



Universidade de Évora - Escola de Ciências e Tecnologia

Mestrado em Engenharia Informática

Dissertação

**E-learning Imersivo - Atividades na perspetiva da primeira
pessoa**

Carlos Manuel da Silva Figueiredo

Orientador(es) | Francisco Manuel Coelho

Évora 2022





Universidade de Évora - Escola de Ciências e Tecnologia

Mestrado em Engenharia Informática

Dissertação

**E-learning Imersivo - Atividades na perspetiva da primeira
pessoa**

Carlos Manuel da Silva Figueiredo

Orientador(es) | Francisco Manuel Coelho

Évora 2022



A dissertação foi objeto de apreciação e discussão pública pelo seguinte júri nomeado pelo Diretor da Escola de Ciências e Tecnologia:

Presidente | Lígia Maria Ferreira (Universidade de Évora)

Vogais | António Anjos (Universidade de Évora) (Arguente)
Francisco Manuel Coelho (Universidade de Évora) (Orientador)

*Dedico esta dissertação ao meu primo Gonçalo, que me apoiou no início da minha jornada na faculdade
mas que infelizmente nos deixou há alguns anos*

Prefácio

Em termos académicos, existem disciplinas com as quais os alunos tendem a ter dificuldades, tal como por exemplo a matemática ou físico-química. Alguma dessa dificuldade é inerente à natureza abstrata de alguns temas, como a álgebra ou reações entre átomos, mas outros temas, como a trigonometria ou as forças que podem ser aplicadas sobre objetos, são intrinsecamente físicos. Nestes casos a aprendizagem pode beneficiar da interação direta com objetos ou cenários adequados.

Por um lado os desenvolvimentos recentes na realidade virtual proporcionam um acesso generalizado às bases técnicas necessários para o consumo e construção de conteúdos pedagógicos vocacionados para estas disciplinas da matemática. Por outro lado o ensino à distância em plataformas digitais tornou-se uma necessidade, não só pelos sucessivos confinamentos da pandemia COVID19, mas por razões económicas, geográficas e sociais.

Este trabalho começa a exploração da aplicação de RV (*Realidade Virtual*) em *E-learning*. Mais concretamente, é descrito um sistema, ainda protótipo, que consiste num ambiente virtual com várias atividades e com percursos que encadeiam estas em função das respostas que o aluno escolhe. Ficaram por concretizar inúmeras opções e desenvolvimentos mas chegou-se a uma versão que concretiza o essencial.

Agradecimentos

Agradeço aos meus pais por me terem suportado ao longo desta jornada.

Agradeço ao meu orientador por ter sido imensamente paciente comigo e por ter aceite responder a todas as minhas questões a qualquer altura.

Agradeço aos meus amigos por me terem ajudado a testar o sistema num *desktop* diferente do que eu uso, permitindo que fizesse ajustes para o mesmo.

Conteúdo

Conteúdo	xi
Lista de Figuras	xiii
Lista de Acrónimos	xv
Sumário	xvii
Abstract	xix
1 Introdução	1
1.1 Definição de E-learning	1
1.2 Definição de Realidade Virtual	3
1.3 Motivação	4
2 Estado da Arte	7
3 Descrição do projeto	13
4 Resolução	15
4.1 H5P	15
4.1.1 Detalhes H5P	16
4.2 Projeto em <i>Javascript</i>	16
4.2.1 Superclasses	17
4.2.2 Funcionamento do projeto	17
4.2.3 Como é realizado o movimento da câmara?	18
4.2.4 Uso de Backend	19

4.3	Implementação do projeto como uma atividade no <i>Moodle</i>	20
4.3.1	Passo 1: criação de uma atividade a partir de ficheiros	20
4.3.2	Passo 2: criação da atividade <i>iframe</i> do H5P	20
4.4	Como correr o projeto	21
4.4.1	Abrir o projeto sem RV	21
4.4.2	Abrir o projeto com RV	21
4.4.3	Possíveis problemas com a observação do projeto	21
4.4.4	Estado intermédio do projeto	21
4.4.5	Estado final do projeto	22
4.4.6	Explicação das funções no código	24
5	Resultados Empíricos	29
6	Conclusão e Trabalho Futuro	31
	Bibliografia	33

Lista de Figuras

1.1	Jornal Boston Gazette	2
1.2	Equipamento de RV	3
1.3	Uma MUD	4
2.1	Comparação da área real e da simulação no Kokoda VR	9
2.2	Ambiente RV de CicerOn	9
2.3	Testadores de CicerOn	10
2.4	Equipamento Google Cardboard	10
4.1	Um dos primeiros ambientes criados. Nota-se a existência de um chão, uma imagem de fundo e um objeto no canto inferior direito que seria observado pela câmara	16
4.2	Estado intermédio do projeto	22
4.3	Estado final do projeto	22
5.1	Vista de cima de todo o ambiente.	30
5.2	Exercício na página do Moodle.	30

Lista de Acrónimos

CSS	<i>Cascading Style Sheets</i>
H5P	<i>HTML5 Package</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hyper Text Transfer Protocol</i>
HTTPS	<i>Hyper Text Transfer Protocol Secure</i>
MOO	<i>MUD Object Oriented</i>
MOOC	<i>Massive Open Online Course</i>
RV	<i>Realidade Virtual</i>
SPOC	<i>Small Private Online Course</i>
SSL	<i>Secure Sockets Layer</i>
UI	<i>User Interface</i>

Sumário

Este trabalho explora a possibilidade técnica de definir certos conteúdos pedagógicos em ambientes de realidade virtual.

O ensino à distância baseado em canais digitais (*E-learning*) é uma necessidade crescente, dado que contribui para a redução de barreiras económicas, geográficas ou sociais. Por outro lado, a manipulação direta de objetos ou a interação com cenários é uma importante forma de aprendizagem, por exemplo, de conceitos geométricos ou espaciais. Esta tese existe na intersecção entre os temas de RV (*Realidade Virtual*) e de *E-learning*.

O desenvolvimento principal deste texto consiste na descrição dos aspectos técnicos de um sistema informático para a construção de um ambiente RV interativo. Este ambiente pode ser visitado através de um computador ou com óculos de RV simples, como o *Google Cardboard*. Neste, o utilizador pode caminhar num percurso entre vários exercícios de escolha múltipla, dependendo da resposta a cada pergunta.

Palavras chave: Realidade Virtual, e-Learning, Ensino, Aprendizagem, Matemática, Geometria

Abstract

Immersive E-learning

Activities in a first-person perspective

This work explores the possibility of defining certain pedagogic contents in a virtual reality environment. Distance learning based on digital channels (E-Learning) is a rising necessity, given that it contributes to the reduction of economic, geographical or social barriers. On the other hand the direct manipulation of objects or the direct interaction with certain scenarios is an important way of learning, for example, spacial or geometrical concepts. This thesis exists in the intersection between the themes of VR and E-Learning. The main development of this text consists in the description of several technical aspects of a digital system for the construction of an interactive VR environment. This environment can be visited through a computer or with the use of simple VR glasses, such as Google Cardboard. In this, the user can travel a set path between several multiple choice exercises, this path being defined by the answer to each question.

Keywords: Virtual Reality, e-Learning, Teaching, Learning, Math, Geometry

1

Introdução

Desde a invenção do primeiro computador, a tecnologia tem evoluído a um ritmo impressionante. Hoje em dia, as tarefas que podem ser executadas através de um computador são praticamente infinitas, desde contactar pessoas a partilhar informação. Neste trabalho iremos falar de duas áreas da informática que estão ambas em rápido crescimento: E-learning e RV (*Realidade Virtual*). Começemos pela definição de ambas:

1.1 Definição de E-learning

Segundo a Universidade de Cambridge, a definição de E-learning é “*the process of providing courses on the internet or an intranet*”. Basicamente, E-learning é qualquer sistema de aprendizagem que possa ser acedido através da Internet. Esta definição é a mais abrangente, no entanto parece haver várias interpretações sobre este tema.

A aprendizagem à distância nem sempre foi por meio de máquinas, sendo que o primeiro caso recordado desta foi em 1728, com um anúncio no jornal Inglês *Boston Gazette* (figura 1.1) que vendia um curso com aulas semanais enviadas pelo correio. Com a evolução da tecnologia, também mudou a maneira de como a

aprendizagem à distância era recebida, como foi demonstrado em 1969 quando a *Open University*, fundada no Reino Unido, providenciou cursos por correio e televisão, tal como cursos em pessoa numa seleção de locais físicos.

