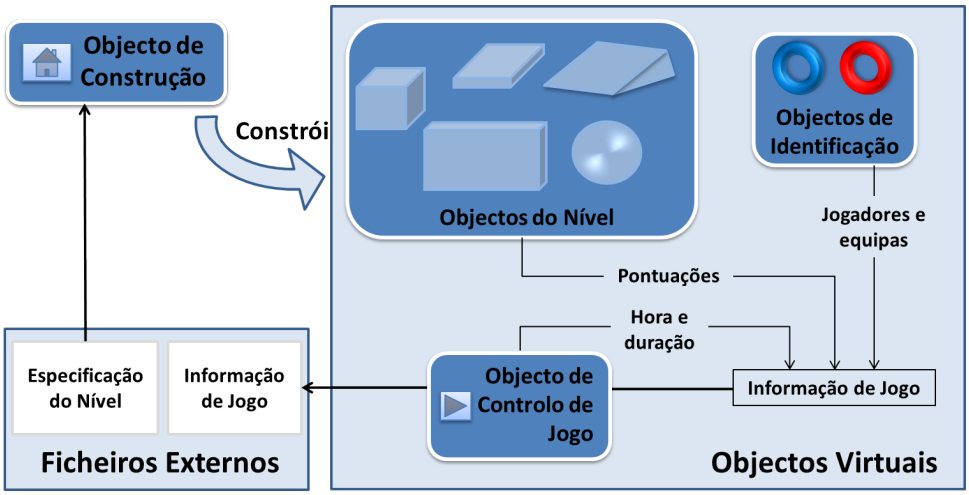


Figura 4.3: Esquema geral dos objectos existentes.

Na figura 4.3 é possível ver um esquema dos objectos existentes no mundo virtual e a sua interacção com os ficheiros externos. A semente (objecto de construção) toma conhecimento da definição do nível pretendida e constrói os objectos virtuais na cena, como foi referido anteriormente. Desses objectos criados fazem parte os objectos de

identificação, os objectos do nível e o objecto de controlo de jogo. Todos eles fornecem informação de jogo, que é gerida e enviada pelo objecto de controlo de jogo para o ficheiro externo, onde fica guardada.

4.3.4 Ficheiros Externos

Os ficheiros externos têm como objectivo efectuar o armazenamento de informação que pode ser visualizada e editada externamente ao mundo virtual. No esquema da figura 4.3 é possível perceber o papel que eles representam na interacção com os objectos existentes. Assim, existe o ficheiro de especificação do nível, que corresponde ao ficheiro com a configuração que se pretende utilizar no jogo. Nele são indicados os objectos virtuais que se pretendem que a semente construa, bem como a sua posição e orientação na área de jogo. O ficheiro de informação de jogo regista os acontecimentos de cada sessão de jogo ocorrida. Nele são guardadas as alocações dos jogadores às equipas, com registo de cada entrada e saída, a data e hora do início e fim do jogo bem como as pontuações das equipas e outras informações do decorrer do jogo que seja conveniente registar. A definição, utilização e formato destes ficheiros é explicada mais detalhadamente no capítulo 5.

4.4 Conclusão

Neste capítulo foi abordada a metodologia para o desenvolvimento de um mecanismo que permita a modelação de níveis de jogo para utilização em ambientes virtuais 3D multi-utilizadores, com o objectivo de dinamizar grupos de trabalho através da cooperação e interacção entre os jogadores.

Tendo em conta as vantagens que disponibilizam, decidiu-se utilizar os ambientes virtuais Second Life e OpenSimulator, já que possibilitam a utilização de um ambiente 3D multi-utilizador com uma estrutura cliente/servidor completamente disponível e funcional e com o processo de autenticação e criação de avatares já disponível. Como a utilização destes ambientes virtuais apresenta limitações a nível de espaço disponível, foi apresentada neste capítulo a abordagem utilizada para rentabilização do espaço disponível, através da reutilização da mesma área para criação alternada de todos os níveis de jogo necessários num mesmo espaço virtual.

Considerando o objectivo desta dissertação de obter níveis de jogo que potenciem a dinamização de grupos, e considerando também a física simples e limitada que se tem disponível nos ambientes virtuais Second Life e OpenSimulator, foi tida em conta a mecânica de jogo existente no projecto FEUP Adventure. Permite uma aprendizagem rápida e eficaz por parte dos jogadores, através de interacções e movimentos simples no ambiente 3D, que requerem a cooperação entre os diferentes elementos das equipas, ao estilo

dos Jogos Sem Fronteiras, em que equipas competem entre si, utilizando elementos e objectos básicos.

Capítulo 5

Implementação

Tendo em conta os objectivos propostos para esta dissertação, foi desenvolvido um mecanismo de modelação de um jogo sério, tendo sido implementados níveis protótipo que apoiem e promovam o processo de integração e de dinamização de grupos de trabalho. Assim, foi desenvolvida uma plataforma que permite facilmente criar níveis de jogo, através da modelação expedita nos ambientes virtuais Second Life e OpenSimulator, com uma mecânica que permite que duas equipas compitam para atingir os objectivos de cada nível.

Neste capítulo é apresentado o trabalho realizado a nível de implementação para o desenvolvimento desta plataforma, com a especificação das ferramentas utilizadas e das decisões técnicas tomadas.

5.1 Arquitectura

A plataforma corre numa arquitectura cliente/servidor, onde os clientes controlam os seus avatares, enviando todas as acções para o servidor do jogo. É possível ver o esquema da arquitectura na figura 5.1. Como clientes existem os Estudantes (ou outro tipo de jogadores), que efectuam os pedidos de interacção com o mundo, e os Supervisores, que assumem o papel de controlo sobre os acessos por parte dos estudantes e controlam o decorrer do jogo, as definições dos níveis e a construção e eliminação dos elementos de jogo. Os clientes interagem com o mundo virtual através de uma GUI, onde controlam o seu avatar e se movimentam nas áreas de jogo.

Toda a configuração do nível de jogo a utilizar encontra-se definida num ficheiro externo ao servidor, *gamedefinition*, onde estão guardadas as especificações de criação do nível de jogo. Neste ficheiro são definidos todos os objectos necessários, a sua posição no mundo virtual e a sua orientação. Esta configuração deve ser controlada pelo supervisor

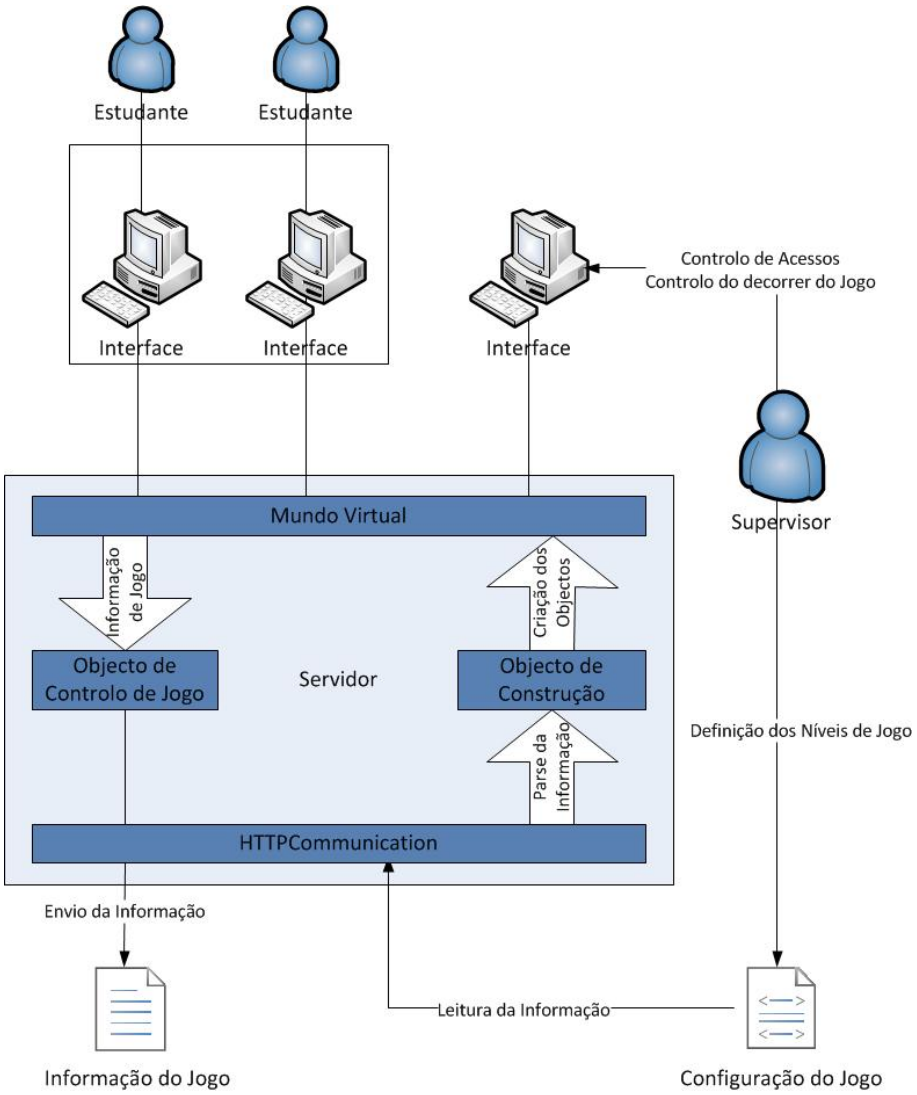


Figura 5.1: Arquitectura da plataforma.

da sessão, que deve seleccionar o *layout* do nível que pretende utilizar para ser jogado pelos estudantes. A estrutura e linguagem deste ficheiro é explicada mais detalhadamente na secção 5.3.

Esta configuração é lida pelo servidor através de um pedido HTTP, que através do *url* do ficheiro obtém todo o seu conteúdo, ao qual é feito o *parsing*. A partir desta análise sintática são interpretados quais os objectos que serão utilizados e quais as suas propriedades. Com esta informação disponível, o objecto inicial de construção (semente) encarrega-se de recriar os elementos de jogo na posição e com a orientação pretendidas. Estes são apresentados no mundo virtual, prontos para serem utilizados no decorrer do jogo.

Aos pedidos de interacção dos clientes (estudantes e supervisor) com o mundo virtual, o servidor serializa as acções, ajusta o mundo virtual e envia as alterações para todos os

clientes afectados (que estejam próximos da acção). As acções do decorrer do jogo: início e fim de jogo, jogadores a entrar ou sair de uma equipa, pontuações, etc, são registadas no objecto de controlo de jogo. Este encarrega-se de enviar toda a informação através de um pedido HTTP para um *script* PHP externo ao servidor, que a recebe e grava num ficheiro (*gameinfo*).

5.2 Plataformas de Desenvolvimento

Nesta secção pretende-se abordar as tecnologias e plataformas utilizadas para o desenvolvimento do protótipo, que correspondem às plataformas alvo desta dissertação a nível da utilização de um ambiente virtual 3D multi-utilizador.

5.2.1 Second Life

A utilização do Second Life como plataforma de desenvolvimento e manutenção do mundo virtual surge tendo em consideração as vantagens que oferece quanto à estrutura já completamente desenvolvida a nível do ambiente virtual 3D, criação dos avatares e toda a estrutura de rede. É possível tirar partido da sua arquitectura cliente/servidor e de toda a estrutura de autenticação que se encontra já implementada.

Fornece um visualizador que, por um lado permite aos jogadores autenticarem-se no mundo virtual, criar e personalizar os seus avatares e interagir e utilizar os objectos virtuais, e por outro lado disponibiliza ferramentas de criação e edição de objectos, que podem ser guardados nos servidores do Second Life e facilmente ser utilizados e recriados no mundo virtual.

O Mundo virtual do Second Life consiste em muitos simuladores interconectados (também conhecidos por regiões ou *sims*) posicionados num mapa cartesiano. Os simuladores são conectados com os seus vizinhos verticalmente e horizontalmente no mapa, formando uma rede que compõe o mundo virtual. Cada simulador tem 256m x 256m de dimensão e é responsável por manter disponível a informação acerca dos objectos e agentes no seu interior, simular a física, executar os *scripts* e guardar e transmitir dados de objectos e texturas presentes no simulador para os clientes (cache) [Ezh09]. A implementação técnica deste protótipo é desenvolvida numa parcela da região pertencente à Universidade do Porto, onde se encontra também desenvolvido o projecto FEUP Adventure. Assim, numa sessão de utilização desta plataforma no âmbito do Projecto FEUP, os estudantes e o supervisor devem dirigir os seus avatares para a ilha virtual da UP, e aí aceder à parcela do Projecto FEUP.

