

Desenvolvimento de Videojogos

Licenciatura em Engenharia Informática – 2020/2021

Guia de Laboratório nº.1

Unity – Interface e conceitos essenciais

Introdução e Objetivos

Neste primeiro trabalho de laboratório, pretende-se explorar o ambiente *Unity*, nomeadamente a sua interface e os conceitos nela representados. Estes serão aplicados para a criação de nossa primeira cena neste ambiente.

Preparação teórico-prática

Introdução à interface do *Unity*

Ao arrancar pela primeira vez, o *Unity* apresenta a interface com o layout apresentado na Figura 1.

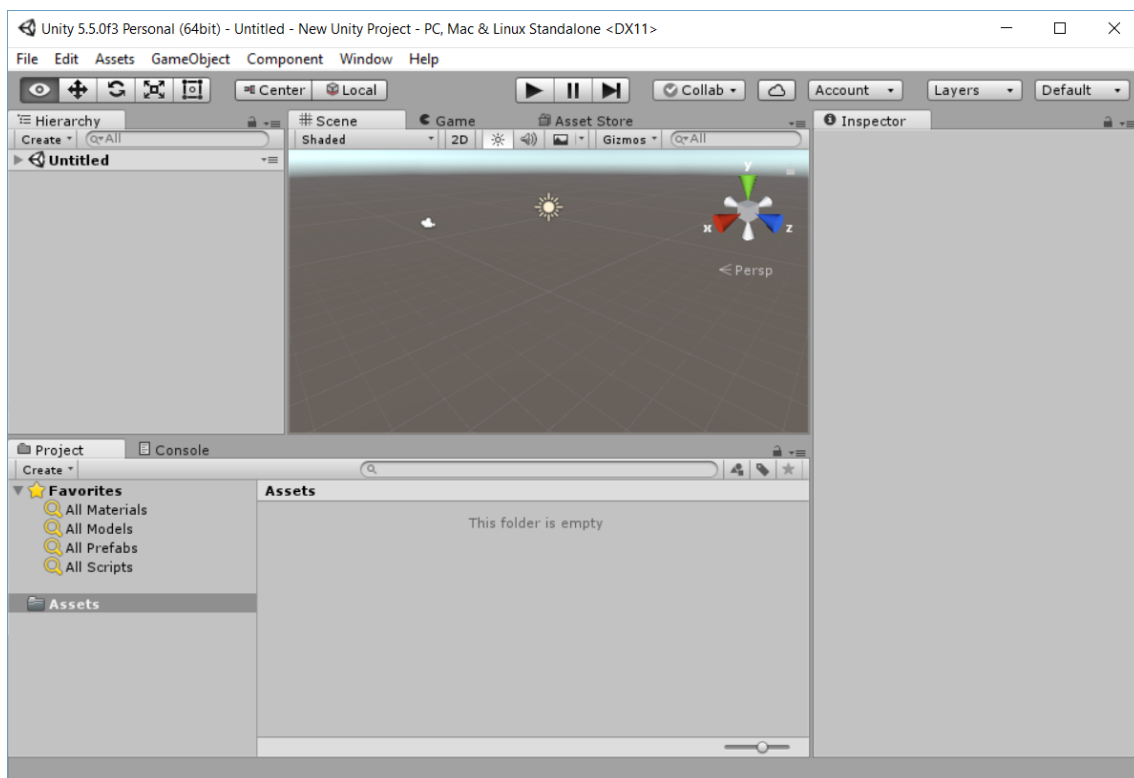


Figura 1 – Layout inicial

É possível observar a existência de 4 divisões principais, com o conteúdo que se descreve em seguida.

Janelas principais

Projeto – Project (em baixo): esta é a biblioteca de recursos a utilizar do desenvolvimento da aplicação. Estes designam-se usualmente por *Assets* e são tudo o que pode ser utilizado na aplicação, nomeadamente modelos, texturas, sons, etc.