De igual forma, com o aparecimento da *Internet* este tipo de ensino também começou a ser transmitido por esses meios, sendo que em 1989 a *University of Phoenix* começou a fornecer vários cursos *online*, incluindo licenciaturas e mestrados [KH16].



Figura 1.1: Jornal Boston Gazette

Este tipo de educação tem o seu conjunto de benefícios e desvantagens.

Um dos benefícios de E-learning é a maneira como permite a todos a aprender ao seu próprio ritmo, podendo impor os horários que mais lhes convêm [AA14]. Adicionam ainda que isto é ainda mais benéfico para pessoas já com carreiras, visto que isto lhes permite focar em ambos estes objetivos.

Este tipo de E-learning (que este pode ser interpretado como um MOOC (*Massive Open Online Course*) não é o único, havendo outro tipo denominado por SPOC (*Small Private Online Course*), este último seguindo os horários e regulações das aulas presenciais, mas podendo apenas ser realizado à distância [KH16]. Ao contrário de SPOCs, MOOCs não removem apenas o obstáculo do espaço entre o estudante e o espaço de ensino, como também têm a capacidade de remover o obstáculo de tempo, sendo que estes permitem que os alunos aprendam ao seu próprio ritmo. É de realçar que MOOCs não têm apenas vantagens, visto que neste caso os alunos são inteiramente responsáveis pela sua aprendizagem, tendo que ter auto-disciplina suficiente para seguir a disciplina sem distrações.

Para estes tipos de educação à distância assíncronos, ou seja em ritmos diferentes que todos os outros estudantes a tirarem o mesmo curso, Kaplan e Haenlein sugerem algumas adaptações pelas instituições de ensino ao ambiente de E-learning para que os estudantes se sintam mais envolvidos e para que a sua motivação seja melhorada, tal como:

Criar um modo de contacto entre estudantes, dando assim um ambiente em que possam interagir uns com os outros, permitindo uma melhor colaboração em certos trabalhos.

Dar a oportunidade de contactar o professor, para que os estudantes possam tirar dúvidas, sendo que estas aulas não envolvem muito contacto e logo a informação pode não ser bem compreendida.

É de salientar também que, embora E-learning tenha começado como uma maneira de transmitir conhecimento académico para estudantes, este também tem vindo a ser usado pelos seus empregadores para ensinar trabalhadores em várias áreas sobre como realizar determinadas tarefas.

1.2 Definição de Realidade Virtual

Realidade Virtual (conhecida em inglês como VR) é definida como sendo os vários casos em que um computador consegue replicar experiências (visuais, auditivas, etc...) para um utilizador de maneira a o fazer sentir que essas são reais (por exemplo, ao ouvirmos sons gravados de chuva a cair conseguimos imaginar isso a acontecer na vida real) [Yoh01].

No entanto, esta não é a definição que é mais reconhecida, sendo essa "*the use of computer generated virtual environments and the associated hardware to provide the user with the illusion of physical presence within that environment*" [JCL97], correspondendo esta ao uso dos equipamentos que usamos agora (ver figura 1.2).



Figura 1.2: Equipamento de RV

Com a possibilidade de manusear objetos num mundo virtual, esta tecnologia tem o potencial de ser útil em várias áreas, devido ao facto de haver a possibilidade de conseguir substituir os custos de materiais voláteis ou caros ou até de ajudar a perceber alguma informação através da interação com o tema em questão. Por esta razão e outras, têm havido vários projetos envolvendo esta tecnologia, alguns dos quais mostraremos aqui.

Esta é a interpretação de RV que temos hoje em dia, sendo que como Schwienhorst aponta, existem pelo menos dois tipos de RV: *high-end*, sendo este o tipo que mais conhecemos hoje em dia em que manuseamos objetos num ambiente muito parecido com o nosso, normalmente com a ajuda de equipamentos sofisticados e *low-end*, sendo este mais simples, usando maioritariamente texto para definir o mundo existente.

Vamos olhar um pouco com o que se pode ser realizado com RV.

Usos de realidade virtual Foi proposto que ferramentas de realidade virtual fossem usadas para o design de produtos no início do seu desenvolvimento. Isto faria com que os designers conseguissem "moldar" o produto que têm em mente com mais facilidade, tal como permitir a todos os outros perceber melhor como este produto iria parecer e funcionar [JCL97]. Realçamos no entanto que esta análise foi feita em 1997, sendo que a tecnologia ainda não era tão avançada como a que temos hoje.

Foi realizada então uma pesquisa sobre esta tecnologia numa altura em que já existiam exemplos desta implementação [SVO11]. No artigo, explicam que existem várias complicações em relação a esta aplicação, tal como a imposição de colisão entre todos os objetos. Para isto, existem dois sistemas de restrições diferentes:

Deteção de *constraints* inter-partes que define restrições que devem ser aplicadas sobre a interação entre vários objetos, usando um "*solver*" de *constraints*. Devido ao grande número de cálculos que têm que ser feitos neste método, esta técnica é imprópria para uso no mundo real.

Essa interação pode ser feita de várias maneiras, sendo que a mais comum é a realização de exercícios, se bem que esta serve mais para testar a compreensão, do que para explicar a matéria em si.

Embora a interação física com algo seja mais benéfica para uma melhor aprendizagem, esta pode não ser muito viável, devido ao facto de materiais para esta poderem ser difíceis de encontrar, serem caros, ou um ponto que é cada vez mais relevante: a inabilidade dos alunos estarem presentes na mesma sala que os seus professores, por uma razão ou outra.

E se houvesse algo que os professores e alunos pudessem utilizar para melhorar a sua aprendizagem/ensino? Algo que permitisse que os alunos interagissem diretamente com a matéria em questão sem ser necessário estarem na mesma sala que todos os outros? Algo que pudesse ser visualizado a partir do seu telemóvel?

Para resolver estes problemas apresentamos o nosso projeto, que vem tentar criar um sistema que visa a facilitar a aprendizagem de conhecimentos usando RV.

2

Estado da Arte

A pesquisa por RV (*Realidade Virtual*) teve um aumento entre os anos de 2016 e 2018. Isto parece ser atribuído ao desenvolvimento do *headset HTC Vive*, sendo este um dos primeiros *headsets* RV comerciais [RMFW20] e [CA20].

Temos vindo a falar do uso de RV para a educação, no entanto será que isso tem algum impacto na aprendizagem? Foi feita uma meta-análise de vários artigos onde foram feitos estudos sobre o uso de RV no ensino superior, analisando tanto as técnicas para a construção dos ambientes em si, como as técnicas para a estruturação dos exercícios de um ponto educacional [RMFW20].

Nesta análise, alguns artigos foram explorados mais que outros, todos indicando usos diferentes de RV. Estes são alguns dos mencionados:

É mencionado o uso de RV para a sensibilização de como evacuar um dado local quando acontece alguma emergência. Os participantes foram testados em várias áreas, como as ações a tomar em caso de emergência e o seu conhecimento espacial, através de questionários e do resultado dos *logs* do sistema depois destes o terem usado, registando o dano tomado e o tempo de saída.

Foram usados vários ambientes de Realidade Virtual para treinar várias pessoas em conhecimentos de engenharia de construção, tal como RV imersivo (o uso de óculos RV e luvas com sensores), um sistema *CAVE* (colocar o utilizador numa sala em que é projetado nas paredes em volta dele o ambiente virtual, sendo este adaptado conforme os movimentos do utilizador) e RV de *desktop* (um ambiente 3D interativo, mas apenas visualizado através do ecrã de um computador). Mais especificamente, treinaram as habilidades de visualização arquitetural, design e treino de segurança de construção. Este sistema consistia num ambiente de modelação 3D, o qual os estudantes poderiam interagir, e um sistema de perguntas e respostas. O projeto concluiu que o uso de RV deu aos participantes um bom controlo sobre o ambiente tal como aumentou a sua concentração nos exercícios.

Foi feito um estudo sobre a utilização de RV na educação em geral, avaliando os benefícios, caso houvessem. O estudo foi feito sobre vários exercícios em RV realizados sobre várias disciplinas, estes exercícios variando na sua estrutura, sendo que alguns eram simples perguntas enquanto que outros simulavam a realização de algumas tarefas como se fossem feitas no mundo real. Os resultados do estudo revelaram que a maioria das áreas treinadas (medicina, engenharia, línguas e psicologia) tiveram benefícios da inclusão de RV no currículo, à excepção de psicologia, esta não tendo nenhuma mudança notável. Foi notado que o uso de RV ajuda a viver experiências perto da realidade, aumenta a motivação e melhora as habilidades treinadas no mesmo.