Para a criação da interacção entre os elementos do mundo virtual e desenvolvimento dos seus comportamentos, o Second Life utiliza uma linguagem de *scripting*, a Linden

Scripting Language (LSL) que é referida mais detalhadamente no capítulo 5.2.3. É possível criar e editar os *scripts* LSL no editor interno do visualizador ou num editor externo. Se um *script* é colocado num objecto, pode controlar o seu comportamento e a sua aparência, podendo, entre outras coisas, alterar a sua cor, movê-lo ou interagir com o mundo. Um único objecto pode conter múltiplos *scripts*, cada um correndo independentemente em relação aos outros (apesar de todos terem efeito no mesmo objecto). Cada *script* do mundo virtual recebe uma porção de tempo do tempo total que o simulador aloca para a execução de *scripts*. Assim, um simulador com muitos *scripts* aloca menos tempo de processamento a cada um deles, em vez de degradar a sua própria performance. Além disso, cada *script* é executado dentro da sua secção de memória, prevenindo que acedam a memória protegida do simulador ou a outros *scripts* (incluindo múltiplos scripts no mesmo objecto), o que faz com que seja muito mais difícil que um simulador falhe devido à execução dos *scripts* LSL [Chr09].

5.2.2 OpenSimulator

O OpenSimulator é um servidor *open source* que, similarmente ao Second Life, hospeda ambientes virtuais 3D multi-utilizadores. É possível conectar ao OpenSimulator através dos mesmos visualizadores cliente com que se conecta ao Second Life, uma vez que ambos utilizam os mesmos protocolos de comunicação. Também a interacção e edição nos mundos virtuais é semelhante, já que o OpenSimulator suporta cerca de 95% de todos os comandos da linguagem LSL.

Pode funcionar em um de dois modos: *Standalone* ou *Grid*. No modo *standalone* tanto o simulador da região como todos os serviços de dados (gestão dos utilizadores, objectos e inventários) são executados no mesmo processo. É possível ter várias regiões em execução, mas todas na mesma máquina. No modo *grid* os serviços de dados são processos diferentes do servidor da região. Eles podem encontrar-se a executar em conjunto ou separados por várias instâncias. Isto permite que possam correr em diferentes máquinas, se necessário [Ope11]. Para a criação do protótipo no OpenSimulator foi utilizado o modo *standalone* com execução local da região, já que, por um lado apenas era pretendido utilizar o ambiente virtual para teste do mecanismo de modelação expedita, sem recorrer à utilização por múltiplos utilizadores, e por outro lado a instalação do modo *grid* requer uma maior compreensão de conceitos como identificadores, localização geométrica, comunicação com servidores, estados e outros conceitos que não se enquadram no âmbito desta dissertação.

Na tabela 5.1 é possível verificar algumas distinções entre o Second Life e o OpemSimulator. Desde logo a não existência de grupos nem de moeda corrente, que limita a possibilidade de distribuir os jogadores por grupos de trabalho ou de facilmente atribuir objectos a jogadores (com a existência de moeda no Second Life, é feita a atribuição de

Tabela 5.1: Comparação dos principais factores entre o Second Life e o OpenSimulator [ACT09].

Característica	**Second Life**	**OpenSimulator**
Áudio	Sim	Sim - módulo FreeSWITCH
Grupos de utilizadores	Sim	Não de origem
Mensagens *Offline*	Sim	Não de origem
Moeda corrente	Sim	Não de origem
Scripting	Sim	Sim - ligeiramente diferente
Upload de texturas	Sim - pago	Sim - grátis
Edição de terrero	Sim	Sim
Modo *Grid*	Sim	Sim
Construção de objectos	Sim	Sim - com algumas falhas.

objectos através da "venda" a custo 0). Também a capacidade de *scripting* não é completamente igual, já que o OpenSimulator não suporta a totalidade da linguagem LSL. Por outro lado o OpenSimulator permite o *upload* de texturas gratuitamente, funcionalidade que no Second Life é paga.

5.2.3 Linden Scripting Language

O Second Life fornece capacidades de *scripting* através da Linden Scripting Language, que é uma linguagem de programação baseada em estados e orientada a eventos. Habitualmente um *script* é anexado a um objecto (através da sua pasta de conteúdos) e usado para programar o comportamento desse objecto no mundo virtual. Um *script* LSL suporta variáveis, funções e estados, no entanto a LSL não é uma linguagem orientada a objectos. Classes e interfaces não existem e conceitos como herança ou encapsulamento não podem ser implementados. isto pode efectivamente corresponder a uma limitação para programadores de jogos, já que a maioria dos jogos criados actualmente usa o paradigma OOP na sua concepção [CSC11]. Também a implementação de código, testes e depuração podem não ser o mais flexível possível, já que todo o trabalho deve ser realizado no mundo virtual, controlando a criação de *scripts* e de objectos através de uma personagem.

Apesar de algumas limitações, a LSL tem mais de 300 funções e mais de 30 eventos. Muitas das operações comuns usadas na programação de jogos, como a detecção de colisões, instanciação de objectos, temporizadores, etc, estão presentes. Os objectos podem ser ligados aos *scripts* LSL, definindo os seus estados, funções e as suas acções de acordo com a ocorrência de eventos [CSC11].

Quando é criado um novo *script*, o código LSL seguinte está presente por defeito:

```
default
{
  state_entry()
  {
```

```
    llSay(0, "Hello, Avatar!");
  }
  touch_start(integer total_number)
  {
    llSay(0, "Touched.");
  }
}
```

Este *script* representa apenas um pequeno exemplo que funciona como um simples comportamento até que um *script* mais complexo seja criado. No entanto, é possível observar no código a criação de estados, o tratamento de eventos e a chamada de funções.

A linguagem também suporta comunicação do lado do servidor através do protocolo HTTP, que permite realizar a comunicação com servidores externos, necessários neste protótipo para a definição dos níveis e para guardar a informação de jogo. A utilização de HTTP permite comunicação flexível e escalável, sendo possível aos objectos virtuais trocar informação através dos seus *scripts* com elementos externos ao mundo virtual.

5.3 Construção dos Níveis

Para ser possível a construção de um nível de jogo em ambos os ambientes Second Life e OpenSimulator é necessário passar por duas fases distintas de implementação: a modelação dos objectos no mundo virtual e a definição do *layout* pretendido para o nível no ficheiro *gamedefinition*.

5.3.1 Modelação dos Objectos

A construção dos objectos presentes nos níveis é feita através das possibilidades de modelação e edição que as plataformas de visualização dos mundos virtuais oferecem. Assim, a criação é efectuada em pleno ambiente virtual, tanto no Second Life como no OpenSimulator, através de um avatar (também é possível importar modelos 3D através de *Sculpt Maps*). Todos os objectos criados no mundo virtual são compostos por objectos primitivos, que representam uma forma geométrica básica (figura 5.2).

Figura 5.2: Formas geométricas básicas que é possível criar no mundo virtual (*Second Life Viewer*).

Estes objectos primitivos, ou *prims*, podem ser ligados para formar objectos maiores e diferentes, como é possível ver na figura 5.3.

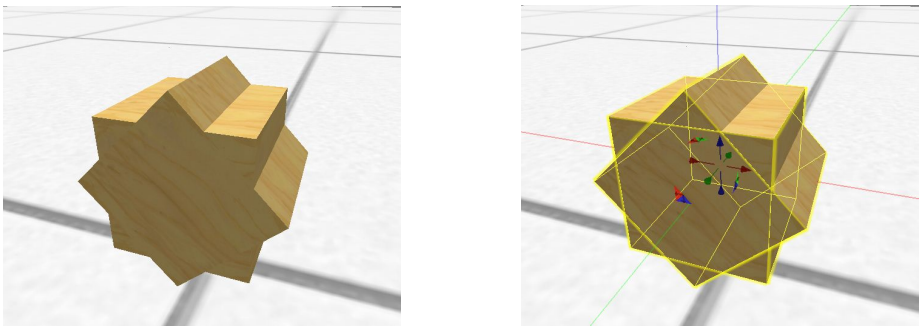


Figura 5.3: Objecto complexo formado pela ligação de duas forma geométricas básicas.

Assim, todos os elementos pretendidos para utilizar na construção de um nível de jogo devem ser previamente criados, editados e guardados no inventário da semente, para ser possível a sua utilização na construção dos níveis. A edição visual de um objecto passa pela definição das suas dimensões, da sua textura, da sua cor e a escolha do seu material. Para definir o seu comportamento bem como para ser possível a sua criação e eliminação no processo de geração procedimental de conteúdos, é necessário a criação do *script* LSL, que deve ser anexado ao objecto. Cada objecto é identificado pelo seu nome, pelo que devem ter nomes representativos e claros, que permitam identificação individual; assim, para identificar os objectos pertencentes à equipa vermelha, estes chamam-se *redTeam<objectType>*, como por exemplo *redTeamRing*, *redTeamRamp* ou *redTeamPlatform*.

A fase de criação dos objectos é o primeiro processo realizado após a definição da mecânica e dos elementos de jogo, já que é necessário garantir que os objectos existem no mundo virtual e que se tem conhecimento acerca deles na definição dos níveis de jogo.

5.3.2 Definição do Nível

Como já foi referido, a construção dos níveis de jogo no ambiente virtual segue a especificação que foi previamente definida no ficheiro externo *gamedefinition*. A comunicação com este ficheiro fica a cargo do objecto semente de construção, cujo nome neste protótipo é *HTTPCommunication*. Quando o supervisor confirma que pretende construir um novo nível, o *script* do objecto entra num estado de comunicação em que efectua o pedido através da chamada à função:

```
llHTTPRequest(string url, list parameters, string body)
```

O argumento *url* corresponde ao endereço em que se encontra o ficheiro *gamedefinition*, que está disponível em http://paginas.fe.up.pt/∼ei06027/lsl/gamedefinition. Os argumentos seguintes da função correspondem ao corpo e parâmetros do pedido HTTP que a função realiza ao *url*, pelo que são passados em branco, para que seja retornado todo o conteúdo do ficheiro.

O conteúdo de resposta encontra-se limitado ao tamanho máximo de 2048 bytes, pelo que é importante utilizar o mínimo de texto possível na definição dos níveis, de forma a evitar a perda de informação na comunicação e consequente falha na modelação no mundo virtual. Assim, o ficheiro de definição dos níveis é construído segundo uma linguagem simples de configuração dos objectos. Permite definir o *layout* do nível a jogar através da indicação dos objectos criados no mundo virtual e possibilita que o objecto *HTTPCommunication* interprete a informação e recrie os objectos virtuais. Em cada linha é inserida a informação de um objecto a desenhar no mundo virtual:

```
//<object_name> <pos_x> <y> <z> <rot_x> <y> <z> (<extra_param>)

redTeamRamp 17.5 0.0 1.5 0.0 0.0 180.0
redTeamPlatform 23.5 0.0 2.8
blueTeamRamp -17.5 0.0 1.5
blueTeamPlatform -23.5 0.0 2.8

platform 10.0 -11.0 4.0
platform -10.0 9.0 4.0

blueTeamRing -1.0 2.5
redTeamRing 1.0 2.5

startGame 0.0 0.0 5.0 180.0 0.0 0.0 60.0
```

Assim, cada linha deve conter o nome do objecto a criar, a sua posição, através das suas coordenadas X, Y e Z, e a sua orientação, também através dos valores de X, Y e Z. A posição fornecida é relativamente à posição da semente, pelo que a disposição de objectos na cena deve ter em conta a posição inicial de *HTTPCommunication*. Considerando a utilização de uma sala virtual para a criação dos níveis de jogo, a semente encontra-se no centro geométrico do espaço, ao nível do chão (figura 5.4), pelo que as coordenadas da posição de cada objecto em *gamedefinition* são interpretadas tendo em conta um referencial imaginário, cuja origem corresponde ao ponto central do espaço virtual, ao nível do chão.

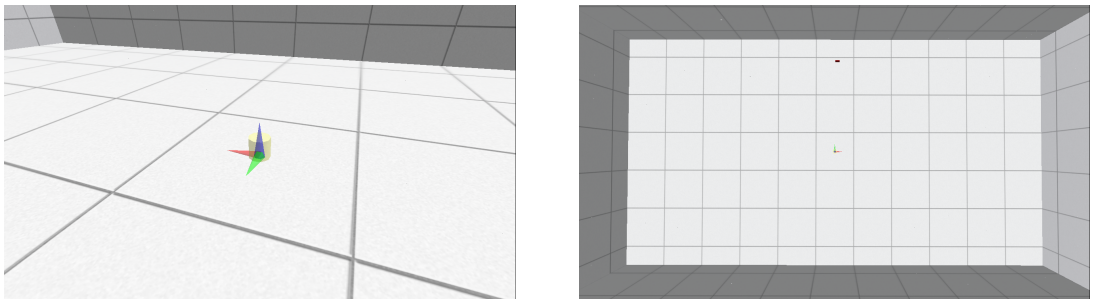
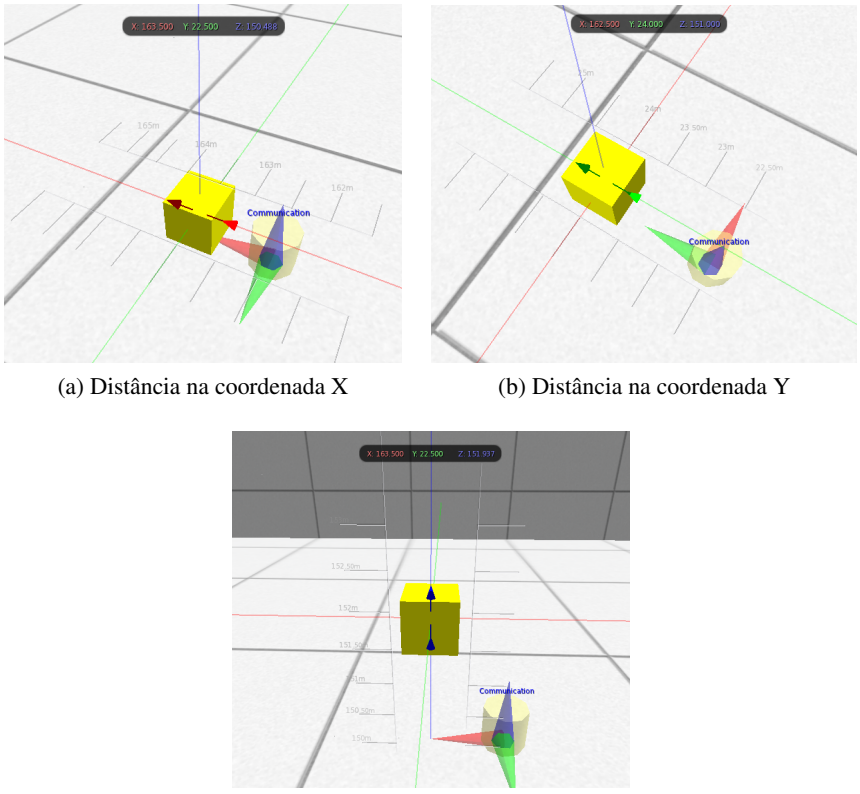


Figura 5.4: *HTTPCommunication* colocado no centro do espaço virtual para a construção do nível.