Cena – Scene (ao centro): é o mundo da aplicação/jogo, no qual são colocados espacialmente os seus elementos a partir da biblioteca de *Assets*. Os elementos numa cena são designados por *Game Objects*. Os elementos presentes na cena são manipuláveis diretamente no editor, sendo que os principais controlos são selecionados na zona esquerda da barra de ferramentas (no topo), apresentada na Figura 2.



Figura 2 – Modos para controlo e manipulação da cena

Hierarquia – Hierarchy (à esquerda): apresenta uma estrutura lógica do conteúdo da cena através de uma hierarquia. Este tipo de abordagem é comum em ambientes tridimensionais, em que existe uma relação de pai (*parent*) e filho (*child*) entre os diversos elementos da cena. As alterações aplicadas a objeto (rodar/mover/escalar) são propagadas pelos seus filhos. Os objetos filhos utilizam definem as suas coordenadas e outras características em relação aos seus pais.

Podem ser dados alguns exemplos práticos onde esta abordagem é útil:

- Exemplo 1: um objeto “carro” tem “rodas” como objetos filho. Ao mover-se o carro as rodas seguem consigo e, da mesma forma, se quisermos modificar na cena o tamanho do carro é natural que as rodas se redimensionem na mesma proporção.
- Exemplo 2: uma câmara pode ser colocada como filha de uma personagem do jogo. Assim, a câmara move-se automaticamente seguindo a personagem.

Inspeção – Inspector (à direita): onde são apresentadas as propriedades do objeto selecionado na cena. Trata-se de uma janela de contexto, ou seja, o conteúdo depende do que estiver selecionado. Este aspeto será explorado mais tarde na disciplina, mas todos os objetos de uma cena têm uma componente de propriedades relativas às suas características espaciais, designada por *transform*). Por exemplo, se clicarmos sobre o ícone de câmara existente na cena que é criada por defeito (📷), podemos verificar as seguintes propriedades:

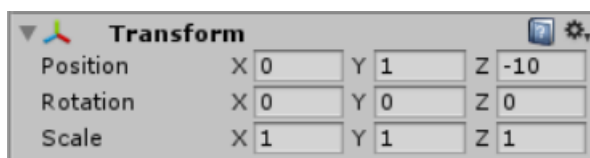


Figura 3 – Janela para configuração da componente Transform de um objeto

Todas as janelas são arrastáveis, pelo que o utilizador pode configurar o ecrã da maneira que achar mais apropriada e com a qual se sentir mais confortável.

Outras janelas importantes

É também possível associar-se uma zona do ecrã a mais do que uma janela, pela utilização de separadores. Por exemplo, é possível ver na Figura 1 que na zona central da cena estão

colocados outros dois separadores: *Game* e *Asset Store*. Adicionalmente, na zona de baixo, existe ainda o separador *Console*. Estes três separadores serão explicados de seguida.

Jogo – Game: onde é possível testar a aplicação/jogo. Este modo é controlável através dos botões apresentados na Figura 4. Um clique no botão *Play* alterna automaticamente da cena para o modo de jogo.



Figura 4 – Controlos do modo jogo

Loja – Asset Store: onde se tem acesso a um “mercado” de Assets de todos os tipos.

Consola – Console: onde são apresentadas mensagens relativas à programação da aplicação. Em modo de edição são usualmente apresentados erros de compilação do código criado (este aspeto não será ainda explorado nesta aula de laboratório). Durante a execução em modo de jogo podem ser apresentados erros causados por exceções e mensagens de *debug*.

Para além destas janelas que são apresentadas no layout por defeito, existem diversas outras janelas que podem ser colocadas na janela através do menu *Window*.

Criação de objetos a partir de primitivas

Para as primeiras experiências a realizar com o motor *Unity* é útil saber adicionar objetos simples à cena, como por exemplo um cubo. Isto pode ser feito da forma apresentada na Figura 5.