Segundo uma pesquisa sobre um total de 38 artigos envolvendo o uso de realidade virtual num ambiente de ensino superior, o desempenho dos estudantes pareceu aumentar sempre que estes eram envolvidos num ambiente imersivo, independente de se o tipo de conhecimento aprendido era declarativo (factos, como "2+2=4") ou prático (como realizar determinadas tarefas) [RMFW20].

Falando de educação escolar, foi realizada uma experiência com alunos do ciclo na Austrália e alunos de uma universidade na Índia para testar também se a imersão poderia ter algum impacto benéfico na educação [CA20]. Para isto criaram duas narrativas de 40 minutos, sendo que uma destas era um vídeo 360° e outra era uma experiência RV. Estas foram feitas sobre uma campanha militar que ocorreu na zona de *Kokoda* na Austrália na Segunda Guerra Mundial (replicada na figura 2.1). Os alunos australianos tinham pouco conhecimento sobre esta (sendo que muitos tinham alguns factos errados, por exemplo pensavam que tinha sido na Primeira Guerra Mundial) e os alunos indianos não tinham qualquer conhecimento sobre esta.

Para que a experiência não fosse drasticamente diferente entre os grupos que usaram o *Kokoda VR* e o *Kokoda 360°*, as mesmas ferramentas que foram usadas para criar os ambientes 3D em RV foram as mesmas usadas para criar os ambientes nos vídeos 360°. A única diferença entre a visualização de *Kokoda VR* para *Kokoda 360°* é que na experiência RV os alunos poderiam mover-se e interagir um pouco com o ambiente.

Os conhecimentos destes foram testados logo antes e 2 semanas depois da experiência para melhor avaliar a evolução dos seus conhecimentos, sendo que todos os alunos apresentaram um resultado positivo para isto, os alunos com melhores resultados sendo os que testaram a experiência RV. Quando questionados sobre a qualidade da experiência, os alunos do *Kokoda VR* exprimiram uma maior imersão, empatia e sentido de presença que os alunos do *Kokoda 360°*.

Ao longo desta pesquisa, ficou claro que o uso de RV pareceu melhorar várias áreas relevantes na aprendizagem, tal como a motivação, o interesse e o melhoramento de habilidades treinadas no ambiente, embora que ainda existam certos casos (como vimos em relação à psicologia) em que este estilo de aprendizagem não aparenta ter grande impacto. Este facto é reforçado por Calvert J. et al., referindo este que quando os conhecimentos dos estudantes foram testados duas semanas depois da experiência, aqueles que tomaram conhecimentos através dos *headsets* RV em vez dos vídeos de 360° tiveram melhores resultados. Quando questionados sobre a sua opinião sobre o ambiente, a maioria relacionou o realismo e a liberdade



Figura 2.1: Comparação da área real e da simulação no Kokoda VR

de interatividade com a qualidade deste.

Os conhecimentos falados até agora podem todos, até um certo ponto, ser transmitidos de uma maneira ou de outra de forma declarativa (factos podem ser simplesmente transmitidos de forma verbal, tarefas podem ser divididas numa lista de pequenas ações, etc...). Existem no entanto conhecimentos que apenas podem ser obtidos através de experiência, tal como a confiança para falar em público. Este foi um dos objetivos para a criação de *CicerOn*, um jogo RV que visa a dar a confiança necessária para falar em público a um certo grupo de pessoas, mais especificamente pessoas com síndrome de *Asperger* [RMR⁺19].

Esta experiência consistiu na criação de um jogo imersivo com vários níveis (figura 2.2), sendo que cada nível não só ficaria mais difícil, como o nível de stress induzido pela comunicação verbal também aumentaria. De maneira a não sobre-carregar os utilizadores, foi criada uma área que serve de base e de selecionador de níveis. Esta foi criada para ser um espaço de relaxamento para o utilizador, que pode ser visitado sempre que este se estiver a sentir demasiado ansioso.

O *loop* de jogabilidade é o seguinte: cada nível começa com o jogador tendo um pedaço de um documento, sendo que cada pedaço encontrado terá uma pista para o próximo. Quando o jogador tiver todas as peças do documento, este revelará alguns parágrafos no seu verso, que o utilizador terá que ler em voz alta na última parte do respectivo nível.

Conforme o utilizador avança nos níveis, as situações em que este tem que ler o documento vão ficando cada vez mais indutoras de stress (começam por ter que ler para um auditório vazio, para 3 pessoas que não prestam atenção, para um grupo de 10 pessoas que por vezes prestam atenção, etc...).



Figura 2.2: Ambiente RV de CicerOn

O projeto foi criado diretamente para adultos, visto que a maioria dos tratamentos para esta desordem são

mais direcionados para crianças e adolescentes devido ao tratamento entre os 0 e 6 anos poder mitigar mais os efeitos de *Asperger*.

Devido à importância de criar um ambiente que fosse acolhedor e não hostil para o público alvo, o desenvolvimento do projeto foi acompanhado de perto por vários especialistas médicos peritos na área do espectro autista, tal como por várias pessoas que têm este síndrome, algumas destas até fazendo parte da equipa de desenvolvimento do projeto (figura 2.3).



Figura 2.3: Testadores de CicerOn

Este jogo foi criado para não necessitar que o utilizador faça pausas (um facto que é mostrado na criação de uma área de relaxamento, em vez de contar com o utilizador retirar os óculos quando se sente sobrecarregado). Isto é algo que pode ser um problema com o uso de óculos RV, uma tecnologia que é famosa por causar enjoo de movimento. Isto implicou que, de maneira a que o sistema fosse o mais realista possível (de maneira a simular com fidelidade as experiências em que se queria que os utilizadores fossem treinados), mantendo a suavidade do movimento da câmara, foi limitado quantos objetos estavam presentes à volta do utilizador a cada altura.

Os trabalhos que temos andado a falar todos usam equipamentos de RV algo avançados que requerem o uso de controladores além de óculos RV, como o *Oculus Rift* por exemplo, no entanto já existiram aplicações de RV no ensino que o fizeram usando a tecnologia que iremos usar neste projeto, ou seja, com algo pelo menos parecido com o *Google Cardboard* (figura 2.4).



Figura 2.4: Equipamento Google Cardboard

Já existem várias salas de aulas a utilizarem RV como um suplemento educativo, em alguns casos utilizando-a para enriquecer os conhecimentos obtidos nas aulas com visitas guiadas virtuais a locais que muitos alunos não poderiam ir, quer por falta de fundos, quer por estes locais já não existirem [Tho18].

Uma dessas salas de aulas é a de *Melinda Lohan* que ensina alunos do liceu em Massachusetts, Estados Unidos, sobre acontecimentos históricos tais como a Segunda Guerra Mundial. É pedido que os alunos usem os seus próprios telemóveis, o equipamento de RV sendo fornecido pela escola. Os alunos são pedidos para verem lições em casa (gravadas pela professora) para depois "visitarem" os locais através de RV na sala de aula.

Como mencionado, estas visitas guiadas são acedidas em dispositivos móveis, no entanto estas não oferecem interatividade, os alunos sendo apenas guiados por um caminho pré-definido. Isto é algo que queremos diferenciar no nosso projeto, criando um ambiente RV em dispositivos móveis que oferece alguma interatividade ao utilizador sem o uso de controladores.

3

Descrição do projeto

Iremos começar por construir um ambiente *3D* que permita que o utilizador interaja com objetos *3D* neste. De seguida iremos incluir uma maneira de mover a câmara para que o utilizador consiga mover-se entre os exercícios. Finalmente iremos proceder à criação de exercícios, estes consistindo de alguns objetos *3D* interactivos e de possivelmente uma introdução em texto ou em vídeo antes de mover o utilizador para o exercício em si.

O ambiente *3D* será inicialmente criado usando a biblioteca gráfica *Three.js*. Esta biblioteca será usada devido ao facto de permitir implementar funcionalidades RV (*Realidade Virtual*) e da documentação ser extensiva e de fácil acesso. Embora esta seja a biblioteca gráfica com que iremos começar a desenvolver o nosso projeto, o nosso objectivo é de preparar este de modo a incluir várias bibliotecas gráficas, algumas que temos em mente sendo o *A-Frame* e o *X3D*.