Os valores dados a cada coordenada correspondem a metros virtuais, pelo que uma linha com a seguinte informação:

```
object 5.0 5.0 5.0
```

iria construir o objecto *object* na posição (5, 5, 5) relativamente ao centro do espaço virtual, isto é, com 5 metros de distância relativamento à posição 0 de cada coordenada (X, Y, Z). É possível ver na figura 5.5 a colocação de um objecto relativamente à posição da semente nas coordenadas X, Y e Z.



(a) Distância na coordenada X

(b) Distância na coordenada Y

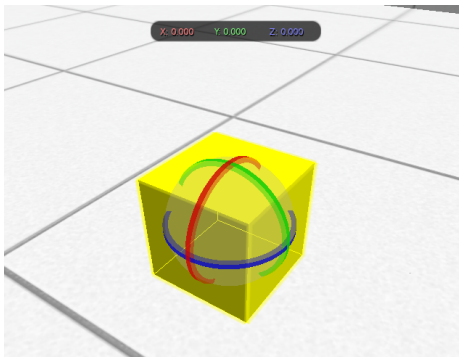
(c) Distância na coordenada Z

Figura 5.5: Relação de distância entre a posição de um objecto e a posição de *HTTPCommunication* nas coordenadas X, Y e Z.

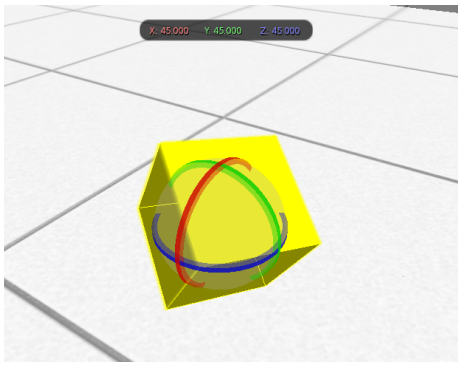
A orientação do objecto é dada pela rotação relativamente ao centro do próprio objecto, e os valores fornecidos correspondem aos graus pretendidos de rotação (de 0^o a 360^o) relativamente aos eixos de coordenadas (X, Y, Z). Assim, uma rotação de 180^o na coordenada X iria colocar o objecto invertido verticalmente. Na figura 5.6 é possível verificar as diferentes orientações de um objecto com rotação nos 3 eixos de coordenadas.

Existem objectos que podem aceitar um parâmetro extra na sua criação (por exemplo o objecto de início de jogo que recebe o tempo de duração do jogo através deste parâmetro), pelo que este valor deve ser colocado no final da linha.

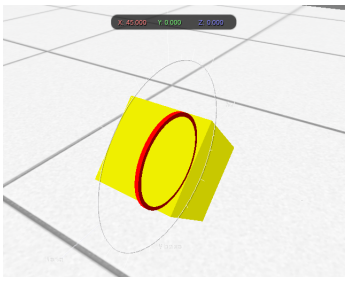
É necessário ter em atenção que a ordem da informação contida em cada linha deve ser respeitada, ou seja, se se pretender colocar o parâmetro extra na construção de um



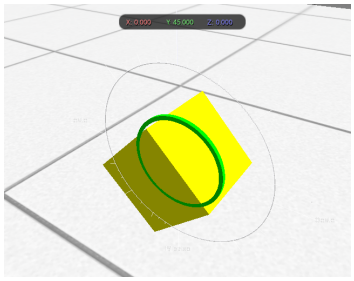
(a) Orientação original



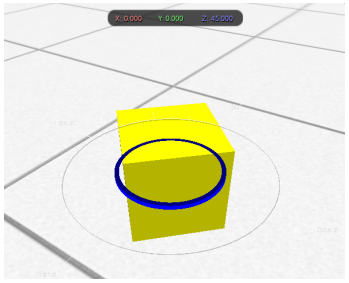
(b) Rotação de 45º nos 3 eixos de coordenadas



(c) Rotação de 45º segundo o eixo de coordenadas X



(d) Rotação de 45º segundo o eixo de coordenadas Y



(e) Rotação de 45º segundo o eixo de coordenadas Z

Figura 5.6: Orientação de um objecto de acordo com a sua rotação nos diferentes eixos de coordenadas.

objecto, é necessário que todos os valores anteriores (posição e rotação) estejam preenchidos, mesmo que sejam indicados com o valor 0. Na linha:

```
redTeamRamp 17.5 0.0 1.5 0.0 0.0 180.0
```

o objecto *redTeamRamp* é criado na posição (17.5, 0, 1.5), com 180º de rotação relativamente ao eixo de coordenadas Z. Para assinalar o valor da rotação na coordenada Z é necessário indicar os valores de rotação das coordenadas X e Y como sendo 0.

Por outro lado se se pretender colocar um objecto na posição (0, 0, 0), sem rotação e sem parâmetro extra, basta indicar o nome do objecto nessa linha - todos os valores são assumidos como 0 por defeito. Assim, na seguinte linha:

```
blueTeamRing -1.0 2.5
```

o objecto *blueTeamRing* é criado na posição (-1, 2.5, 0) com a orientação original (sem rotação), pois todos os valores após o último inserido são considerados com o valor 0.

5.4 Comunicação

No contexto do desenvolvimento de jogos, a comunicação entre os elementos de jogo é fundamental, na medida em que permite trocar e partilhar informação e registar e avaliar os dados de jogo. A LSL fornece funções e eventos que permitem que os *scripts*

comuniquem com utilizadores, com outros *scripts* e com *scripts* e programas que corram em servidores externos ao mundo virtual. Dependendo do que se pretende realizar, existem aspectos que é necessário ter em conta na escolha do método a utilizar para realizar a comunicação. Assim, é possível analisar as possibilidades de comunicação na tabela 5.2, considerando os principais métodos e as finalidades pretendidas na realização deste protótipo.

Tabela 5.2: Análise dos métodos de comunicação da linguagem LSL [Cat10].

Método	**Outros utilizadores**	**Outros objectos**	***Scripts* no mesmo objecto**	**Enviar para servidor externo**	**Receber de servidor externo**	**Notas**
Chat: llWhisper, llSay, llShout	Sim	Sim	Não	Não	Não	É necessário estar à distância mínima
Chat: llRegionSay	Não	Sim	Não	Não	Não	Alcance da região; não funciona no canal de chat 0
LLDialog: Criar	Sim	Não	Não	Não	Não	Apenas o utilizador que solicitou o diálogo pode receber
LLDialog: Resposta	Sim	Sim	Não	Não	Não	É necessário estar à distância mínima de chat do local de criação do diálogo
Instant Message	Sim	Não	Não	Não	Não	É necessário estar à distância mínima de chat do local de criação do diálogo
Link Message	Não	Não	Sim	Não	Não	Apenas *scripts* contidos no mesmo objecto podem receber
HTTP	Não	Não	Não	Sim	Sim	Apenas podem ser iniciadas conexões com servidores externos; é necessário existir um *script* no servidor para processar o pedido

Como já referido, a comunicação com os ficheiros externos é feita através da utilização do protocolo HTTP que a LSL oferece. Apesar de existirem outras possibilidade de comunicação com servidores externos (não presentes na tabela 5.2), a utilização de HTTP representa a forma mais flexível e eficaz de comunicar, já que não apresenta *delay* nas suas execuções e tem capacidade tanto para enviar como para receber informação.

Quanto à comunicação entre elementos e objectos do jogo foram utilizadas diferentes alternativas, dependendo do contexto e da utilização necessária, já que cada função disponível para comunicação oferece diferentes propriedades e possibilidades.

A utilização do chat no Second Life é feita através de canais: existem 4 294 967 294 canais possíveis de ser utilizados, no intervalo de -2 147 483 648 a 2 147 483 647. O canal 0 é o canal público que surge na janela de chat e que permite aos utilizadores comunicarem entre si. A utilização dos restantes canais permite recorrer a uma comunicação segura entre objectos sem que as mensagens sejam intersetadas ou adulteradas por outros utilizadores. Assim, foram definidos canais para cada tipo de comunicação:

- 2001 - comando de apagar por parte do dono dos objectos;
- 2002 - comando de apagar por parte do objecto *HTTPCommunication*;
- 2003 - comunicação interna de *HTTPCommunication* para resolução do llDialog;
- 2004 - comando de começar/terminar jogo por parte do objecto de início de jogo *startGame*;
- 2005 - envio de informação no decorrer do jogo ao objecto *startGame*.

Estes canais são utilizados para garantir, por um lado que a informação seja transmitida apenas por *scripts* que tenham conhecimentos do canal a utilizar e por outro lado que não se escute e receba informação desnecessária, através do filtro que é feito ao escutar os canais pretendidos.

A função padrão para comunicação é:

```
llSay(integer channel, string text)
```

que transmite no canal *channel* a mensagem *text*. No entanto esta função apresenta limitações, nomeadamente ao nível do seu alcance e possibilidade de chegar a objectos distantes, já que uma chamada a esta função apenas é escutada num raio de 20 metros virtuais. Assim, existem alternativas, cujo alcance está representado na tabela 5.3.

Tabela 5.3: Alcance das funções de chat [Cat08].

Função	**Raio de alcance**
llWhisper	10 m
llSay	20 m
llShout	100 m
llRegionSay	Toda a região

Considerando o alcance de cada função, é utilizada a função llShout para realizar comunicação entre os diferentes objectos presentes no jogo. A utilização desta função permite garantir que todos os objectos se encontram no raio de alcance e conseguem receber qualquer mensagem proveniente do espaço virtual onde se encontra o nível. Assim, o objecto *HTTPCommunication* utiliza esta função para comunicar a opção de apagar a cena a todos os objectos através do canal 2002; o objecto *startGame* recorre também a

llShout para informar os objectos do início e do fim do jogo, através do canal 2004; todos os objectos respondem através do canal 2005 a *startGame* as alterações e toda a informação do decorrer do jogo, através de llShout.

A comunicação com os utilizadores é realizada através do canal 0, no caso de se pretender passar informação a todos os utilizadores (como o início/fim de jogo ou a entrada de um jogador para uma equipa), ou através de diálogos, no casa de se pretender obter a opção de um utilizador (p.e. a opção de criar ou destruir os níveis de jogo por parte do supervisor). A utilização de diálogos é feita através da função:

```
llDialog(key id, string message, list buttons, integer chat_channel)
```

Esta função permite comunicar com um utilizador através de uma caixa *popup* de diálogo que apresenta determinadas opções para o utilizador seleccionar. A escolha de um botão de opção faz com que o avatar correspondente responda através do canal *chat_channel*. Assim, novamente para evitar que a mensagem de resposta seja intersetada e/ou alterada, é utilizado um canal próprio de comunicação - o objecto *HTTPCommunication* utiliza o canal 2003 para resolução da resposta a llDialog.

5.4.1 Interacção entre Objectos

Todos os objectos existentes no mundo virtual cuja finalidade é serem utilizados na construção dos níveis de jogo e depois eliminados, de forma a ser possível reutilizar o espaço virtual, têm em comum código LSL que permite receber informação do objecto *HTTPCommunication*, para a sua destruição:

```
string listen_name = "HTTPCommunication";

default
{
  state_entry()
  {
     llListen(2001,"",llGetOwner(),"");
     llListen(2002,listen_name, "","");
  }
  (...)
```

Esta secção de código permite que todos os *scripts* filtrem os canais definidos para a passagem de informação acerca da eliminação dos objectos (2001 e 2002). Além disso, para garantir que apenas recebem as mensagens pretendidas, todas as mensagens passadas por esses canais são filtradas quanto ao remetente.