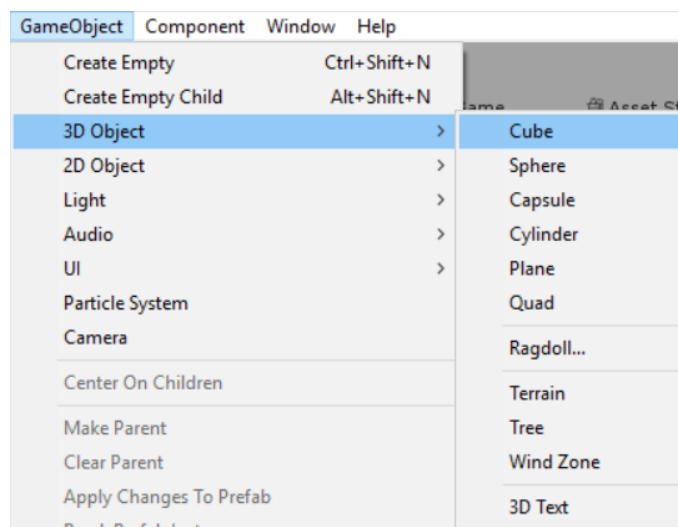


Figura 5 – Inserção de um cubo

O cubo é automaticamente posicionado na cena na direção em que esta está a ser visualizada. Não é apresentado nenhum *Asset* correspondente ao objeto, pois os elementos primários são automaticamente parte do sistema e não um recurso propriamente dito. Mais tarde veremos que é possível criar *Assets* específicos para representar certas configurações de elementos. Por exemplo, de forma muito resumida, pode ser criado um *Asset* que seja especificamente um “cubo Azul e posicionado com uma rotação de 45 graus em todos os eixos”. Este conceito é comum em *Unity* sob a designação de *Prefab*, e tem um conjunto de implicações e características que terão que ser explorados com mais detalhe posteriormente.

Por agora, é possível modificar a posição, escala e orientação deste cubo utilizando as marcas de eixo apresentadas junto ao mesmo (e recorrendo aos diversos tipos de controlo de cena apresentados atrás na Figura 2). Para uma edição mais precisa podemos também recorrer à janela de inspeção.

Asset Store

Finalmente, para ganhar motivação nas primeiras tarefas construindo desde logo cenas com bom apelo gráfico, podemos desde logo recorrer ao uso de *Assets* obtidos a partir da *Asset Store*. De momento, é aconselhável importarmos fundamentalmente modelos tridimensionais.

Em primeiro lugar, é necessário aceder à *Asset Store*, na janela respetiva dentro do editor. É possível navegar nas várias categorias para encontrar o tipo de objeto pretendido.



Por exemplo, navegando até **“3D Models/Characters/Robots”** é possível encontrar um modelo de um robot fornecido gratuitamente pela *Unity Technologies*, como se apresenta na Figura 6.