Para testarmos a viabilidade do nosso projeto em relação ao uso deste em ambientes de ensino como o *Moodle*, iremos usar uma aplicação denominada *Bitnami*. *Bitnami* é uma biblioteca de instaladores de aplicações *web*. Alguns exemplos de aplicações disponíveis nesta biblioteca são o *Moodle*, o *Gitlab* e o *Wordpress*.

Como iremos criar exercícios de maneira a serem implementados na plataforma *Moodle*, iremos também incluir alguma implementação de um tipo de conteúdo H5P (*HTML5 Package*). H5P é um plugin compatível com várias plataformas (*Wordpress, Drupal, Moodle, etc...*), trazendo várias “templates” de conteúdos com que é possível criar vários exercícios (alguns exemplos destes são “Palavras Cruzadas”, “Perguntas de Escolha Múltipla”, “Questionários”, etc...). Depois de criado um exercício baseado numa destas “templates”, este pode ser “empacotado” num ficheiro do tipo “.h5p” para que possa então ser partilhado por entre as plataformas compatíveis com o H5P.

O tipo de conteúdo que iremos tentar implementar inicialmente é o “Virtual Tour 360º”, que nos permite ver uma imagem como se esta estivesse envolvida à nossa volta. Este tipo de conteúdo também inclui uma maneira de colocar pontos ao longo da imagem que nos permitem transitar para outro lugar, esta sendo a maneira que estamos a pensar em implementar interação do utilizador.

4

Resolução

Antes da realização do projeto, devido ao número de partes do trabalho que teriam que trabalhar em conjunto (como o *Moodle*, a utilização do H5P (*HTML5 Package*), o uso de *Javascript* para a criação das ferramentas para a criação de ambientes 3D etc...) foram feitos alguns testes sobre cada uma destes componentes para que nos familiarizássemos com estes.

Primeiro criou-se uma instância do *Moodle* numa máquina, para que fosse mais fácil testar o H5P e todas as funcionalidades com que os alunos eventualmente iriam interagir.

4.1 H5P

Com a ajuda do programa *Bitnami*, foi possível criar uma instância do *Moodle* na nossa máquina. Na criação do *Moodle*, foram criadas duas contas diferentes: um administrador, que iria ficar encarregue da construção dos exercícios; e o aluno, que serviria de simulação para o que os alunos iriam eventualmente ver.

Depois da construção deste, seguiu-se para a testagem do funcionamento das ferramentas do H5P no ambiente do *Moodle*.

4.1.1 Detalhes H5P

Como nenhum tipo de conteúdo do H5P tinha exatamente todas as funcionalidades que se queria, foi decidido fazer alguns testes apenas em *Javascript* e HTML (*HyperText Markup Language*) para que se conseguisse descobrir se era possível criar um ambiente como o que tinha sido proposto.

Estes testes tinham como objectivo a observação da validade de várias funcionalidades no *Javascript*, mais concretamente:

- A criação de um espaço de 360°, com a adição de um “chão”, ou seja, a criação de uma esfera que servisse de *background* e uma superfície plana.
- A adição de vários objetos diferentes nesse espaço
- A introdução de um movimento de câmara que seguisse um percurso pré-definido sobre as formas mencionadas acima
- A possibilidade de detectar a interação com os objetos, por exemplo de detectar quando um objeto está a ser clicado pelo utilizador.

Estes consistiram na construção de uma esfera para servir de ambiente e de um conjunto de objetos dentro dessa esfera (figura 4.1). A câmara depois observava um objeto de cada vez antes de mover para o próximo, numa ordem pré-definida.

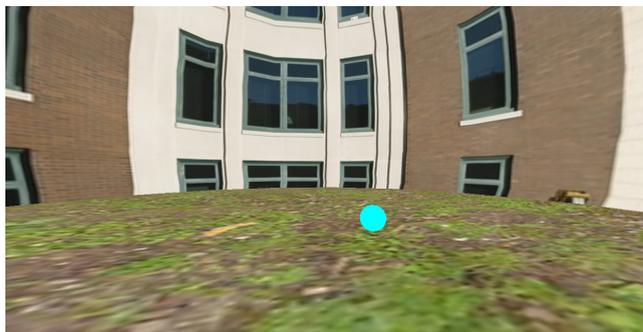


Figura 4.1: Um dos primeiros ambientes criados. Nota-se a existência de um chão, uma imagem de fundo e um objeto no canto inferior direito que seria observado pela câmara

Quando a funcionalidade de todos estes aspectos foi verificada, prosseguiu-se à construção de um ambiente para a construção de exercícios de aprendizagem RV (*Realidade Virtual*).

4.2 Projeto em *Javascript*

O projeto foi dividido em 9 classes diferentes:

Marks contendo a definição e construção de todos os objetos possíveis de serem incluídos na atividade .

GraphicalBackend representando a renderização do ambiente e dos objetos neste. Este pode ser alterado para vários ambientes gráficos, no entanto neste projeto apenas foi implementado o sistema gráfico de *Three.js*.

ThreeJsBackend contendo todas as funcionalidades gráficas impostas pelo [*GraphicalBackend*], mas realizando estas usando as bibliotecas do *Three.js*.

3D_Environment que constrói o terreno, mais especificamente o chão do ambiente, sendo este responsável por criar as elevações e aplicar uma textura neste.

GraphicalFrontend que irá fazer a atualização de todos os parâmetros em cada *frame*, tal como a posição da câmara e o tempo que o utilizador estiver a olhar para um determinado objeto.

Activity seria responsável pela organização da ordem dos objetos, sendo que se um determinado objeto fosse observado durante um determinado tempo, a câmara não iria para a posição deste, mas para uma posição pré-determinada nesta classe.

Exercise sendo responsável por definir a estrutura dos exercícios, tal como a ordem em que estes devem ser percorridos.

Student regista os eventos realizados pelo utilizador e a hora em que estes foram levantados, sendo que os únicos eventos existentes atualmente são de responder a uma pergunta e de entrar ou sair de uma pergunta.

Controller servindo de organizador de classes, sendo responsável por ligar o objeto retornado pelo *frontend* depois deste ser observado por um determinado tempo e a posição para onde a câmara deveria ir. Também é responsável por definir os *eventListeners* de maneira a detectar o movimento do cursor e outros eventos.

4.2.1 Superclasses

O ficheiro *Marks.js* e o ficheiro *GraphicalBackend.js* têm ambos superclasses, isto é, ambos contêm classes que delimitam todas as funções e variáveis que as suas subclasses terão em comum.

Mais especificamente, na classe *Marks*, delimitam-se as formas, localização (coordenadas x,y e z) e cor. Na classe *GraphicalBackend* delimitam-se as funções *initScene*, *setScene*, *togglePlayerCamera*, *togglePhoneCamera*, *drawObj* e *areObjectsBeingTargeted*, enquanto que as suas variáveis são *scene*, *camera*, *renderer*, *raycaster*, *mouse*, *floor* e *background*.

4.2.2 Funcionamento do projeto

O projeto realizado funciona da seguinte maneira: primeiro são definidos todos os exercícios (incluindo os objetos que fazem parte de cada um), seguidamente é criado o ambiente no ficheiro que serve de *backend* (que neste caso é o *ThreeJsBackend*), primeiro chamando o *3D_Environment* para construir o relevo e depois criando uma mira para dar a entender a posição do centro do ecrã. De seguida, o *Controller* chama o ficheiro de *backend* para desenhar cada objeto que é necessário incluir na cena. Segue-se então o início do *loop* de animação em que a cena, os objetos e a câmara são renderizados e é verificado se a câmara ou o cursor estão sobrepostos num dado objeto.

Interação do utilizador

Até agora tem-se explicado a parte estática do projeto, ou seja, a parte que não muda com o utilizador. No entanto, neste projeto o utilizador tem a possibilidade de interagir com este e de alterar a maneira de como

este é apresentado. Isto é realizado com a interação do utilizador com os objetos criados no ambiente (sem contar com os objetos que compõem a cena, sendo estes a esfera que representa o espaço envolvente, o plano que representa o chão e os objetos dos quais a mira é composta).

A detecção de objetos foi implementada tanto em *desktop* como em dispositivos móveis.