Quando é dada uma ordem para eliminar a cena, é enviada uma mensagem de “delete” que activa o seguinte *listener*:

```
listen(integer channel, string name, key id, string message)
{
   (...)
   if(channel == 2002 && name == listen_name)
   {
     if((message == "delete") ||
      (message == "delete " + llGetObjectName()))
        llDie();
   }
}
```

Desta forma, quando *HTTPCommunication* envia a mensagem "delete" através do canal 2002, os objectos executam a função *llDie()* que os elimina e faz desaparecer do mundo virtual.

Existem também os objectos de jogo que têm algum tipo de interacção no decorrer do jogo, pelo que necessitam de saber quando é iniciado o jogo e quando é terminado. Esta informação é emitida pelo objecto *startGame*, que envia a mensagem "START_GAME" através do canal 2004:

```
listen(integer channel, string name, key id, string message)
{
   (...)
   if(channel == 2004)
   {
     if((message == "START_GAME"))
     {
       (...)
       llSetText("Game running", <0,0,1>, 1);
       state game;
     }
   }
}
```

Quando estes objectos recebem esta mensagem de início de jogo entram no estado *game*, onde executam as acções correspondentes (criação de objectos, registo de pontuações, etc.) necessárias no decorrer do jogo. O final do tempo de jogo é comunicado no mesmo canal (2004), através da mensagem "END_GAME".

Em qualquer altura um objecto de jogo pode comunicar informação importante, como a entrada ou saída de um jogador de uma equipa, ou as pontuações obtidas por cada equipa, ao objecto *startGame*. Esta comunicação é efectuada através do canal 2005, que *startGame* escuta, recebe a informação e guarda-a.

5.4.2 Informação de Jogo

O objecto *startGame*, que é o responsável por controlar o decorrer do jogo, guarda toda a informação, desde a duração das sessões de jogo e as horas em que iniciaram e terminaram, até à informação que recebe de todos os objectos de jogo através do canal 2005. No final de cada jogo, este objecto entra num estado de comunicação (*sending*) em que se encarrega de registar toda a informação num ficheiro externo através de um pedido HTTP.

```
state sending
{
  state_entry()
  {
    (...)
    llListen(2005,"","","");
    //5 seconds pause waiting for messages
    llSetTimerEvent(5.0);
  }
  listen(integer channel, string name, key id, string message)
  {
  if(channel = 2005)
    {
      //save game info
      info += message + "\n";
    }
  }
  timer()
  {
    send_info();
    (...)
  }
}
```

Após uma pausa de 5 segundos em que o *script* aguarda por uma eventual comunicação de qualquer objecto de jogo, é chamada a função send_info() que efectua o seguinte pedido:

```
request = llHTTPRequest(http_request_url,
                        [HTTP_METHOD, "POST",(...)],
                        "message="+messageURL);
```

em que *http_request_url* corresponde ao *url* de um *script* PHP (*receiveinfo.php*). A informação chega a este código PHP, que recebe o pedido de POST e imprime no ficheiro *gameinfo* o conteúdo de "message".

O ficheiro *gameinfo* fica assim com o registo dos jogos e funciona como um *log* de toda a informação acerca das sessões de jogo efectuadas. Este registo tem o seguinte formato:

```
MESSAGE START
redTeamRing: Test User with red team
blueTeamRing: Test User3 with blue team
blueTeamRing: Test User2 with blue team
redTeamRing: Test User1 with red team
Game start: 2011-05-19T13:38:32
blueTeamPlatform 1
blueTeamPlatform 2
redTeamPlatform 1
blueTeamPlatform 3
redTeamPlatform 2
redTeamPlatform 3
blueTeamPlatform 4
redTeamPlatform 4
redTeamPlatform 5
redTeamPlatform 6
blueTeamPlatform 5
redTeamPlatform 7
Game end: 2011-05-19T13:39:33
MESSAGE END
```

É possível analisar a informação contida no ficheiro, para perceber qual a equipa que teve a melhor prestação no jogo. Assim, no exemplo de jogo anterior, os jogadores *Test User* e *Test User1* pertenciam à equipa *red* e conseguiram 7 pontos, enquanto que os jogadores *Test User2* e *Test User3* pertenciam à equipa *blue* e conseguiram 5 pontos. O jogo teve uma duração de 1 minuto e teve início às 13h38 do dia 19/05/2011.

5.5 Conclusão

Neste capítulo foi apresentado o trabalho realizado a nível da implementação para o desenvolvimento do protótipo de uma plataforma de modelação expedita de níveis de jogo para utilização em ambientes virtuais.

A arquitectura assenta na metodologia cliente/servidor, em que os utilizadores (supervisor e jogadores) são os clientes que interagem com o mundo virtual, controlado pelo servidor. O supervisor é responsável por definir a estrutura dos níveis que serão modelados no mundo virtual.

As plataformas Second Life e OpenSimulator permitem a utilização de um ambiente virtual 3D multi-utilizador para construção e modelação de objectos virtuais, com possibilidade de utilizar a linguagem LSL para controlar o comportamento dos objectos, o decorrer do jogo e a comunicação.

Foram também especificados os processos para a construção dos níveis de jogo, passando pela modelação dos objecto virtuais, até à definição dos níveis através de um ficheiro externo. É utilizada uma linguagem simples de definição que permite especificar a posição e a orientação dos objectos no espaço virtual.

Por fim, foram analisados os métodos de comunicação e de interacção entre os diferentes elementos - jogadores, objectos virtuais e ficheiros externos, para controlo das sessões de jogo, transmissão de informação e registo dos dados de jogo.

Capítulo 6

Resultados

Esta ferramenta foi desenvolvida com o intuito de ter disponível uma plataforma de modelação de níveis de jogo num ambiente virtual que permitam ajudar na dinamização de grupos de trabalho, ajudando na integração através da comunicação e interacção entre os elementos, ao mesmo tempo que facilita e rentabiliza a utilização de espaços virtuais através da modelação expedita do conteúdo, utilizado nos ambientes virtuais 3D multi-utilizador Second Life e OpenSimulator.

Para ser possível verificar as vantagens desta plataforma e a viabilidade da sua utilização futura numa situação real (como a sua utilização na unidade curricular Projecto FEUP ou noutra situação de ensino) é necessário proceder a uma série de testes e comparações relativamente a projectos no mesmo âmbito. Assim, tendo em conta que os níveis de jogo realizados neste protótipo são baseados no projecto FEUP Adventure, este constituirá o caso de estudo para a análise desta plataforma que é realizada neste capítulo. Esta escolha deve-se ao facto de a utilização do projecto FEUP Adventure em sessões de teste com estudantes do 1º ano do ensino superior ter permitido verificar a sua capacidade de estimular a integração e de ter obtido um *feedback* positivo por parte dos estudantes que participaram nas sessões [CSC11]. É também importante referir que, apesar da recriação dos níveis do FEUP Adventure, esta plataforma de criação de níveis virtuais não está limitada à mecânica desse projecto, sendo possível a construção dos mais variados níveis de jogo e diferentes mecânicas, de acordo com a disposição efectuada na definição do nível dos diferentes elementos de jogo existentes. O protótipo implementado foi validado com os 2 primeiros níveis de jogo do FEUP Adventure que serviu de motivação inicial para este trabalho.

É também analisada a utilização dos ambientes virtuais Second Life e OpenSimulator e comparados os resultados obtidos em ambos, para perceber as diferenças, vantagens e limitações das suas utilizações como base para a plataforma desenvolvida.

6.1 Criação de um Nível

A criação de um nível é efectuada através da interacção com o objecto inicial de construção, a semente (apelidada de *HTTPCommunication*). Quando se "toca" em *HTTPCommunication* este apresenta um menu com as opções de criar um novo nível ou de apagar os objectos existentes (figura 6.1).

Figura 6.1: Menu de criação/eliminação do nível de jogo.

Ao seleccionar a opção de criar um novo nível, a semente começa a criação da cena (após comunicar com *gamedefinition*). Para cada linha correspondente a um objecto nas regras definidas em *gamedefinition*, a semente desloca-se para a posição de destino desse objecto e cria uma cópia do objecto no mundo virtual, através do seu inventário. Assim, todos os objectos são criados sucessivamente até toda a cena estar criada no ambiente virtual, como é possível ver na figura 6.2.

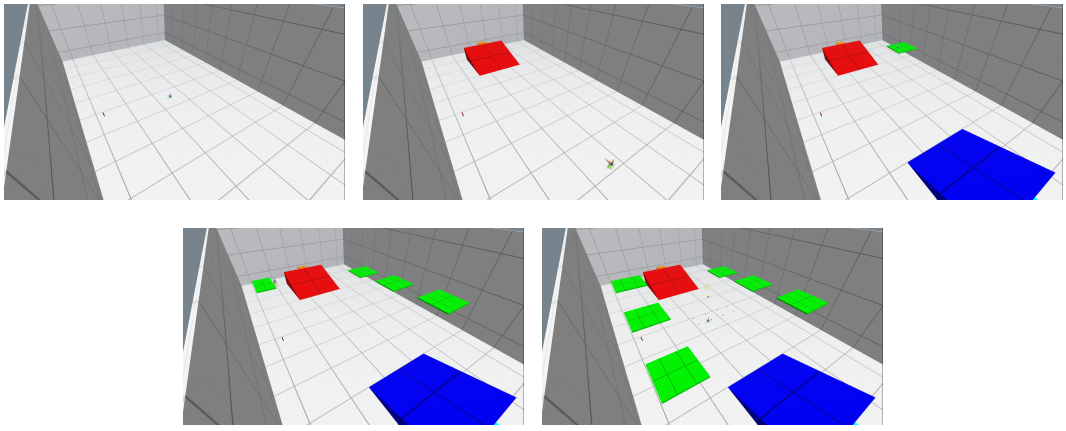
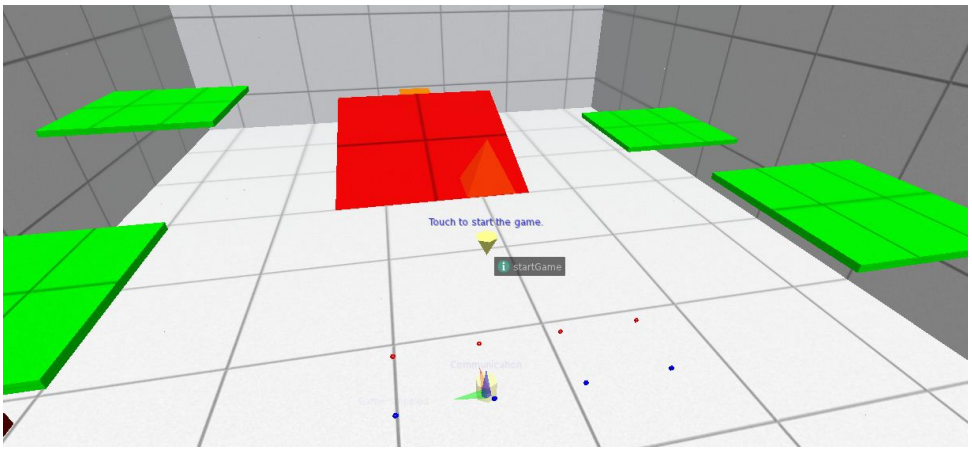


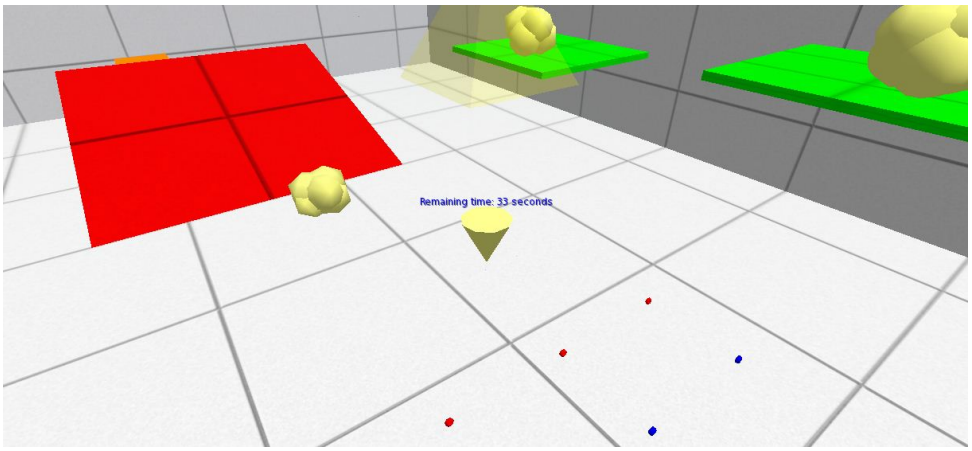
Figura 6.2: Criação procedimental de um nível.