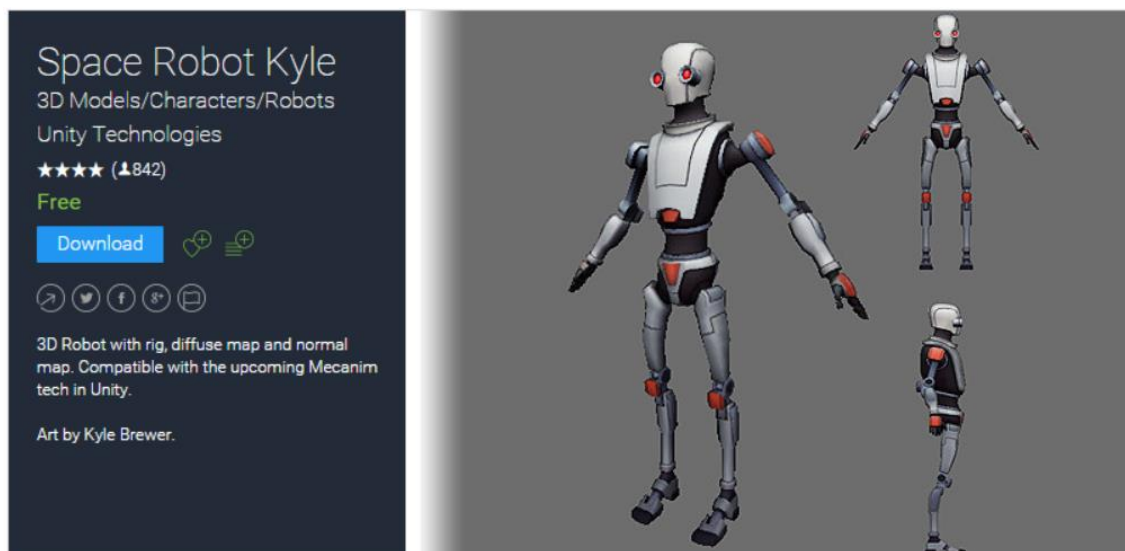


Figura 6 – Exemplo de Asset disponível na Asset Store: Space Robot Kyle

Para transferir este Asset e adicioná-lo ao projeto clica-se no botão *Download* e seguidamente em *Import*.

Na janela apresentada na Figura 7, selecionam-se todos os elementos do pacote fornecido e clica-se em *Import*.

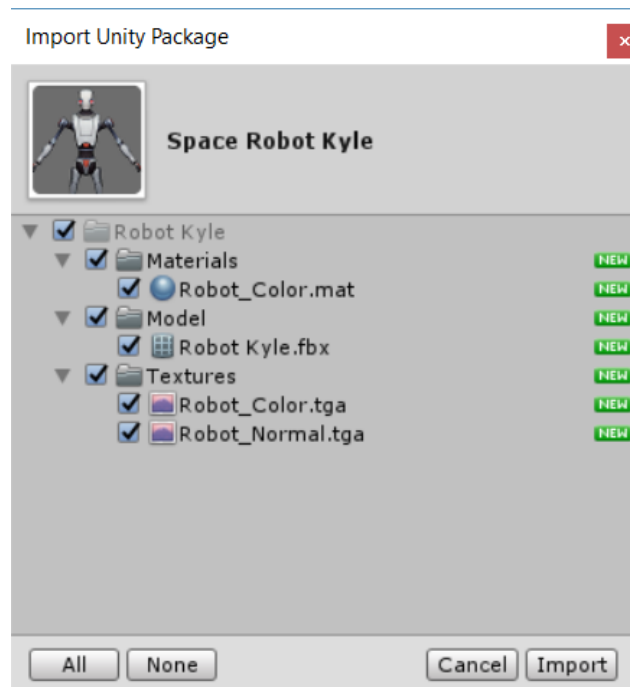


Figura 7 – Janela de importação de um pacote

Após o passo anterior, o robot já faz parte dos Assets da aplicação, como é possível observar na Figura 8

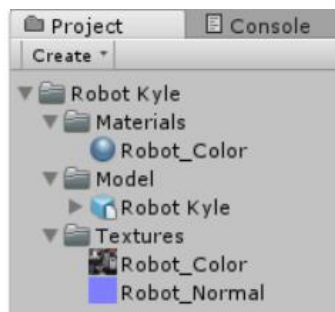


Figura 8 – Exemplo de Asset na janela de projeto após ter sido importado a partir da Asset Store

Há que ter em conta que o robot é composto por vários elementos, nomeadamente:

- *Textures/Robot_Color*, que é uma imagem com a textura que é aplicada no robot.
- *Textures/Robot_Normal*, que é um *normal map* da imagem anterior. De forma muito resumida, é uma imagem “especial” que fornece informação dos ângulos da superfície associada à textura e é utilizada no cálculo da cor do objeto em relação à luz que recebe.
- *Materials/Robot_Color* que é a especificação do material do robot. Veremos posteriormente como é feita a definição de materiais. Neste caso, o material é definido com recurso às duas texturas anteriores.
- *Model/Robot Kyle*, que é o modelo do robot em associação com o seu material.