No *desktop* existem duas formas de detectar objetos:

- Se o utilizador clicar no ecrã, o cursor ficará invisível e a câmara seguirá o movimento deste até que o utilizador clique novamente no ecrã ou que este pressione a tecla “*Esc*” no seu teclado. Neste modo, a detecção dos objetos é feita pela posição da câmara, que neste modo será o centro do ecrã. De modo a que o utilizador saiba a posição exata da câmara (visto que o cursor não está visível, não existe por padrão uma maneira deste saber esta), foi criado uma mira no meio do ecrã para demonstrar a posição exata do centro do ecrã.

- Se o utilizador não clicar no ecrã, a detecção dos objetos é feita pela posição do cursor.

Em ambientes móveis, a única maneira de detectar objetos é de clicar no ecrã, e da mesma maneira que a câmara seguia a posição do cursor no *desktop* quando este clicava no ecrã, a câmara segue a orientação do dispositivo móvel.

Um objeto é considerado detectado quando a câmara ou o cursor (conforme os modos explicados acima) se sobrepõem a um objeto interagível no espaço de tempo de 6 segundos.

O que acontece quando um objeto é detectado?

Se a câmara ou o cursor se sobrepuserem num objeto durante menos que 6 segundos e deixarem de se sobrepor, nada acontecerá e terão que voltar a selecionar o objeto desejado. Se um objeto for selecionado por 6 segundos consecutivos, no entanto, a/o câmara/cursor deixará de ser controlada pelo utilizador temporariamente e acontecerá uma pequena animação em que a câmara ir-se-á mover para uma posição diferente, esta sendo relevante ao objeto detectado e mantendo-se sempre a uma altura relevante ao chão (até que chega à posição do objeto em si, a câmara sendo então elevada para a altura do objeto), sendo que de seguida o utilizador terá novamente o controlo da câmara.

Neste momento os únicos objetos que podem ser detectados são as respostas da pergunta em que o utilizador se encontra atualmente.

4.2.3 Como é realizado o movimento da câmara?

De maneira a conseguir-se alterar a posição da câmara de uma maneira em que não fosse muito desorientante para o utilizador, ou seja, de maneira a que a transição da câmara de uma posição para outra fosse o mais suave quanto possível, foram usados *tweens*. *Tweens* essencialmente são o cálculo de números intermédios entre dois valores num determinado espaço de tempo, por exemplo, se quiser mudar a posição da câmara do ponto $x=0$ para o ponto $x=6$ num espaço de 6 segundos, sabe-se que em cada segundo a câmara estará no ponto correspondente aos segundos passados desde o começo, mas todos os pontos do meio terão que ser calculados. Usando a biblioteca *Tween.js* permite-nos fazer isso mesmo, alterar ligeiramente a posição da câmara em cada *frame* enquanto ainda nos permite orientar a posição para onde a câmara está a olhar em cada um destes, sendo que se não se alterasse a posição do olhar, quando a câmara parasse de se mover, esta iria olhar de repente para o objeto em questão, tornando esta transição muito desorientante, especialmente se o utilizador estiver a ver o projeto por óculos RV.

4.2.4 Uso de Backend

De maneira a conseguir acomodar o uso de vários subsistemas gráficos, foi criada uma classe *GraphicalBackend* que servirá de classe mãe para todos os subsistemas gráficos incluídos, contendo esta todas as classes e variáveis que se considerem globais em termos dos subsistemas gráficos (por exemplo, todos os subsistemas terão que desenhar os objetos na cena, todos eles terão uma câmara e um renderizador, etc...).

Como previamente mencionado, futuramente é esperado incluir vários subsistemas gráficos de maneira a fornecer opções a quem queira criar uma lição neste sistema. No estado atual do projeto o único subsistema gráfico que foi incluído foi o *THREE.js*.

Devido a cada subsistema gráfico ter maneiras diferentes de desenhar objetos e luzes nas cenas, diferentes variáveis que representam a posição da câmara e outras definições importantes, apenas indicámos na classe *GraphicalBackend* os nomes das diferentes variáveis e funções que terão que ser usadas. Isto foi feito de maneira a que as outras classes que não a que indica o *backend* consigam funcionar sem que sejam necessárias alterações, ou seja, quando uma classe que indica um subsistema gráfico que não *Three.js* é adicionada, apenas é necessário indicar como esta faz as tarefas definidas em *GraphicalBackend*.

Backend Three.js

Este foi o subsistema escolhido para a criação do nosso projeto. Este subsistema contém uma enorme variedade de classes, incluindo classes para desenhar objetos específicos, como esferas (*SphereGeometry*), planos (*PlaneGeometry*) entre outros. Houveram no entanto quatro classes que foram muito importantes para a criação da interação do utilizador com estes objetos: *Raycaster*, *DeviceOrientationControls* e *PointerLockControls* e *VRButton*.

A classe *raycaster* é a que verifica a posição de um vector (podendo este ser a posição do cursor ou qualquer outra posição) com a posição dos objetos fornecidos a esta (normalmente por uma lista de objetos). Para a nossa implementação, criaram-se duas opções de *Raycaster*: ou pela posição do cursor (caso o utilizador ainda não tenha clicado no ecrã) ou pelo centro exato do ecrã, representado em código por um vector 2D vazio (caso o utilizador já tenha clicado no ecrã).

As classes *DeviceOrientationControls* e *PointerLockControls* são as classes que estão encarregues de, como os nomes implicam, orientar a câmara com o dispositivo (quando o projeto é aberto num dispositivo móvel) e bloquear o movimento do cursor (quando o projeto é aberto num computador), sendo assim possível movimentar a orientação da câmara com o movimento do dispositivo ou com o cursor sem ter receio de que este saia da janela. O bloqueio do cursor é ativado com um clique, sendo que depois deste evento o cursor desaparece (embora ainda esteja na mesma posição mesmo que se mova o cursor), sendo esta a razão pela qual foi incluído uma mira de maneira a facilitar a orientação do utilizador para onde este está a olhar, sendo que sem este o utilizador teria que utilizar métodos externos para descobrir para onde a câmara estava a olhar exatamente. De maneira a desbloquear o cursor existem dois métodos: ou o utilizador volta a clicar novamente, ou prime a tecla "Esc" no seu teclado. O controlo da câmara em dispositivos móveis funciona da mesma maneira, mas ao clicar diretamente no ecrã, tanto para bloquear e desbloquear esta.

A classe *VRButton* é a classe que nos permite usar as funcionalidades RV, sendo que esta está encarregue de, como o nome indica, criar um botão na página que, quando pressionado pelo utilizador, transforma a aplicação de maneira a que esta possa ser vista como um ambiente de realidade virtual.

4.3 Implementação do projeto como uma atividade no *Moodle*

Depois de criado o ambiente RV, o próximo passo foi a implementação deste como uma atividade H5P no *Moodle*, de maneira a que alunos pudessem interagir com este. Para atingir este objectivo, foram realizados 2 passos:

4.3.1 Passo 1: criação de uma atividade a partir de ficheiros

Devido ao facto das atividades de *iframe* do H5P não aceitarem ficheiros *.html* ou *.js* (sendo estes extremamente importantes no nosso projeto devido a este consistir apenas destes tipos de ficheiro), antes de criar a atividade H5P, tivemos primeiro que criar uma atividade interativa a partir de ficheiros. A criação desta segue-se pelos seguintes passos:

1. Cria-se um ficheiro *.zip* com todos os ficheiros necessários para o funcionamento do nosso projeto.
2. Na área das atividades no *Moodle*, liga-se o modo de editar (é um botão azul no canto superior direito da página) e arrasta-se o ficheiro *.zip* para o tópico onde o quer adicionar.
3. Clica-se na opção de criar uma nova atividade.
4. Depois de adicionar o ficheiro, clica-se na nova atividade, clica-se no ficheiro adicionado e selecciona-se *unzip* para retirar as pastas do projeto.
5. Quando todas as pastas tiverem sido retiradas, pode-se então eliminar o ficheiro *.zip*. Selecciona-se o ficheiro *.html* que serve de ligação em todos os outros e clica-se no botão que diz *Set as main file*.
6. De seguida, basta escolher o modo de “Aparência” do resultado, escolhendo *Open*, fazendo com que seja aberta uma nova página em *fullscreen* quando é aberta esta atividade.
7. Por fim, clica-se no botão *Save and Display* para se guardar o progresso e ver o conteúdo criado ou no botão *Save and return to course* para guardar o progresso e voltar para a página do curso.