Todos os objectos criados neste protótipo encontram-se a postos para executar o jogo logo após a sua criação. De forma a controlar o início e fim das sessões de jogo e a sua duração, bem como guardar a informação das equipas e dos seus resultados é necessário

garantir que o objecto *startGame* é criado. Assim, neste protótipo a sua utilização é sempre feita no final da definição das regras de construção dos níveis, colocado no centro da área de jogo, alguns metros acima da chão (figura 6.3a). Foi criado como um cone invertido, que é possível ser utilizado pelo supervisor para colocar o seu avatar (sentado) enquanto decorre o jogo. O jogo começa quando o supervisor "toca" no objecto e durante o seu decorrer é apresentado o tempo restante até ao final (figura 6.3b).



(a) *startGame* criado no centro da área de jogo



(b) Jogo a decorrer

Figura 6.3: Objecto de controlo de jogo *startGame*.

Para ser possível alocar os jogadores às equipas foram criados "anéis" de identificação que são atribuídos aos jogadores. Estes anéis devem ser utilizados pelos jogadores durante o decorrer do jogo de forma ser possível identificar a sua equipa, através da sua cor. O número e colocação dos anéis no ambiente de jogo é definida em *gamedefinition*. É possível ver na figura 6.4 a colocação dos anéis no centro de jogo, de forma a ser possível a sua atribuição aos jogadores.

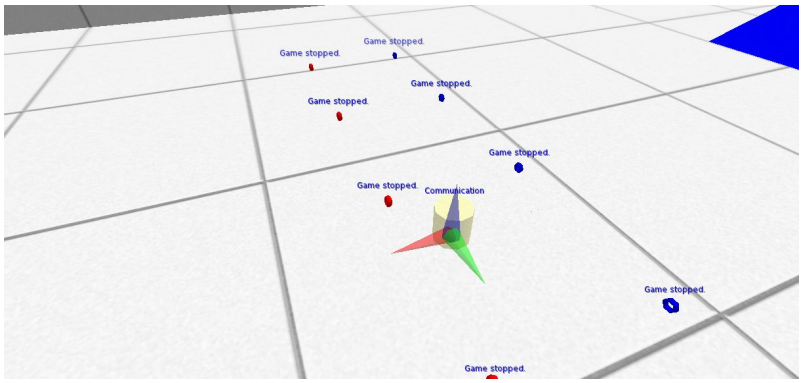


Figura 6.4: Objectos de identificação dos jogadores criados no centro de jogo.

6.2 Performance da Plataforma

De forma a poder avaliar a performance do mecanismo de modelação expedita foi criado um *script* de *debug* que guarda a informação acerca do tempo de carregamento dos objectos e da distância percorrida pela semente para a sua criação no ambiente virtual. Desta forma é possível verificar o tempo médio de criação de um nível, relativamente ao número de objectos e à distância entre eles. Os dados obtidos na reprodução do nível 1 do projecto FEUP Adventure (apresentado em 6.3.1) são os seguintes:

```
1: 0.000000-DEBUG START-2011-06-02T10:38:46.809150Z
2: 0.089844-HTTP RESPONSE
3: 0.765625 (+0.765625)-CREATED: redTeamRamp-17.564169
4: 1.207031 (+0.441406)-CREATED: redTeamPlatform-6.139219
5: 2.542969 (+1.335938)-CREATED: blueTeamRamp-41.020607
6: 2.976562 (+0.433594)-CREATED: blueTeamPlatform-6.139219
7: 4.082031 (+1.105469)-CREATED: platform-35.280163
8: 4.507812 (+0.425781)-CREATED: platform-2.000000
9: 5.152344 (+0.644531)-CREATED: platform-10.246951
10: 5.554688 (+0.402344)-CREATED: platform-2.000000
11: 6.199219 (+0.644531)-CREATED: platform-10.246951
12: 6.621094 (+0.421875)-CREATED: platform-2.000000
13: 7.488281 (+0.867188)-CREATED: platform-28.284271
14: 7.937500 (+0.449219)-CREATED: platform-2.000000
15: 8.585938 (+0.648438)-CREATED: platform-10.246951
16: 9.011719 (+0.425781)-CREATED: platform-2.000000
17: 9.707031 (+0.695312)-CREATED: platform-10.246951
18: 10.105469 (+0.398438)-CREATED: platform-2.000000
19: 10.777344 (+0.671875)-CREATED: blueTeamRing-16.911535
20: 11.175781 (+0.398438)-CREATED: blueTeamRing-2.500000
21: 11.578125 (+0.402344)-CREATED: blueTeamRing-2.500000
22: 11.976562 (+0.398438)-CREATED: blueTeamRing-2.500000
23: 12.378906 (+0.402344)-CREATED: redTeamRing-7.762087
```

```
24: 12.871094 (+0.492188)-CREATED: redTeamRing-2.500000
25: 13.296875 (+0.425781)-CREATED: redTeamRing-2.500000
26: 13.742188 (+0.445312)-CREATED: redTeamRing-2.500000
27: 14.144531 (+0.402344)-CREATED: createRoundCubes-6.576473
28: 14.578125 (+0.433594)-CREATED: startGame-2.000000
29: 14.839844 (+0.261719)-INITIAL POSITION-5.000000
30: 14.886719-DEBUG END-2011-06-02T10:39:01.693817Z
```

Cada linha contém a seguinte informação:

```
<tempo total)> <(+tempo da operação)> - <operação> - <distância percorrida>
```

em que os tempos apresentados (total e da operação) são em segundos e a distância percorrida é em metros do mundo virtual. Analisando os dados é possível concluir que toda a operação de criação do nível demorou aproximadamente 15 segundos, tendo sido criados 26 objectos. No gráfico da figura 6.5 é possível ver a relação entre o tempo de carregamento de cada objecto com a distância que a semente teve que percorrer para a sua criação.

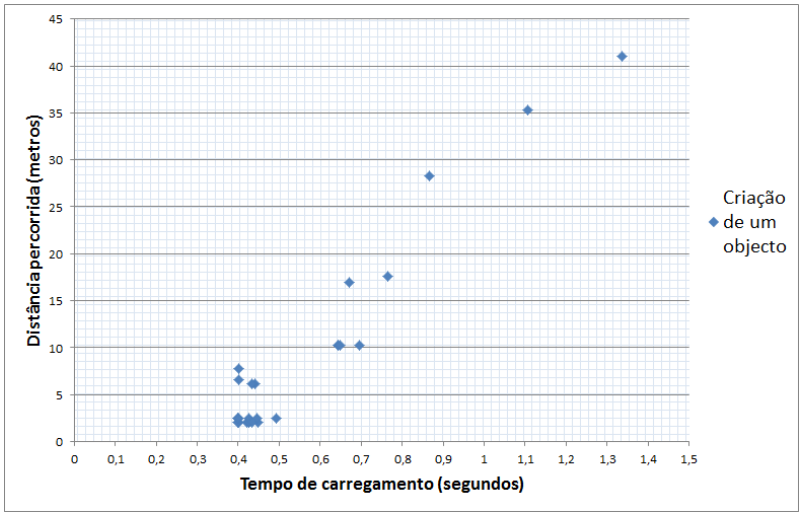


Figura 6.5: Relação entre o tempo e a distância gastos na criação dos objectos do nível 1.

É possível concluir que o tempo necessário para criar cada objecto aumenta com a distância que é necessário percorrer para colocar o objecto na sua posição.

Para avaliar esta relação foi criado um nível de teste com 40 plataformas com distâncias crescentes entre elas de 1 metro. O processo demorou 31 segundos, tendo-se obtido o gráfico da figura 6.6 com a relação tempo-distância de criação de cada objecto.

Verifica-se que o tempo de criação de cada objecto aumenta a cada 10 metros de distância: isto deve-se ao processo de movimentação de *HTTPCommunication* no espaço virtual. É também possível verificar que, considerando objectos que envolvem uma deslocação até 40 metros, o tempo de criação de cada objecto é inferior a 1,5 segundos. Assim, para se obter o melhor rendimento ao nível do tempo de criação da cena virtual,

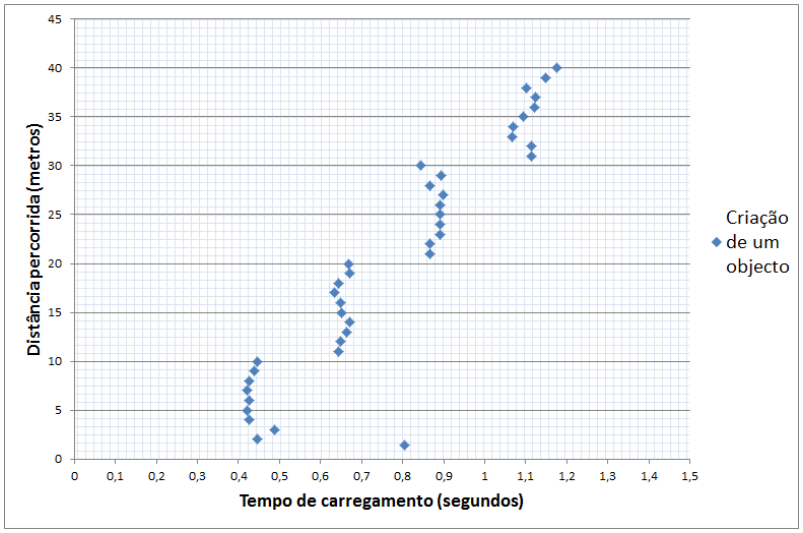


Figura 6.6: Relação entre o tempo e a distância gastos na criação dos objectos de um nível de teste.

a definição do *layout* deve ser pensada de forma a que a ordem de criação dos objectos diminua a deslocação de *HTTPCommunication*. Através de uma boa ordem de colocação dos objectos em *gamedefinition* (distâncias inferiores a 10 metros) é possível criar uma cena completa com todos os objectos a serem criados com intervalos de aproximadamente 0,5 segundos (como é verificável na figura 6.6).

6.3 Caso de Estudo: FEUP Adventure

Para o teste da plataforma de desenvolvimento de níveis de jogo foram reproduzidos os níveis 1 e 2 existentes no projecto FEUP Adventure. Este projecto permite estimular a integração e interacção entre os estudantes, através de de um conjunto de níveis de jogo virtuais que necessitam de ser jogados em equipa. Estes níveis colaborativos representam objectivos bastante simples e específicos para a sua realização por parte dos estudantes, através de estratégias de coordenação e cooperação entre os jogadores de uma equipa. Os níveis são baseados na física, na movimentação de objectos e na orientação do avatar de cada jogador. Assim, para perceber a mecânica de cada nível e o tipo de interacção que exige dos jogadores é feita nos capítulos seguintes a descrição dos níveis recriados neste projecto, enquanto se procede à verificação do cumprimento dos objectivos no desenvolvimento deste caso de estudo.

6.3.1 Nível 1

O primeiro nível consiste num jogo onde o objectivo de cada equipa é conseguir pontuar o máximo possível, durante um tempo limitado (por exemplo 5 minutos). Cada

equipa pontua transportando "blocos" que se encontram espalhados pelo cenário através da sua rampa e largando-os na plataforma móvel que se encontra no topo da rampa correspondente. Na figura 6.7 está representado o mapa do nível 1: para garantir que o jogo é equilibrado para ambas as equipas, as zonas de jogo são simétricas e cada jogador começa o jogo no topo da rampa correspondente à sua equipa.

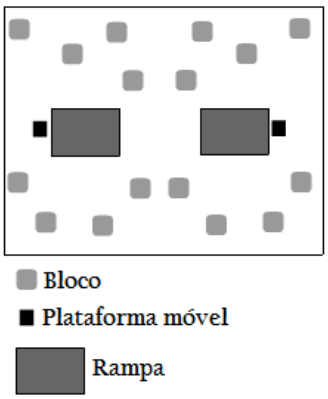


Figura 6.7: Mapa do nível 1 do FEUP Adventure [CSC10].

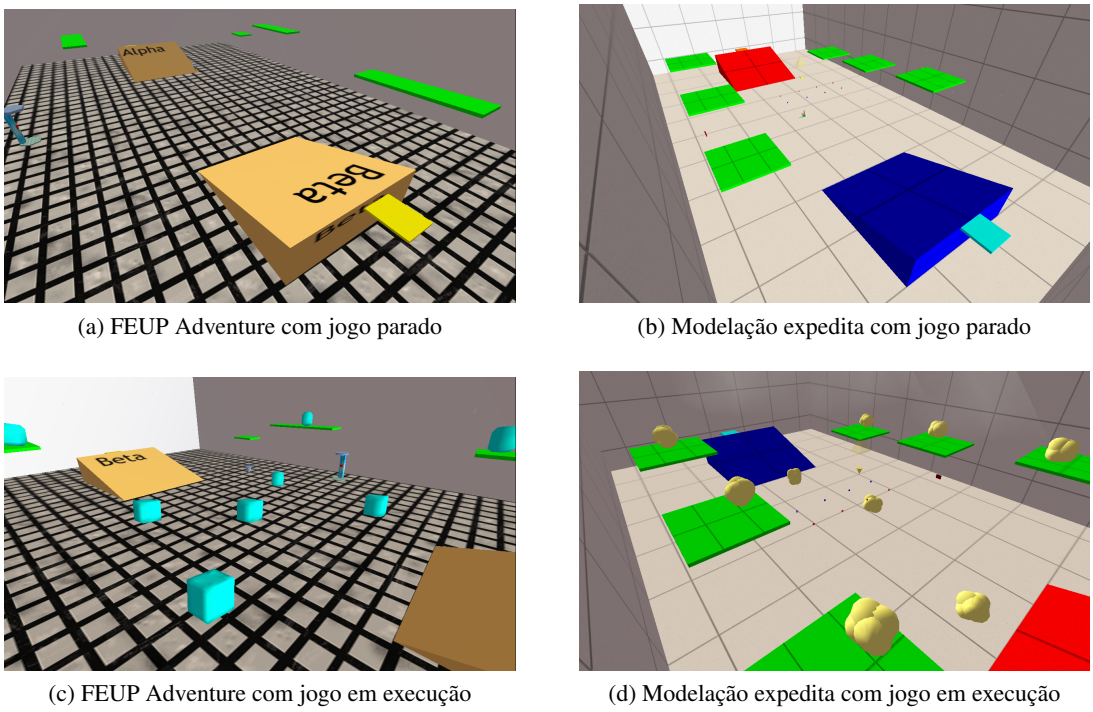
Cada jogador apenas pode aceder à rampa da sua equipa, sendo aplicado um "empurrão"[1] a cada jogador que entre em contacto com a rampa da equipa adversária.