Para adicionar o robot à cena, deve arrastar-se o elemento *Robot Kyle* para a cena e depois posicioná-lo. O resultado será algo como o apresentado na Figura 9.



Figura 9 – Exemplo de utilização em cena de um Asset importado

Neste caso, é também possível verificar-se a utilidade da utilização de primitivas simples como auxílio ao desenvolvimento. Os cubos base do *Unity* têm dimensão 1 de largura, comprimento e altura. Sugere-se que, durante o desenvolvimento da aplicação, se estabeleça a medida a que corresponde esta unidade (tipicamente 1 unidade = 1 metro). Assim, a cena tem um cubo com 1m^3 . Por observação e comparação, o robot aparenta ter uma altura apropriada (um valor entre 1.5m e 2m).

Guardar a cena

A cena que é criada é também ela um Asset. Para guardar a cena que se está a editar acede-se à opção *Save Scene* dentro do menu *File*.

Trabalho Laboratorial – Exercício

Criar uma cena semelhante à apresentada na Figura 10:



Figura 10 – Resultado final pretendido para o exercício

Os elementos utilizados na cena apresentada podem ser obtidos gratuitamente na *Asset Store*. Considere os seguintes elementos/abordagens:

- Cenário: Pode ser obtido de um ou mais packs disponibilizados. Sugere-se uma pesquisa como “*low poly environment*”, ou similar. Neste exemplo foi utilizado o pacote: <https://assetstore.unity.com/packages/3d/environments/landscapes/low-poly-simple-nature-pack-162153>
- Personagem: Na categoria “*Models/Characters/Humanoids*” há várias personagens disponíveis. Foi utilizada a seguinte: <https://assetstore.unity.com/packages/3d/characters/humanoids/1-toon-teen-135513>
- Chão: No exemplo foram utilizados os modelos incluídos no cenário. Como alternativa, é possível utilizar e redimensionar um cubo. A textura de relva foi aplicada com os seguintes passos: fazer download de uma textura (uma imagem que seja um padrão contínuo); adicionar ao projeto Unity (arrastar o ficheiro para a aba *Project*); arrastar a imagem para cima do modelo em que se pretende mudar o material (neste caso, o chão).

Material complementar para consolidação

Bibliografia

Sue Blackman, *Beginning 3D game development with unity 4 (2nd edition)*, Apress, 2013. [Páginas 22 a 47]

Tutoriais de Apoio – Complementares para consolidação

Visão geral da interface

<https://unity3d.com/pt/learn/tutorials/topics/interface-essentials/interface-overview?playlist=17090>

Vista da cena (Scene)

<https://unity3d.com/pt/learn/tutorials/topics/interface-essentials/scene-view?playlist=17090>

Vista de jogo (Game)

<https://unity3d.com/pt/learn/tutorials/topics/interface-essentials/game-view?playlist=17090>

Hierarquia e relação entre elementos na cena

<https://unity3d.com/pt/learn/tutorials/topics/interface-essentials/hierarchy-and-parent-child-relationships?playlist=17090>

Painel de projeto e importação de *assets*

<https://unity3d.com/pt/learn/tutorials/topics/interface-essentials/project-panel-and-importing?playlist=17090>

Inspector

<https://unity3d.com/pt/learn/tutorials/topics/interface-essentials/inspector?playlist=17090>

Gerar versão de distribuição

<https://unity3d.com/pt/learn/tutorials/topics/interface-essentials/build-and-player-settings?playlist=17090>

Documentação Oficial

Introdução ao *Unity*

<https://docs.unity3d.com/Manual/UnityOverview.html>