**Vertigo**

**Software usado na aplicação:**

* Unity 5.6
* Blender
* Oculus

A aplicação Vertigo toma partido de alguns assets já existentes na store do Unity, nomeadamente o pacote “OVR”, da Oculus, para importação dos scripts que dizem respeito ao Oculus DK2; e o KinectManager, da Microsoft, para importação dos assets que dizem respeito ao sensor Kinect, desde texturas a scripts de movimento.

O script **KinectManager.cs** proveniente do pacote do sensor Kinect foi alterado internamente para ir de encontro às necessidades da aplicação. Foi adicionado um GameObject público e a interação da função Update fica dependente de uma variável global.

Os restantes scripts e texturas existentes no projeto foram construídos de raiz e serão descrevidos abaixo:

**Assets/Models\_Characters\_Objects/Test\_Cube/Scripts:**

**AvatarBoxConnection.cs –** Função usada apenas para debug, fazendo o avatar seguir um GameObject (público).

**AvatarKeyControl.cs –** Função usada apenas para debug, forçando o avatar a ter um movimento to translação, de acordo com o input de teclas (setas de direcção).

**CameraController.cs** – Script para a camera referente ao Oculus, no contexto do Unity, seguir o avatar de um modo sincronizado e fiável ao utilizador

**CollisonControl.cs** – Controla a gravidade do Avatar baseado numa variável global.

**FeetCollision.cs** – Script que deteta a posição atual dos pés do avatar, no contexto do mundo virtual, em cima da tábua e que controla o valor de uma variável global associada ao controlo da gravidade.

**GlobalVariables**.**cs** – Script para guardar as variáveis globais que são consultadas/alteradas pelos restantes scripts da aplicação

**LavaTimeStop.cs** – Script que controla a paragem da aplicação e aparecimento do popUP final de fim de jogo quando o utilizador cai da tábua

**MoveBox.cs** – Script Auxiliar de movimento do cubo de testes de movimento

**Assets/Scripts:**

**Agent.cs** - Tratamento da NavMesh dos lobos e reinicio do mesmos

**EndGamePopUP.cs** – Script que faz gestão do popUP de fim de jogo quando o utilizador completa a experiência

**RockFalling.cs** - Script de gestão das rochas na sua totalidade, desde controlo do movimento e áudio até ao reinicio das mesmas

Todas as texturas existentes no projeto, à exceção dos lobos, foi construída pelos membros da equipa usando Blender e a própria engine da aplicação, Unity.

**Equipamento necessário:**

* Kinect V1 XBOX360/Windows
* OculusRift DK2
* Tábua de 2,585m comprimento e 0,27m largura

**Key Bindings:**

* Espaço
  + Efetua o reinicio/resumo da aplicação.
  + Quando o utilizador termina a experiência, premir 1x para reiniciar a aplicação e esperar novo utilizador; premir novamente para resumir.
  + Quando o utilizador cai, premir 1x para reinicio da aplicação para o estado inicial, premir novamente para resumir.
  + Quando o utilizador sai do raio de ação do sensor, o mundo é reiniciado automaticamente; necessário premir 1x para resumir.
* K
  + Desativa o popUP de fim do jogo (caso o utilizador não entre num dos estados em que requer reinicio da aplicação; ex.: voltar para trás).
* P
  + Pausa/resume a aplicação.

**Execução:**

* Para executar a aplicação clicar no ficheiro executável: **Vertigo\_Final.exe**
* A aplicação é iniciada já em movimento, por isso posicionar um utilizador na zona de inicio antes de executar; caso não exista, premir “Espaço” 1x para reiniciar e esperar utilizador.
* Durante a execução o utilizador é livre no que quer fazer, as únicas interações requeridas do supervisor são as especificadas na secção “Key Bindings” onde na maior parte dos casos será necessário apenas usar a tecla “Espaço”.
* Quando é necessário mudar de utilizador, carregar na tecla “Espaço” 1x até que o avatar fique parado na zona de inicio. O ponto de referência deste caso serão os lobos parados e a pedra maior no topo da colina.
* Para fechar a aplicação usar a tecla "**Esc**".

**Cuidados a ter / Modos de utilização:**

* Garantir que apenas 1 utilizador está presente no raio de ação do Kinect durante a experiência. Caso necessária intervenção de um supervisor, garantir que este não se coloca à frente da camara de modo a que o esqueleto do utilizador esteja sempre visível.
* Antes da experiência, informar o utilizador de 2 casos a ter em conta:
  + Devido ao curto espaço de interação, o utilizador deve manter a calma e executar a experiência a um passo lento de modo a não chocar com a mesa onde o Kinect irá estar posicionado.
  + Informar o utilizador que deverá manter-se quieto até que detete movimento ou que comece a ouvir as pedras a cair. Deste modo é dado tempo suficiente para o supervisor sair do raio de ação do sensor Kinect e não causar interferências com a experiência (só quando a tecla “Espaço” for carregada é que a experiência começa e o Kinect deteta utilizadores”).

**Notas:**

* Para correr a aplicação é necessário ter em conta os seguintes aspetos:
  + Instalar o SDK do Kinect (desnecessário se já estiver instalado)  
    <https://www.microsoft.com/en-us/download/confirmation.aspx?id=40277>
  + Instalar a aplicação da Oculus (desnecessário se já estiver instalado)  
    <https://www3.oculus.com/en-us/setup>
  + Colocar a tábua alinhado com o Kinect dando distância de pelo menos 0,5m.
  + Ligar a aplicação: Ter em conta que a calibração do ângulo de rotação do Oculus é feita aquando do inicio da aplicação, por isso o utilizador deve-se posicionar-se virado para a câmera do Kinect até que a aplicação acabe de iniciar; isto só é necessário aquando do inicio da aplicação.

 

Fig1: Área de interação da experiência. Fig2: Posicionamento da camara na superfície face ao chão

**Contactos:** franciscoteixeira@ua.pt [luismiguelgameiro@ua.pt](mailto:luismiguelgameiro@ua.pt)