Para a reprodução deste nível neste projecto foram recriados todos os objectos necessários para a criação do cenário, recriando as suas propriedades fundamentais (dimensões, orientação, material, etc) que garantem que a experiência de jogo seja igual nas duas versões (figura 6.8).

De forma a garantir um melhor controlo sobre o número de blocos no cenário foi criado um objecto responsável pela sua criação e distribuição no nível. No início do jogo ele distribui automaticamente os blocos pelo cenário e durante o decorrer do jogo o supervisor da sessão pode requerer nova distribuição de blocos - no projecto FEUP Adventure os blocos são criados uns em relação aos outros, com o primeiro bloco criado a criar o segundo, este a criar o seguinte, etc.

Fazendo uma comparação directa à prestação dos elementos presentes no nível em ambos os projectos, é possível verificar melhorias a nível do seu comportamento, devido ao facto de a sua actividade ser controlada através de um objecto principal (*startGame*). Assim, a criação de blocos é feita apenas no decorrer do jogo, tendo uma distribuição mais simétrica relativamente às rampas das equipas; por outro lado, como são criados através de um objecto próprio, e não uns através dos outros, existe apenas um *script* responsável a ser executado, o que permite melhor performance na totalidade de *scripts* em execução. Também as plataformas de cada equipa apenas estão em movimento durante o decorrer do

[1] Esta acção é realizada através de uma função da linguagem LSL (*llPushObject*), que aplica um impulso em determinado objecto ou avatar



(a) FEUP Adventure com jogo parado

(b) Modelação expedita com jogo parado

(c) FEUP Adventure com jogo em execução

(d) Modelação expedita com jogo em execução

Figura 6.8: Comparação da construção do nível 1 do FEUP Adventure com a construção através de modelação expedita.

jogo e os jogadores só são expulsos da rampa da equipa adversária neste período. Desta forma, enquanto o jogo não está a decorrer, todos os elementos de jogo encontram-se em "pausa", sem consumir demasiados recursos a nível de execução de *scripts* e a nível de *rendering* de objectos na cena virtual.

6.3.2 Nível 2

O segundo nível de jogo também opõe duas equipas com o objectivo de pontuar o máximo possível num tempo limitado. Existem esferas que se movimentam ao longo de uma rampa e vão de encontro a um bloco que as faz desaparecer. Cada jogador deve tentar desviar a rota das esferas de forma a que estas não vão contra o bloco e atinjam a parede correspondente à sua equipa situada atrás do bloco, que lhe dará pontos. Existem duas rampas, correspondentes às paredes "objectivo" de cada equipa. As equipas devem, por um lado tentar que as esferas atinjam a sua parede, e por outro evitar que atinjam a parede da equipa adversária na outra rampa. Para tal, todos os jogadores têm acesso a toda a área de jogo (mapa da figura 6.9), de forma a ser possível aplicar a estratégia pretendida (tentar pontuar ou tentar evitar que a outra equipa pontue).

Da mesma forma que no nível 1, foi utilizado o objecto *startGame* que controla o decorrer do jogo e apenas neste período é que as esferas são criadas. Os elementos presentes foram também recriados de forma a garantir a mesma experiência de jogo nas duas

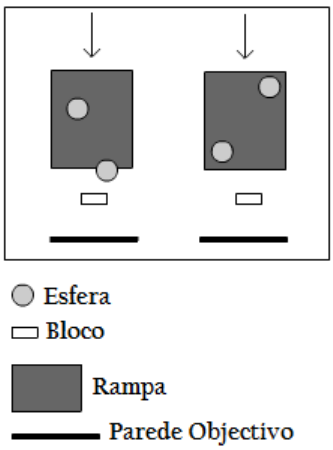
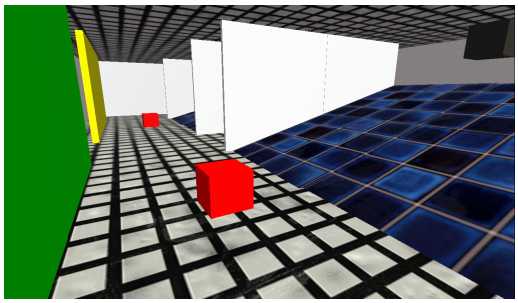
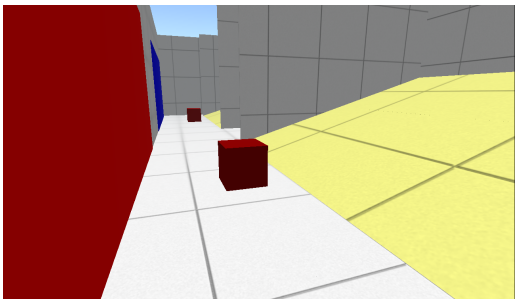


Figura 6.9: Mapa do nível 2 do FEUP Adventure [CSC10].

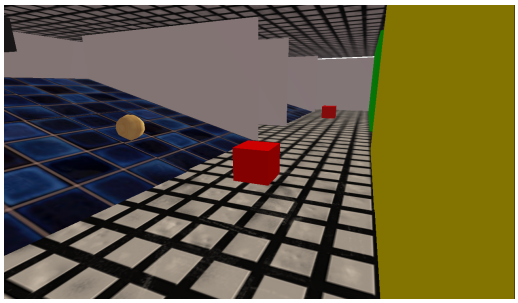
versões (figura 6.10). Assim, verifica-se que a utilização desta plataforma não prejudica nem deteriora a criação dos níveis de jogo e é possível recriar na totalidade a mecânica de jogo presente no projecto FEUP Adventure.



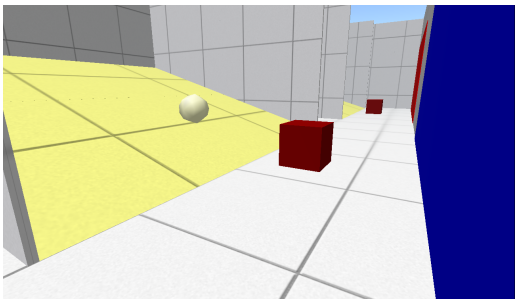
(a) FEUP Adventure com jogo parado



(b) Modelação expedita com jogo parado



(c) FEUP Adventure com jogo em execução



(d) Modelação expedita com jogo em execução

Figura 6.10: Comparação da construção do nível 2 do FEUP Adventure com a construção através de modelação expedita.

Na reprodução do nível 2 nesta plataforma são criados 39 objectos, tendo um tempo médio de criação de 21 segundos, através da modelação expedita. Este tempo deve-se essencialmente à forma como a construção dos objectos se encontra estruturada relativamente à ordem pela qual são desenhados no espaço virtual: os objectos são criados de

forma a evitar uma grande deslocação da semente, tendo um tempo médio de 0,5 segundos por objecto, no entanto existem duas grandes áreas de jogo que se estendem por toda a área de jogo e que implicam uma maior deslocação para a criação de alguns objectos, levando a um maior tempo (como se verificou no capítulo 6.2).

Uma vantagem fundamental presente na construção dos níveis através do mecanismo de modelação expedita desta plataforma diz respeito à possibilidade de estabelecer no ficheiro de definição do nível alguns parâmetros que não estão à partida disponíveis, como o tempo de jogo que o supervisor pretende para a sessão ou o intervalo de tempo com que as esferas são construídas. Além disso, o número de elementos no nível é facilmente editável, podendo o supervisor adicionar ou retirar ,por exemplo, plataformas fixas em qualquer posição do ambiente virtual. Estas alterações de outra forma apenas seriam possíveis alterando directamente o código LSL dos *scripts* dos objectos ou as suas propriedades internas (posição no espaço, orientação, etc.). Considerando que na utilização do Second Life, e no caso do projecto FEUP Adventure, os objectos não são editáveis, não é possível a um supervisor efectuar estas alterações sem que seja dono dos respectivos objectos.

Além disso, a possibilidade de criar e eliminar os diferentes níveis permite maior liberdade na utilização do espaço virtual, já que a utilização de um objecto virtual no projecto FEUP Adventure requer que ele seja permanente e inalterado, sendo que qualquer eliminação de elementos no espaço virtual é definitiva e só é recuperado através da sua criação manual.

É possível consultar a definição completa de ambos os níveis 1 e 2 utilizada para recriação deste caso de estudo no anexo A.

6.4 Ambiente Virtual

O desenvolvimento de um jogo num ambiente virtual pode consistir numa tarefa diferente se comparada com a utilização de um *game engine* com um SDK adequado ao desenvolvimento de jogos. Apresenta limitações ao nível da utilização de uma linguagem não orientada a objectos e processos restritos de interacção para os utilizadores [CSC11], nomeadamente o controlo da interacção física entre os avatares e os diferentes elementos virtuais (esferas, cubos, etc). Também a utilização de excertos de filmes ou de personagens não-jogáveis (*non-player character* ou NPC) animadas, habitualmente utilizados em jogos digitais, é fortemente limitada nestes ambientes pré-estabelecidos - a utilização de um NPC é habitualmente feita através de um objecto estático que interage com os jogadores através de caixas de texto [CSC11].

Porém, é possível tirar partido das suas vantagens e tornar o ambiente virtual numa

plataforma atractiva para o desenvolvimento de determinados tipos de jogo, nomeadamente jogos que apresentem uma mecânica simples bem definida e em que seja valorizada a experiência colaborativa dos seus jogadores em detrimento de gráficos e física elaborada. A sua facilidade de utilização, as suas funcionalidades e a sua acessibilidade permitem que um ambiente virtual como o Second Life seja uma plataforma ideal para o desenvolvimento de jogos sérios, onde as equipas são habitualmente compostas por poucos elementos e apresentam objectivos simples e rápidos [CSC11].

OpenSimulator

A utilização do OpenSimulator permitiu verificar a sua potencialidade para ser utilizado como alternativa ao Second Life, no âmbito da utilização de um ambiente virtual 3D multi-utilizador. Desde logo é importante referir que a plataforma foi implementada em modo *standalone*, pelo que as conclusões retiradas não permitem verificar a sua total potencialidade e capacidade. No entanto, foram efectuados alguns testes no modo *grid*, em *sandboxes* públicas, para avaliar o comportamento da física e ser possível comparar com o modo *standalone*.

Apresenta pontos positivos a nível da capacidade de criação e gestão de uma região virtual e a nível do controlo sobre os acessos dos avatares. A criação dos objectos virtuais é semelhante em ambos os ambientes virtuais, e a execução da plataforma de modelação expedita tem o mesmo comportamento tanto no Second Life como no OpenSimulator, não se verificando diferenças tanto a nível da reprodução dos elementos no espaço virtual como a nível de comunicação com os ficheiros externos.

