

Desenvolvimento com Motores de Jogos II

Aula 01 - Introdução ao Unity 3D

Apresentação

Olá! Na disciplina de Motores de Jogos I você aprendeu a utilizar o motor de jogos Unity3D para criação de jogos em duas dimensões (2D), além de como criar comportamentos personalizados para os elementos do jogo utilizando a linguagem de programação C# (**C Sharp**).



Você deve estar pensando que há um grande desafio pela frente e está completamente certo!

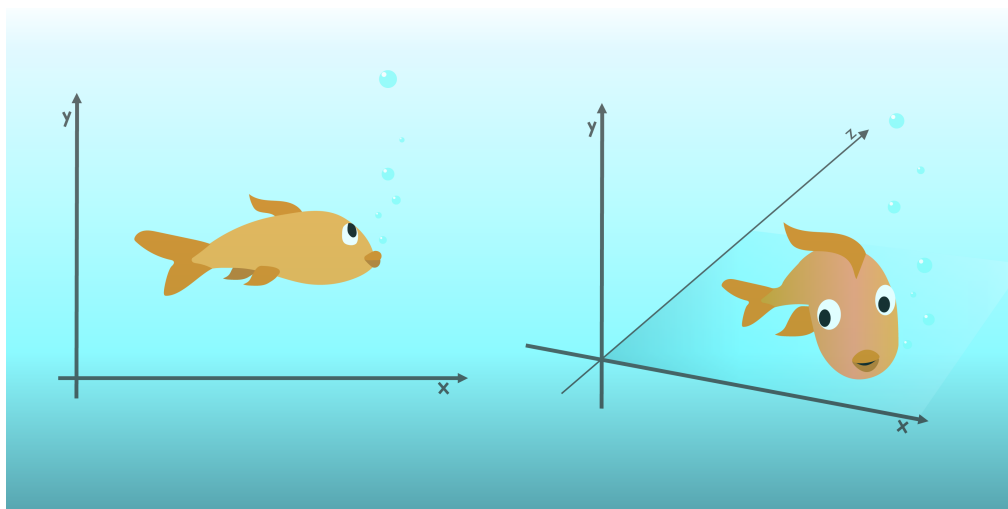
Mas fique tranquilo, estamos aqui para esclarecer todas as suas dúvidas, não deixe de partilhá-las nos fóruns ou com seus colegas no encontro presencial!

Em continuidade à disciplina anterior, em Motores de Jogos II exploraremos o mundo dos jogos em 3 dimensões, também utilizando o motor de jogos Unity.

O editor do Unity permite a manipulação dos objetos do jogo em cenas, porém, conforme visto anteriormente, para que se possa adicionar comportamentos mais complexos nesses objetos é necessário que o desenvolvedor adicione esses comportamentos através de código utilizando componentes chamados de **Scripts**. Os **scripts** no Unity podem ser criados por meio das linguagens de programação C# e Javascript. Nesta disciplina utilizaremos a linguagem de programação C# para os nossos projetos.

Entretanto, existem algumas diferenças entre a criação de jogos 2D, visto na disciplina anterior, e a criação de jogos em 3D que será abordada.

Essas diferenças, apesar de muitas, serão vistas parte por parte e em pouco tempo você já estará criando jogos em 3 dimensões! Veja nas figuras abaixo um pouco dessa distinção entre elementos criados em 2D e 3D. ;)



Nessa disciplina os exemplos estão desenvolvidos utilizando a versão 5.X da engine, que recebeu atualizações até dezembro de 2017. Recentemente o Unity recebeu um novo modo de numeração de suas versões, baseado no ano de lançamento e na sua revisão, por exemplo 2017.2.2 ou 2017.3.1, etc. Não se preocupe pois todo o conteúdo da disciplina é totalmente compatível. A versão mais recente estável no momento da última revisão deste material (01/04/2018) é a 2017.4.0. Quando for instalar o Unity em sua máquina pessoal sempre utilize a versão mais recente estável da ferramenta e evite utilizar versões beta e de teste para a criação de jogos em um ambiente de produção.

Objetivos

- Aprender elementos dos jogos em 3D utilizando o motor de jogos Unity;
- Diferenciar projetos 2D e 3D;
- Conhecer a Interface do Unity, assets e objetos primitivos;
- Aprender a utilizar o Transform (Posição, Rotação e Escala).