4.3.2 Passo 2: criação da atividade *iframe* do H5P

Agora que já foi criada uma atividade no *Moodle*, pode-se então criar uma atividade *iframe* que usará a anterior como base. A criação desta segue-se com os seguintes passos:

1. Clicar com o botão direito do cursor na atividade previamente criada e escolher “Copiar Ligação”.
2. Clicar no botão *Add an activity or resource*.
3. Clicar na opção *Interactive Content*.
4. Na lista de bibliotecas, escolher a *Iframe Embedder*.
5. Preencher todos os campos requeridos: *Title*(Título), *Width*(Largura), *Minimum width*(Largura Mínima), *Height*(Altura).
6. No último campo, *Source*, cola-se o *url* da atividade anterior que foi guardada.

7. Por fim, clica-se no botão *Save and Display* para se guardar o progresso e ver o conteúdo criado ou no botão *Save and return to course* para guardar o progresso e voltar para a página do curso.

Se o/a administrador/a desejar alterar o exercício, de maneira a atualizar este no *Moodle*, poderá simplesmente ligar o modo de edição (botão azul no canto superior direito do curso onde quer alterar o exercício), clicar em editar e copiar para a secção dos ficheiros os novos ficheiros com as alterações feitas. Depois destas alterações feitas, deve guardar estas ao clicar num dos dois botões para salvar o progresso no fim da página.

4.4 Como correr o projeto

4.4.1 Abrir o projeto sem RV

Embora o objectivo inicial do projeto tenha sido de incluir funcionalidade RV, este também funciona sem esta. Para isto, basta criar um servidor HTTP (*Hyper Text Transfer Protocol*) e de seguida abrir o *browser* na página em que o servidor foi aberto.

4.4.2 Abrir o projeto com RV

Para abrir o projeto sem RV era apenas preciso criar um servidor HTTP. Se quiser ver este através de RV, ter-se-á que criar um servidor HTTPS (*Hyper Text Transfer Protocol Secure*). Estes requerem o uso de certificados especiais (chamados certificados SSL (*Secure Sockets Layer*)), sendo que estes podem ser assinados por uma entidade de autoridade (CA) ou podem ser "certificados auto-assinados", ou seja criados antes da ligação do servidor pelo próprio programador, este último tipo de certificado não sendo muito seguro fora de testes. Estes certificados são utilizados para verificar que o *website* é de confiança.

4.4.3 Possíveis problemas com a observação do projeto

Os *browsers* hoje em dia, de maneira a evitar carregarem os mesmos ficheiros vezes e vezes sem conta, guardam alguma informação dos documentos abertos no *browser* em si. Isto pode criar problemas caso o código seja alterado, sendo que neste caso as mudanças não serão visíveis.

De maneira a resolver isto, basta ir às "Definições" do seu *browser* e limpar os *cookies*, mais especificamente, os *cookies* de dados de sites.

4.4.4 Estado intermédio do projeto

Antes de olharmos para como o trabalho ficou no final, vamos primeiro observar o estado intermédio do projeto (figura 4.2) para melhor observarmos a evolução deste. Este pode ser observado na figura abaixo.

Este consiste nas seguintes funcionalidades:

Um *background* de uma única cor, criado com HTML



Figura 4.2: Estado intermédio do projeto

Um chão com “solavancos”, de maneira a simular a textura de um terreno normal. Este foi adaptado para a versão mais recente de Three.js do trabalho *Low-Poly Scene (Three.js)*, realizado por *Douglas Duhaime*.¹

Vários objetos 3D como esferas e cubos, espalhados pelo chão.

Uma versão inicial de um objeto *Question*, que consiste em: uma caixa de texto representante da pergunta em si; duas caixas de texto debaixo da pergunta a representar duas possíveis respostas; os selecionadores de resposta, estando estes ao lado de cada resposta respectiva, sendo estes a única parte do objeto *Question* com que o utilizador pode interagir.

Uma mira representante do centro da câmara que, quando é movimentada para um objeto para o qual a câmara pode ser deslocada, pulsa (ou seja, aumenta e diminui ligeiramente de tamanho), roda em direção de relógio e muda de cor, para indicar ao utilizador que este está a observar algo interagível.

Um deslocamento uniforme da câmara para uma nova posição quando um objeto interagível é observado durante 6 segundos.

4.4.5 Estado final do projeto

Finalmente, iremos olhar para o estado final deste projeto (figura 4.3) e explicar as diferenças deste em relevância ao seu estado intermédio.

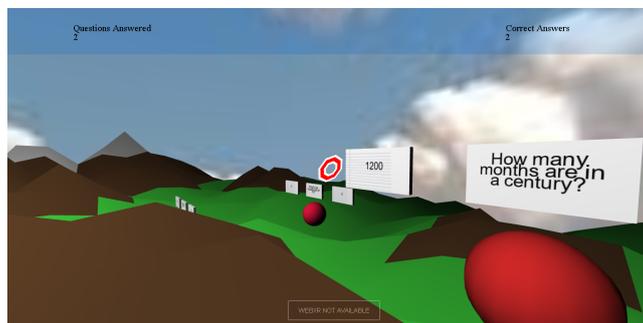


Figura 4.3: Estado final do projeto

¹<https://bl.ocks.org/duhaime/4131d0c02ef02257a880eb912767b8dc>

Features finais:

O terreno foi severamente expandido e alterado, de maneira a ter mais relevo e elevação, de modo a que se pareça mais com a realidade.

Devido à nova elevação do terreno, o movimento da câmara foi alterado para levar em conta a elevação deste quando esta se desloca de um ponto a outro, ou seja, quando a câmara se mover de um ponto para outro estará sempre à mesma elevação em relação ao terreno, sendo que quando chegar ao ponto de destino será elevada para a altura da nova pergunta.

Os objetos “*Question*” foram mudados, sendo que estes agora consistem numa caixa de texto que contém a pergunta em si, uma resposta de cada lado da pergunta (que podem ser interagidas diretamente, sem necessitar de um objeto externo) e de um objeto debaixo desta.

Quando uma questão é acertada, a câmara move-se para a próxima (caso ainda não esteja na última), no entanto se errar, é movida para a questão anterior (caso não esteja na primeira).

A mira agora desaparece quando estamos a selecionar uma resposta. Para demonstrar quanto tempo temos para selecionar a pergunta, em vez da mira pulsar e mudar de cor, agora é um objeto que aparece a envolver a pergunta que tem essas propriedades.

Foi adicionada uma luz num ponto no céu de maneira a não estar tudo iluminado de uma maneira homogénea.

O material de certos objetos (maioritariamente os objetos debaixo das perguntas) foi mudado de maneira a conseguir produzir sombras.

Foi adicionado um “*background*” com a textura de nuvens, devido ao facto de, quando ativámos o modo RV em dispositivos móveis, todo o tipo de CSS (*Cascading Style Sheets*) deixa de ser observável, logo a cor de “*background*” também não aparece com CSS.

4.4.6 Explicação das funções no código

Além do ficheiro “Main.js” e “Marks.js”, todos os ficheiros do projeto têm funções que realizam determinadas tarefas. Nesta secção cada função será explicada de uma maneira sucinta. Segue-se o diagrama de classes do projeto:

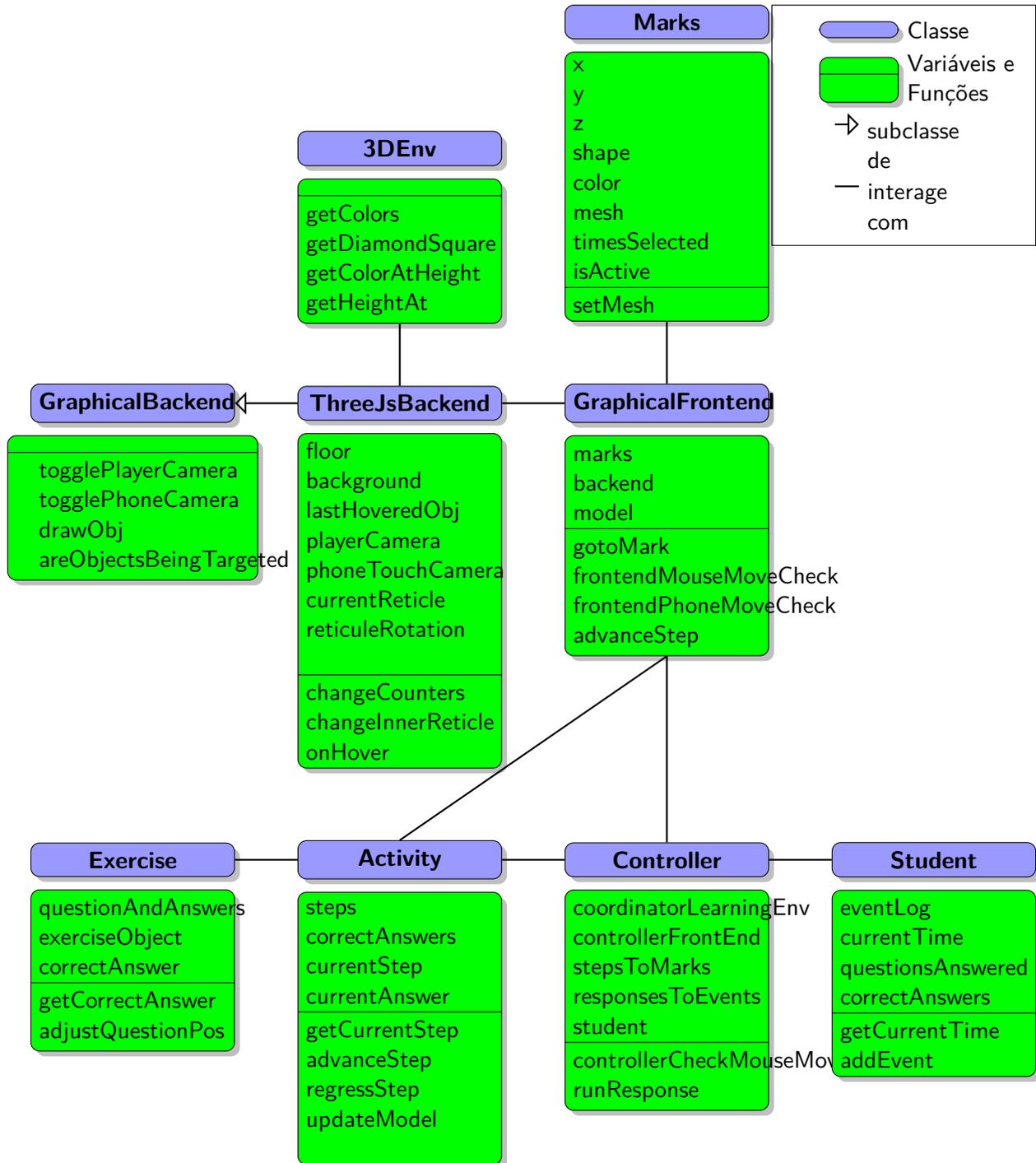


Figura 4.4: Diagrama de classes

3D_Environment

Esta classe está encarregue de construir o terreno, incluindo as suas dimensões, a sua elevação e a textura deste em diferentes pontos.

getDiamondSquare - Cria a forma do terreno. Este é criado por primeiro correndo o algoritmo de *Diamond Square*. De seguida, retira-se as bordas deste (para termos um terreno mais circular). Seguimos para a escala dos pontos para que estes tenham alguma elevação. Depois, verificamos se cada ponto não tem uma altura demasiado diferente dos pontos adjacentes (para que o terreno seja mais suave). Finalmente, verificamos se um pré-determinado número de pontos tem uma dada altura, sendo que caso isto não se verifique, o terreno é computado novamente.

rand - Retorna um número randomizado, entre 0 e um dado limite

mean - Retorna a média de um dado *array*

initialize - Usando o algoritmo de *Diamond Square*, cria a forma básica do terreno.

render - Coloca a textura em cada ponto do terreno, baseado na sua altura.

getColorAtHeight - Retorna uma determinada cor, dependendo da altura que é dada.

getHeightAt - Retorna a altura num dado ponto.

Controller

Age como centro de controlo entre todos os outros ficheiros. Por exemplo, quando uma pergunta é respondida, este verifica através da *Activity* se a resposta foi correta, sendo que vai comandar as mudanças necessárias em *Activity* e em *GraphicalFrontend* para que seja tudo adaptado à nova pergunta.

controllerCheckMouseMove - Sempre que o cursor é movido no *browser*, esta função detecta se a mira neste momento está sobre um objeto interativo. Se sim, chama outra função para tratar dos eventos.

setup - Cria os *eventListeners* de maneira a detectar o movimento do dispositivo móvel e o movimento do cursor no *browser*. Também trata da permissão das funcionalidades RV, tomando atenção ao subsistema gráfico que está a ser usado.

runResponse - Quando uma pergunta é respondida, esta função determina se a resposta foi a correta ou não. Dependendo se a pergunta foi acertada ou não, esta função determina para que pergunta a câmara irá ser deslocada, atualiza os contadores de *Questions Answered* e *Correct Answers* e adiciona novos eventos ao *eventLog* do utilizador.

Graphical Frontend

Lida com as mudanças visuais do ambiente que não implicam a construção de novos objetos, tais como a movimentação da câmara, por exemplo.

animationLoop - Trata de todas as atualizações que se fazem *frame a frame* (por exemplo, atualiza todos os Tweens, verifica à quanto tempo um objeto está a ser selecionado, renderiza o ambiente e autoriza a funcionalidade RV).

gotoMark - Cria os *tweens* que irão mover a câmara para uma posição intermédia entre a posição atual desta e a posição do objeto agora selecionado.

frontendMouseMoveCheck/frontendPhoneMoveCheck - Chamam as funções respectivas do backend para obter a posição atual do cursor no *browser*/dispositivo móvel

setup - Guarda os *meshes* dos objetos que podem ser selecionados para que estes possam ser acedidos com mais facilidade no futuro e adiciona ao ambiente os *eventListeners* que permitem ligar o movimento da câmara num *browser* e num dispositivo móvel. Também adiciona um *eventListener* para detectar quando o ecrã é redimensionado.

advanceStep - Muda os objetos que podem ser selecionados pelo utilizador, caso este vá para um exercício diferente. Nesta versão do projeto, os objetos que podem ser selecionados são as respostas do exercício para o qual o utilizador se está a mover.

Graphical Backend

Serve de “esqueleto” para os subsistemas gráficos. Inclui todas as funções e variáveis de que possivelmente todos os subsistemas gráficos partilham. Por exemplo, todos os subsistemas gráficos precisam de uma câmara e de um *renderer*.

initScene - Inicializa vários parâmetros importantes para o *backend* gráfico, como a cena, a câmara, o renderizador, o *raycaster*(usado para detectar luz) e ajusta todos estes parâmetros ao domínio do HTML.

setScene - Cria a interface, mais especificamente, introduz no ambiente um plano transparente para servir de indicador do desempenho nos exercícios (quantos foram respondidos no total e quantos foram acertados) e introduz também dois objetos em forma de anel para servirem de mira. Adiciona também os controlos da câmara no *browser* e do dispositivo móvel(*PointerLockControls* e *DeviceOrientationControls*, respectivamente).

togglePlayerCamera/togglePhoneCamera - Detectam quando a câmara é ativada no *browser*/dispositivo móvel(ao clicar com o cursor ou tocar no ecrã, respectivamente).

drawObj - Desenha um determinado objeto interativo na cena. A elevação deste vai ser relativa à elevação do plano no ponto onde este objeto está a ser desenhado.

onMouseMove/onPhoneMove - Obtêm a posição atual do cursor no *browser*/dispositivo móvel.

areObjectsBeingTargeted - Verifica se a mira está sobre um objeto interativo. Se sim, cria uma borda à volta da pergunta que começa pulsar e mudar de cor. Isto é feito através da criação de vários *Tweens* que serão atualizados no *animationLoop* no *GraphicalFrontend*.

ThreeJsBackend

Constrói o ambiente fisicamente, isto é, dá forma a tudo desde o terreno aos exercícios. Por exemplo, decide a posição de uma determinada pergunta, qual a forma que terá e a textura que será aplicada nesta.

setDisplay - Cria uma barra transparente no topo do ecrã do dispositivo que conterà o número de perguntas respondidas e o número de respostas acertadas, sendo estes valores actualizados ao longo do percurso pela atividade.

changeCounters - Atualiza os contadores de *Questions Answered* e *Correct Answers* sempre que uma pergunta é respondida pelo utilizador.

resizeWindow - Redimensiona a janela quando a janela do *browser* é alterada (ex: quando passa de modo paisagem para modo de fotografia num dispositivo móvel).

fixText - Formata uma dada *string* para que esta não ultrapasse o limite de um dado comprimento, sendo que se ultrapassar, é adicionada uma nova linha e continua-se a escrever o resto do texto e por aí adiante.

changeInnerReticle - Altera o estado da borda da resposta a ser seleccionada. Esta pode passar do estado de “pulsar e mudar de cor” para “parada” e vice-versa.

onHover - Altera uma variável para que se saiba se um objeto está a ser observado mas ainda não passou tempo suficiente para a câmara se mover, se nenhum objeto está a ser observado ou se um objeto está a ser observado e a câmara está a mover-se.