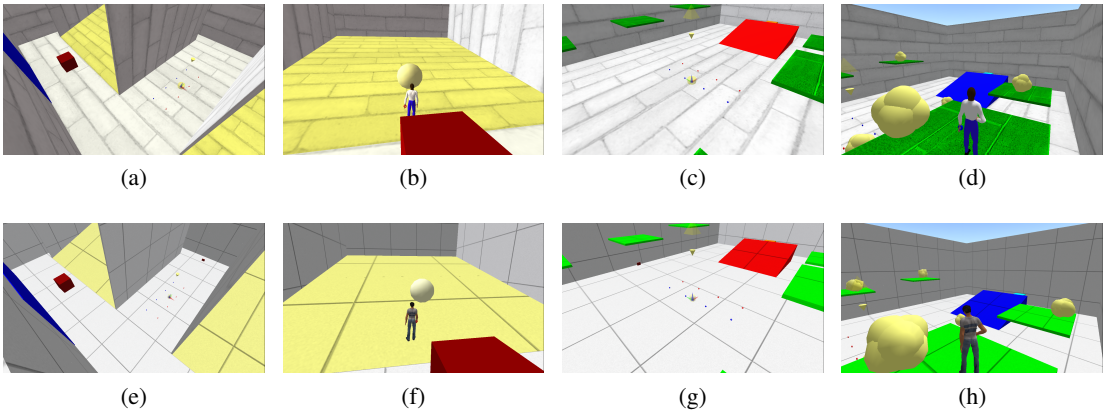
No entanto, apresenta algumas diferenças que o tornam limitado, nomeadamente no seu controlo da física, no controlo do avatar e na interacção com objectos. A física presente é um pouco mais fraca e limitada, quando comparada ao Second Life. Verifica-se que o contacto entre o avatar e os objectos virtuais não é intuitivo nem permite ter uma correcta sensação da física, ocorrendo por vezes atrasos e incorrecções na apresentação das imagens do mundo virtual, com o avatar do jogador a atravessar objectos virtuais ou objectos a não responderem imediatamente ao contacto. O comportamento dos objectos não corresponde ao mesmo obtido no Second Life, nomeadamente quando se trata de objectos complexos, formados por vários objectos básicos. A detecção de colisões por parte destes objectos não é efectuada correctamente por todos os seus elementos, e o mesmo acontece com os objectos que os avatares podem vestir. Isto constituiu uma limitação na utilização dos anéis de identificação dos jogadores neste protótipo, bem como no reconhecimento de jogadores em plataformas adversárias. Também não permite criação de grupos, o que impede que sejam criados grupos de estudantes, ou de diferentes sessões, por exemplo. Também a não existência de uma moeda corrente não permite a fácil transferência de objectos entre avatares que é possível no Second Life através da venda. Além disso, o facto de não suportar 100% da linguagem LSL pode originar algumas falhas

no comportamento dos *scripts* e dos respectivos objectos virtuais. Em geral, é possível constatar as suas vantagens e desvantagens na tabela 6.1.

Tabela 6.1: Vantagens e desvantagens da utilização do OpenSimulator [ACT09].

Vantagens	**Desvantagens**
Controlo sobre actualizações dos servidores	Menos estável que o Second Life
Sem restrição de idade	Necessidade de apoio técnico
Flexibilidade	Falta de algumas funcionalidades da LSL
Controlo dos custos	Documentação dispersa
Modular	Aplicação Beta
Open source	Ausência de inventário pré-feito
Escalabilidade	Falta de mercado
Controlo sobre acessos de utilizadores	Falta de uma grande comunidade
Arquivamento do inventário	Ninguém responsável pelo desenvolvimento
	A física é fraca

A utilização do OpenSimulator foi importante, essencialmente na fase inicial de implementação do protótipo. A possibilidade de ter uma região a executar *offline* permite ter uma primeira aproximação com a criação de objectos no espaço virtual bem como com a criação e teste de *scripts* LSL, sem incorrer em custos monetários de aquisição ou manutenção do espaço virtual, como acontece no Second Life, já que é possível obter todo o *software* necessário para executar uma região virtual local gratuitamente.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figura 6.11: Resultados obtidos no OpenSimulator (a, b, c e d) e no Second Life (e, f, g e h).

Na figura 6.11 é possível verificar as semelhanças de resultados na criação dos níveis virtuais no OpenSimulator e no Second Life, em que as diferenças visuais se devem à utilização de diferentes texturas. Também o tempo de execução e de criação dos objectos virtuais é, em média, igual nas duas plataformas.

6.5 Conclusão

Este capítulo teve como objectivo apresentar os resultados obtidos na realização desta dissertação, através da demonstração e teste de execuções do mecanismo de modelação expedita.

Assim, foi apresentada a criação de um nível e todo o processo de modelação dos objectos virtuais no espaço virtual. Foi também avaliada a performance da execução deste mecanismo, através da execução de um *script* de *debug*, que permite registar os tempos de criação dos vários elementos e avaliar a sua relação com a distância percorrida pela semente de construção na sua criação.

Tendo em conta os objectivos e a mecânica de jogo que o projecto FEUP Adventure apresenta, este foi utilizado como caso de estudo para teste deste protótipo. Como tal, foram reproduzidos os níveis deste projecto, com as implementações necessárias para a sua utilização com o mecanismo de modelação expedita e para permitir um melhor comportamento dos objectos virtuais e do controlo das sessões de jogo. Neste capítulo foi apresentada a utilização do caso de estudo, com comparação entre ambos os projectos e com a verificação das vantagens da utilização do mecanismo de modelação expedita.

Foi também analisada a utilização dos ambientes virtuais 3D, com avaliação da potencialidade de utilizar a plataforma *opensource* OpenSimulator como alternativa ao Second Life, que requer custos monetários para a manutenção do espaço virtual. Apresenta vantagens e desvantagens que devem ser tidas em conta aquando da decisão de utilizar um ambiente virtual 3D multi-utilizador.

Capítulo 7

Conclusões e Trabalho Futuro

Neste trabalho desenvolveu-se o conceito e uma metodologia para modelação expedita de níveis de um jogo sério para a dinamização de grupos, num ambiente virtual 3D multi-utilizador.

Inicialmente foi realizado um estudo do estado da arte ao nível da criação de jogos, tendo sido avaliados diferentes *game engines* e ambientes virtuais. Os jogos sérios foram também analisados, permitindo verificar a potencialidade que estes oferecem para motivar os seus utilizadores, o que possibilita a sua utilização na educação e na aprendizagem. Também os *Multi-user Virtual Environments* foram estudados, tendo-se verificado que permitem potenciar o processo de aprendizagem através da cooperação e competição entre os seus utilizadores. Projectos como o FEUP Adventure permitem verificar que a utilização de um jogo sério construído num ambiente virtual como o Second Life obtém bons resultados quando utilizado com estudantes e que possibilita a dinamização de grupos de trabalho, atráves da comunicação e interacção entre os jogadores.

No seguimento do projecto FEUP Adventure, e após a análise das diferentes ferramentas disponíveis para o desenvolvimento de jogos e das suas vantagens e desvantagens, decidiu-se utilizar um ambiente virtual multi-utilizador dado já dispor de todo o espaço virtual 3D pronto a ser utilizado. Os ambientes escolhidos, Second Life e OpenSimulator, permitem dispor de toda a estrutura cliente/servidor, com fácil acesso à criação de objectos no mundo virtual, e apresentam todo o mecanismo de autenticação e de criação de avatares bem estruturado e pronto a utilizar.

Considerando as limitações de espaço inerentes à utilização de um mundo virtual, e tendo em vista o objectivo desta dissertação de utilizar modelação de forma expedita que permita reutilizar o espaço disponível e acelerar o processo de criação de objectos, foi apresentada uma metodologia para criação alternada de níveis de jogo num mesmo espaço virtual. A abordagem consiste em criar todos os objectos necessários e guardá-los

no inventário de um objecto semente, que corresponde a um objecto inicial de construção. Na construção de um nível, a semente comunica com um ficheiro de forma a verificar a especificação do nível pretendida. Todo o nível é assim criado, com as propriedades, localização e orientação dos objectos definidas.

Foi assim desenvolvido um protótipo de uma plataforma de modelação expedita de níveis de jogo para utilização em ambientes virtuais, tendo sido apresentado e especificado o trabalho realizado a nível do seu desenvolvimento. A implementação desta plataforma passa pela construção e controlo dos objectos virtuais e do seu comportamento, bem como controlo do decorrer de jogo e da comunicação entre os diferentes elementos.

Tendo em conta os seus objectivos e a sua mecânica de jogo, o projecto FEUP Adventure constituiu um caso de estudo para a verificação da utilização da plataforma de modelação expedita, tendo sido reproduzidos com sucesso os seus níveis 1 e 2.

7.1 Satisfação dos Objectivos

Considerando os objectivos traçados para o desenvolvimento desta dissertação, é possível afirmar que todos eles foram cumpridos e satisfeitos na totalidade.

Foi realizado o estudo bibliográfico e investigado o estado da arte ao nível do desenvolvimento de jogos, nomeadamente jogos sérios e ao nível da utilização de ambientes virtuais multi-utilizadores (MUVEs). Este estudo permitiu verificar o nível de utilização e a potencialidade que estes temas oferecem na utilização com grupos de trabalho e na promoção da sua dinamização e integração.

Através da utilização de um ambiente virtual multi-utilizador como o Second Life ou o OpenSimulator é possível garantir a construção dos níveis num ambiente virtual 3D, onde cada jogador tem disponível um avatar que controla e com o qual interage com o ambiente, pelo que a decisão de utilizar tais ambientes cumpriu desde logo estes objectivos.

O objectivo de agilizar o processo de criação dos níveis foi atingido, através da modelação expedita dos objectos virtuais. Neste mecanismo desenvolvido a modelação expedita é conseguida através da criação sucessiva dos diferentes elementos virtuais a inserir no cenário, cujas propriedades se encontram previamente definidas. Desta forma é possível criar um nível completo em poucos segundo. Isto permite resolver o problema da utilização de espaço virtual limitado, já que é possível utilizar a mesma área para reconstruir os mais variados níveis de jogo, através da criação e eliminação alternada dos elementos de jogo. Por outro lado permite também a criação de um nível de jogo pronto a ser utilizado em poucos segundos.

Foi também cumprido o objectivo de criação de níveis protótipo, tendo sido reproduzidos os níveis 1 e 2 do projecto FEUP Adventure. Estes níveis criados podem ser utilizados com grupos de jogadores, estando disponíveis para serem jogados assim que são criados. Apresentam uma mecânica simples que requer que os jogadores de cada

equipa se coordenem e interajam de forma a atingir os objectivos e a vencer a equipa adversária.

Com a criação destes níveis protótipo, verificou-se a capacidade da plataforma de modelar níveis simples, que permitem envolver um grupo de utilizadores, no sentido de cooperarem e competirem, no cumprimento dos objectivos de cada nível.

7.2 Trabalho Futuro

Apesar da satisfação dos objectivos propostos, este projecto apresenta vários pontos onde é possível evoluir, seja através da melhoria dos processos utilizados ou através da implementação de novos módulos que permitam novas funcionalidades e características.

Desde logo, um aspecto em que o protótipo desenvolvido apresenta algum inconveniente diz respeito à forma como é especificado o ficheiro de definição do nível - apesar de ser através de uma linguagem simples, o facto de ser necessário escrever a definição manualmente torna a especificação de um nível num processo demorado e susceptível a erros. É necessário conhecer os objecto existentes e as suas propriedades, e a definição da sua posição e orientação no espaço virtual pode exigir algumas sessões de tentativa-erro. Se por um lado a existência de definições pré-criadas permite a utilização e modelação rápida de um nível de jogo, a definição de um novo nível pode não ser um processo simples para um supervisor de uma sessão de utilização deste protótipo. Esta dificuldade pode ser ultrapassada através da criação de uma aplicação que permita gerir os objectos existentes, tomar conhecimento das suas propriedades e indicar a sua posição pretendida no ambiente virtual. Tal aplicação seria externa ao ambiente virtual e funcionaria como uma interface para a especificação dos níveis virtuais, eventualmente através de um pré-visualizador 3D do nível a construir.

Outro aspecto que é possível adicionar a este protótipo diz respeito à criação de mais níveis de jogo, com a criação de mais objectos interactivos e de novos tipos de mecânicas para utilizar com os grupos. Apesar de os níveis criados no protótipo permitirem verificar a capacidade de criar mecânicas de jogo que possam ser utilizadas com grupos de utilizadores, não representam todas as possibilidades que se podem obter nos ambientes virtuais e que a plataforma de modelação expedita pode reproduzir.

Também a implementação de um sistema de gestão das equipas de jogo seria uma mais-valia para este projecto. Seria um sistema que permitisse registar os jogadores presentes, as suas pontuações obtidas anteriormente e permitiria agendar sessões de jogos. Desta forma, a plataforma poderia funcionar tanto a nível interno de uma instituição (escola ou empresa) como a nível global, com todos os utilizadores do mundo virtual.

Em geral, o trabalho desenvolvido nesta dissertação tem bastantes possibilidades de evolução, partindo do protótipo desenvolvido. Além disso, este protótipo constitui um mecanismo funcional de modelação expedita num ambiente virtual, que, saindo do âmbito

geral da dissertação, seria interessante utilizar para outros objectivos que não a criação de jogos.

Referências

[ACT09] Jason Underwood Aline Click e Michael Taylor. Open sim as an alternative to second life, 2009. Disponível em http://www.slideshare.net/Aliandrews/open-sim-as-an-alternative-to-second-life4, acedido pela última vez em Junho de 2011.

[Cat08] CatherineOmega. Lsl wiki: Chat, 2008. Disponível em http://lslwiki.net/lslwiki/wakka.php?wakka=Chat, acedido pela última vez em Maio de 2011.

[Cat10] CatherineOmega. Lsl wiki: Communications, 2010. Disponível em http://lslwiki.net/lslwiki/wakka.php?wakka=communications, acedido pela última vez em Maio de 2011.

[Chr09] ChristopherOmega. Lsl wiki: script, 2009. Disponível em http://lslwiki.net/lslwiki/wakka.php?wakka=script, acedido pela última vez em Maio de 2011.

[CIT] Research Center CITEd. Multi-user virtual environments for education. disponível em http://www.cited.org/index.aspx?page_id=159, acedido pela última vez em Janeiro de 2011.

[CSC10] A. Cruz, A. Sousa e A. Coelho. Using games to promote student integration in universities trough the use of virtual worlds - work in progress. *SLACTIONS2010*, 2010.

[CSC11] A. Cruz, A. Sousa e A. Coelho. Technical analisys and approaches for game development in second life. *In Actas da CISTI'2011 (6ª Conferência Ibérica de Sistemas e Tecnologias de Informação) - First Iberian Workshop on Serious Games and Meaningful Play (SGaMePlay'2011), vol. II, UTAD, Chaves, 2011*, 2011.

[Dan07] Danc. Content is bad, Fevereiro 2007. Lost Garden, disponível em http://www.lostgarden.com/2007/02/content-is-bad.html, acedido pela última vez em Abril de 2011.

[DC07] Edward Dieterle e Jody Clarke. *Multi-User Virtual Environments for Teaching and Learning*, pages 1–11. Idea Group, Inc., Hershey, 2007.

[Der07] A Derryberry. Serious games : online games for learning. 2009(June 17), 2007.

[Des11] MSU Serious Game Design. Serious games companies, 2011. Disponível em http://seriousgames.msu.edu/companies.php, acedido pela última vez em Janeiro de 2011.

[Dev04] DevMaster. Cryengine, 2004. DevMaster.net, 3D engines database, disponível em http://www.devmaster.net/engines/engine_details.php?id=57, acedido pela última vez em Fevereiro de 2011.

[Dev07] DevMaster. Torque game engine, 2007. DevMaster.net, 3D engines database, disponível em http://www.devmaster.net/engines/engine_details.php?id=3, acedido pela última vez em Fevereiro de 2011.

[Dev08] DevMaster. Second life, 2008. DevMaster.net, 3D engines database, disponível em http://www.devmaster.net/engines/engine_details.php?id=306, acedido pela última vez em Maio de 2011.

[Dev10a] DevMaster. Unity, 2010. DevMaster.net, 3D engines database, disponível em http://www.devmaster.net/engines/engine_details.php?id=256, acedido pela última vez em Fevereiro de 2011.

[Dev10b] DevMaster. Unreal development kit (udk), 2010. DevMaster.net, 3D engines database, disponível em http://www.devmaster.net/engines/engine_details.php?id=635, acedido pela última vez em Fevereiro de 2011.

[DF06] S De Freitas. Learning in immersive worlds: A review of game-based learning. *JISC eLearning Innovation*, 3.3:73, 2006.

[Ezh09] EzharFairlight. Lsl wiki: Simulator, 2009. Disponível em http://lslwiki.net/lslwiki/wakka.php?wakka=simulator, acedido pela última vez em Maio de 2011.

[FEU10] FEUP. Projecto feup 2010, 2010. Disponível em http://www.fe.up.pt/si/noticias_geral.ver_noticia?p_nr=11008, acedido pela última vez em Dezembro de 2010.

[Fre09] W. Freeman. The top 10 game engines revealed, 2009. develop, disponível em http://www.develop-online.net/news/32250/The-top-10-game-engines-revealed, acedido pela última vez em Fevereiro de 2011.

[Gmb11] Crytek GmbH. Educational development license, 2011. CryEngine, Lincensing - Educational SDK, disponível em http://mycryengine.com/?conid=42, acedido pela última vez em Janeiro de 2011.

[HC09] M.L.L. Honey e K. Connor. Transforming Learning: Using a Multi-User Virtual Environment for simulation. 2009.

[Hob06] Neville Hobson. Get a second life now, 2006. Disponível em http://www.nevillehobson.com/2006/06/24/get-a-second-life-now/, acedido pela última vez em Julho de 2011.

[KDCN06] D J Ketelhut, C Dede, J Clarke e B Nelson. A multi-user virtual environment for building higher order inquiry skills in science. 2006.

[Kid10] Software Kids. Time engineers, 2010. Disponível em http://www.software-kids.com/html/time_engineers.html, acedido pela última vez em Fevereiro de 2011.

[KK05] Linda Kao e Yasmin Kafai. A totally different world: Playing and learning in multi-user virtual environments. *North*, 2005.

[KOS09] Eric Klopfer, Scot Osterweil e Katie Salen. moving learning games forward. *Flora*, page 59, 2009.

[McK10] Mike McKay. Sloodle - tutorials, 2010. Disponível em http://www.sloodle.org/blog/?page_id=33, acedido pela última vez em Julho de 2011.

[MSS04] Alice Mitchell e Carol Savill-Smith. The use of computer and video games for learning - a review of the literature. page 0, 2004.

[NAH+06] Michael Nitsche, C Ashmore, W Hankinson, R Fitzpatrick, J Kelly e K Margenau. *Designing procedural game spaces: A case study*, pages 10–12. Citeseer, 2006.

[NMC06] NMC. Nmc sparking innovation, learning & creativity, 2006. Disponível em http://www.nmc.org/, acedido pela última vez em Fevereiro de 2011.

[Ope11] OpenSimulator. Configuration, 2011. Disponível em http://opensimulator.org/wiki/Configuration, acedido pela última vez em Maio de 2011.

[PDDFP10] Panagiotis Petridis, Ian Dunwell, Sara De Freitas e David Panzoli. An engine selection methodology for high fidelity serious games. *2010 Second International Conference on Games and Virtual Worlds for Serious Applications*, pages 27–34, 2010.

[PGF08] A. Peachey, J. Gillen e R. Ferguson. Fluid leadership in a multi-user virtual environment educational project with teenagers: Schome Park. In *Ecologies of Diversities: The Developmental and Historical Interarticulation of Human Mediational Forms: Meeting of the International Society for Cultural and Activity Research, San Diego, USA, September*, pages 8–13. Citeseer, 2008.

[Raf08] Igor Rafailov. Games: Uma nova forma de aprender, treinar e recrutar, 2008. RH, disponível em http://www.rh.com.br/Portal/Geral/Blog_Igor_Rafailov/5593/games-uma-nova-forma-de-aprender-treinar-e-recrutar.html, acedido pela última vez em Junho de 2011.

[Rud09] Nicholas Eugene Rudzicz. Arda: a framework for procedural video game content generation. *Science*, (Fevereiro), 2009.

[SBGH04] Kurt Squire, Mike Barnett, Jamillah M Grant e Thomas Higginbotham. Electromagnetism supercharged!: learning physics with digital simulation games. *International Conference on Learning Sciences*, page 513, 2004.

[SLO11] SLOODLE. Sloodle - slis second life wiki, 2011. Disponível em http://slisweb.sjsu.edu/sl/index.php/Sloodle, acedido pela última vez em Fevereiro de 2011.

[Tee08] R. Teed. What is game-based learning?, Maio 2008. SERC, Carleton College Starting Point, disponível em http://serc.carleton.edu/introgeo/games/whatis.html, acedido pela última vez em Janeiro de 2011.

[Tee10] R. Teed. What makes a good game?, Maio 2010. SERC, Carleton College Starting Point, disponível em http://serc.carleton.edu/introgeo/games/goodgame.html, acedido pela última vez em Janeiro de 2011.

[Uni09a] Harvard University. Introducing river city, 2009. The River City Project. Disponível em http://muve.gse.harvard.edu/rivercityproject, acedido pela última vez em Julho de 2011.

[Uni09b] Harvard University. River city interface, 2009. The River City Project. Disponível em http://rivercity.activeworlds.com/rivercityproject/view/rc_views_interface.htm, acedido pela última vez em Julho de 2011.

[Val08] Valerie. Second life in higher education: Surveying pros and cons, 2008. Educational Development Centre. Disponível em http://edc.carleton.ca/blog/index.php/2008/02/05/second-life-in-higher-education-surveying-pros-and-cons/, acedido pela última vez em Julho de 2011.

[VCM+10] Andreas Vilela, Márcio Cardoso, D Martins, Arnaldo Santos, Lúcia Moreira, Hugo Paredes, Paulo Martins e Leonel Morgado. *Privacy challenges and methods for virtual classrooms in Second Life Grid and OpenSimulator*, pages 167–174. IEEE, 2010.

[Wik11] Wikipedia. Massively multiplayer online game, 2011. Wikipedia, disponível em http://en.wikipedia.org/wiki/Massively_multiplayer_online_game, acedido pela última vez em Janeiro de 2011.

[Wis10] WisdomTools. Astroengineer: Moon rover, 2010. Disponível em http://wisdomtools.com/projects/edu_astro.html, acedido pela última vez em Fevereiro de 2011.

[WM07] J. Wollongong e S. McDunnough. An introduction to second life and its educational possibilities, 2007.

[ZKD08] Kaiwen Zhang, Bettina Kemme e Alexandre Denault. *Persistence in massively multiplayer online games*, pages 53–58. ACM Press, 2008.

Anexo A

Definição dos Níveis

Nesta secção são apresentadas as definições necessárias para a reprodução dos níveis 1 e 2, com todos os seus elementos, tal como foram utilizadas no ficheiro externo de definição do nível de jogo.

A.0.1 Nível 1

```
//<object_name> <pos_x> <pos_y> <pos_z> <rot_x> <rot_y> <rot_z> (<extra_param>)
//red team zone
redTeamRamp 17.5 0.0 1.5 0.0 0.0 180.0
redTeamPlatform 23.5 0.0 2.8

//blue team zone
blueTeamRamp -17.5 0.0 1.5
blueTeamPlatform -23.5 0.0 2.8

platform 10.0 -11.0 4.0
platform 10.0 -9.0 4.0
platform 0.0 -11.0 5.0
platform 0.0 -9.0 5.0
platform -10.0 -11.0 6.0
platform -10.0 -9.0 6.0

platform 10.0 11.0 6.0
platform 10.0 9.0 6.0
platform 0.0 11.0 5.0
platform 0.0 9.0 5.0
platform -10.0 11.0 4.0
platform -10.0 9.0 4.0

blueTeamRing -1.0 -5.0 1.0
blueTeamRing -1.0 -2.5 1.0
blueTeamRing -1.0 0.0 1.0
blueTeamRing -1.0 2.5 1.0

redTeamRing 1.0 -5.0 1.0
redTeamRing 1.0 -2.5 1.0
redTeamRing 1.0 0.0 1.0
redTeamRing 1.0 2.5 1.0
```

```
createRoundCubes 0.0 0.0 7.0
startGame 0.0 0.0 5.0 180.0 0.0 0.0 60.0
```

A.0.2 Nível 2

```
//<object_name> <pos_x> <pos_y> <pos_z> <rot_x> <rot_y> <rot_z> (<extra_param>)
//1st ramp zone
gameWall 5.0 5.0 7.5 0.0 90.0 0.0
gameWall 5.0 0.0 7.5 0.0 90.0 0.0
bigRamp 10.0 7.5 1.5 0.0 0.0 270.0
bigRamp 20.0 7.5 1.5 0.0 0.0 270.0
bigRamp 10.0 7.5 1.5 180.0 0.0 270.0
bigRamp 20.0 7.5 1.5 180.0 0.0 270.0
bigRamp 10.0 7.5 4.5 0.0 0.0 270.0
bigRamp 20.0 7.5 4.5 0.0 0.0 270.0
bigRamp 10.0 -2.5 1.5 0.0 0.0 270.0
bigRamp 20.0 -2.5 1.5 0.0 0.0 270.0
gameWall 25.0 5.0 7.5 0.0 90.0 0.0
gameWall 25.0 0.0 7.5 0.0 90.0 0.0

eater 15.0 -10.0 1.0
redTeamWall 15.0 -15 5.0 0.0 0.0 90.0
createSphere 15.0 7.5 12.0 0.0 0.0 0.0 5.0

//2st ramp zone
gameWall -5.0 5.0 7.5 0.0 90.0 0.0
gameWall -5.0 0.0 7.5 0.0 90.0 0.0
bigRamp -10.0 7.5 1.5 0.0 0.0 270.0
bigRamp -20.0 7.5 1.5 0.0 0.0 270.0
bigRamp -10.0 7.5 1.5 180.0 0.0 270.0
bigRamp -20.0 7.5 1.5 180.0 0.0 270.0
bigRamp -10.0 7.5 4.5 0.0 0.0 270.0
bigRamp -20.0 7.5 4.5 0.0 0.0 270.0
bigRamp -10.0 -2.5 1.5 0.0 0.0 270.0
bigRamp -20.0 -2.5 1.5 0.0 0.0 270.0
gameWall -25.0 5.0 7.5 0.0 90.0 0.0
gameWall -25.0 0.0 7.5 0.0 90.0 0.0

eater -15.0 -10.0 1.0
blueTeamWall -15.0 -15 5.0 0.0 0.0 90.0
createSphere -15.0 7.5 12.0 0.0 0.0 0.0 5.0

blueTeamRing -1.0 -5.0 1.0
blueTeamRing -1.0 -2.5 1.0
blueTeamRing -1.0 0.0 1.0
blueTeamRing -1.0 2.5 1.0

redTeamRing 1.0 -5.0 1.0
redTeamRing 1.0 -2.5 1.0
redTeamRing 1.0 0.0 1.0
redTeamRing 1.0 2.5 1.0

startGame 0.0 0.0 5.0 180.0 0.0 0.0 60.0
```