Activity

Decide o percurso do utilizador conforme este responde às perguntas, por exemplo, se um utilizador acerta na pergunta em que está atualmente, *Activity* decide para que pergunta este irá a seguir.

updateModel - Atualiza todos os parâmetros do modelo que podem mudar. Esta função é chamada em todas as *frames* no *animationLoop* no *GraphicalFrontend*.

getCurrentStep - Retorna o exercício atual.

advanceStep

regressStep - Respectivamente avançam para o próximo exercício e retornam para o exercício anterior. Se usarmos o *advanceStep* quando estivermos no último exercício, este não faz nada e o mesmo se passa com o primeiro exercício e o *regressStep*.

Exercise

Define como os exercícios são estruturados.

getCorrectAnswer - Retorna a resposta correta para a questão atual.

adjustQuestionPos - Ajusta a posição da pergunta de acordo com o *exerciseObject* no exercício, para que possamos usar essa posição como referência para ajustar a posição da pergunta em si e das suas respostas.

Student

Contém toda a informação sobre o percurso que o utilizador tomou até o tempo atual, mais especificamente quando entrou, saiu e respondeu a um exercício

getCurrentTime - Retorna a hora exacta actual (retorna a hora no formato horas:minutos:segundos)

addEvent - Adiciona um novo evento ao *eventLog* actual do utilizador. Este evento terá várias informações, tal como: a hora exata atual, o tipo de evento, a resposta do evento (caso aplicável) e o exercício no qual este evento foi levantado. Atualmente existem 3 tipos diferentes de eventos possíveis: *Entered Exercise*, *Left Exercise* e *Answered Exercise*, sendo que este último é o único evento que pode ter uma resposta.

5

Resultados Empíricos

A biblioteca *Three.js* permitiu-nos criar um ambiente (ver figura 5.1) com objetos 3D com os quais conseguimos interagir com a ajuda da câmara, enquanto também permite ao utilizador visualizar este ambiente em RV (*Realidade Virtual*). Com esta também conseguimos introduzir uma luz no ambiente e também projetar sombras nos objetos.

A biblioteca *Tween.js* também nos permitiu introduzir movimento no ambiente, quer ao fazer os contornos das perguntas pulsar, quer ao movimentar a câmara de um ponto para outro.

Apenas com estas duas ferramentas seria possível criar exercícios mais complexos, tal como criar um ambiente mais parecido com o nosso objetivo principal. Por exemplo, poderíamos criar uma caixa de texto a explicar uma determinada matéria, criando um "botão" que os alunos poderiam selecionar para avançar para a parte prática, onde teriam que escolher de entre vários objetos 3D para responderem a uma pergunta, em vez de apenas escolherem entre 2 respostas escritas.



Figura 5.1: Vista de cima de todo o ambiente.

Foi possível implementar o projeto como um exercício na nossa aplicação *Moodle* (ver figura 5.2). Isto significa que futuramente este poderia ser adaptado para várias disciplinas e partilhado através desta plataforma.



Figura 5.2: Exercício na página do Moodle.

Infelizmente não foi possível incluímos o tipo de conteúdo “Virtual Tour 360°” do H5P (*HTML5 Package*) no nosso projeto, tendo em vez disso implementado o tipo de conteúdo “Iframe Embedder”. Isto permite-nos fazer do projeto um exercício H5P (que pode então ser facilmente partilhado por outras plataformas) enquanto oferece a habilidade de colocar e retirar este de “Fullscreen”.

6

Conclusão e Trabalho Futuro

O uso de RV (*Realidade Virtual*) não só pode ser aplicado ao ensino, como já tem vindo a ser testado para este propósito há alguns anos. Este permite aumentar a concentração e o interesse nas matérias representadas devido à imersão que esta fornece aos utilizadores.

Como já estabelecido anteriormente, não foi possível criar o projeto da maneira como inicialmente este foi imaginado. Inicialmente este consistiria na criação de um sistema que permitiria a criação de lições interativas, incluindo a adição de objetos (alguns de leitura ou vídeos para a explicação de certas matérias antes de autorizar a interação do aluno), criação de exercícios, etc...

No entanto, devido à complexidade e inicial inexperiência com as linguagens e bibliotecas utilizadas, no tempo estabelecido apenas foi possível criar um ambiente 3D com vários objetos, sendo que alguns destes podem ser interagidos pelo movimento da câmara, logo permitindo o uso de RV para a interação com o sistema. Também foi criado o "esqueleto" do que seria o sistema de criação de exercícios mencionado acima.

Também conseguimos atingir o nosso objectivo de criar uma atividade H5P (*HTML5 Package*) numa instância do *Moodle* (de modo a que fosse testado o que os alunos eventualmente iriam ver num ambiente

pedagógico) que usasse este sistema como base. No entanto, a eficácia destas atividades em dispositivos móveis não pôde ser testada, devido ao facto da simulação do *Moodle* ser apenas local à máquina em que era corrida, logo não podendo ser acedida por outros dispositivos.

Existem várias mudanças que se poderiam realizar a este projeto para o melhorar, sendo a mais óbvia de adaptar este trabalho a vários subsistemas gráficos, sendo este um dos objetivos sobre os quais este foi realizado. Poderíamos também testar este em várias máquinas com resoluções de ecrã diferentes, tendo assim a certeza de que todos os aspectos da UI (*User Interface*) estão adaptados para cada uma, assim permitindo a qualquer pessoa interagir com o projeto na sua máquina sem problemas. Em termos dos exercícios, poderíamos criar exercícios um pouco mais complexos, talvez com uma explicação de uma determinada matéria anteriormente, possivelmente através de um vídeo e também poderíamos criar outros tipos de exercícios que não de escolha múltipla, estes sendo um pouco limitados nas suas aplicações.

Bibliografia

- [AA14] Nelly Abaidoo and Valentina Arkorful. The role of e-learning, advantages and disadvantages of its adoption in higher education. *International Journal of Instructional Technology and Distance Learning*, 2:29–42, 2 2014.
- [CA20] James Calvert and Rhodora Abadia. Impact of immersing university and high school students in educational linear narratives using virtual reality technology. *Computers and Education*, 159, 2020.
- [JCL97] Sankar Jayaram, Hugh I Connacher, and Kevin W Lyons. Virtual assembly using virtual reality techniques. *Computer-Aided Design*, 29, 8 1997.
- [KH16] Andreas M. Kaplan and Michael Haenlein. Higher education and the digital revolution: About moocs, spocs, social media, and the cookie monster. *Business Horizons*, 59, 7 2016.
- [RMFW20] Jaziar Radianti, Tim A. Majchrzak, Jennifer Fromm, and Isabell Wohlgenannt. A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. *Computers Education*, 147, 4 2020.
- [RMR⁺19] Diego Rojo, Jesus Mayor, Jose Jesus Garcia Rueda, Laura Raya, and Mike Potel. A virtual reality training application for adults with asperger's syndrome. *IEEE Computer Graphics and Applications*, 39, 3 2019.
- [Sch02] Klaus Schwienhorst. The state of vr: A meta-analysis of virtual reality tools in second language acquisition. *Computer Assisted Language Learning*, 15, 7 2002.
- [SVO11] Abhishek Seth, Judy M. Vance, and James H. Oliver. Virtual reality for assembly methods prototyping: a review. *Virtual Reality*, 15, 3 2011.
- [Tho18] Meredith Thompson. Making virtual reality a reality in today's classrooms. 2018.
- [Yoh01] Myeung-Sook Yoh. The reality of virtual reality. *Proceedings Seventh International Conference on Virtual Systems and Multimedia*, 2001.



UNIVERSIDADE DE ÉVORA
INSTITUTO DE INVESTIGAÇÃO
E FORMAÇÃO AVANÇADA

Contactos:

Universidade de Évora
Instituto de Investigação e Formação Avançada — IIFA
Palácio do Vimioso | Largo Marquês de Marialva, Apart. 94
7002 - 554 Évora | Portugal
Tel: (+351) 266 706 581
Fax: (+351) 266 744 677
email: iifa@uevora.pt