Iniciando com o Unity

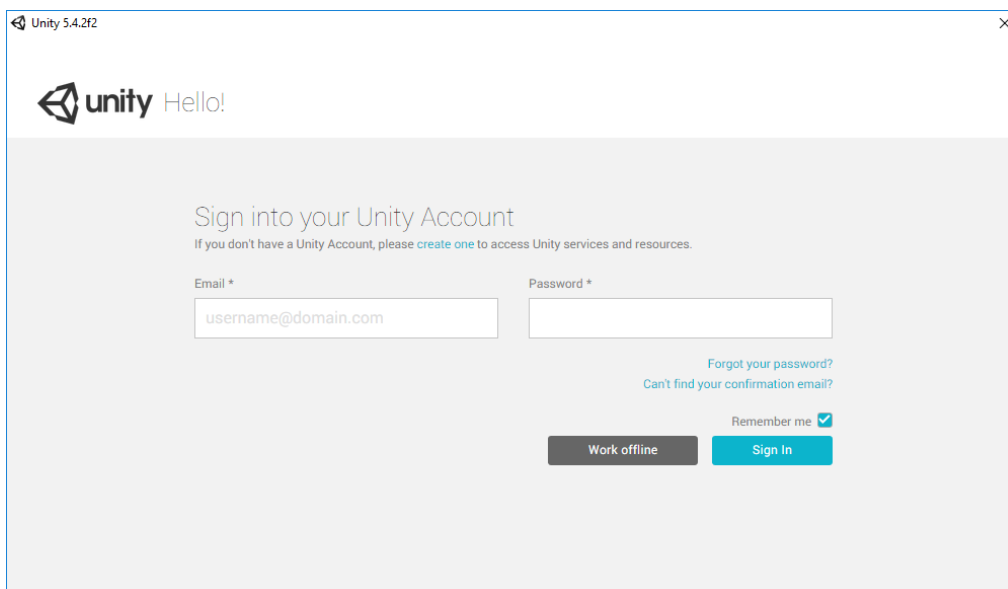
Vamos começar!

Sempre que você for criar um novo projeto no Unity é necessário que seja escolhido o tipo do projeto entre 2D e 3D. Como estudaremos o Unity em 3D, vamos criar um projeto com esse tipo.

Inicialmente, execute o Unity em seu computador e espere aparecer a tela inicial.

Vale lembrar que se você está utilizando uma nova instalação do Unity ou um novo usuário na máquina, a tela de autenticação do Unity aparecerá solicitando seu *Email* (e-mail) e *Password* (senha), conforme a figura 1.

Figura 01 - Captura de tela do Unity – Game Engine.



Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

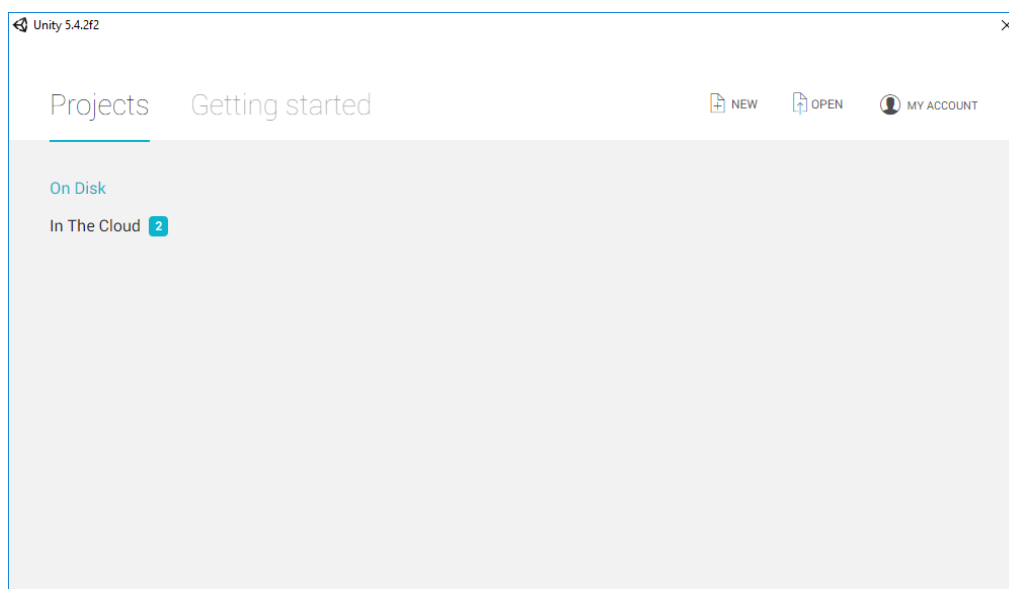
Se você não tem uma conta, clique no link “Create One” e ele o levará para o site do Unity, onde você poderá criar sua conta seguindo as instruções. Com a conta criada preencha o e-mail e a senha e clique em “Sign In”. Se você marcar a opção

“Remember Me” não será necessário digitar essas informações novamente.

Caso o Unity mostre uma tela para que escolha qual licença possui, pode escolher normalmente a opção “Personal Edition”, e logo depois escolha a opção confirmando que você não está utilizando o Unity profissionalmente ou possui ganhos menores que o limite de 100 mil dólares anuais (uma das duas últimas opções). Esses são os requerimentos para se utilizar o Unity gratuitamente.

Assim que terminar o processo de autenticação esta tela inicial do Unity deverá aparecer:

Figura 02 - Tela inicial do Unity. Captura de tela do Unity – Game Engine.



Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

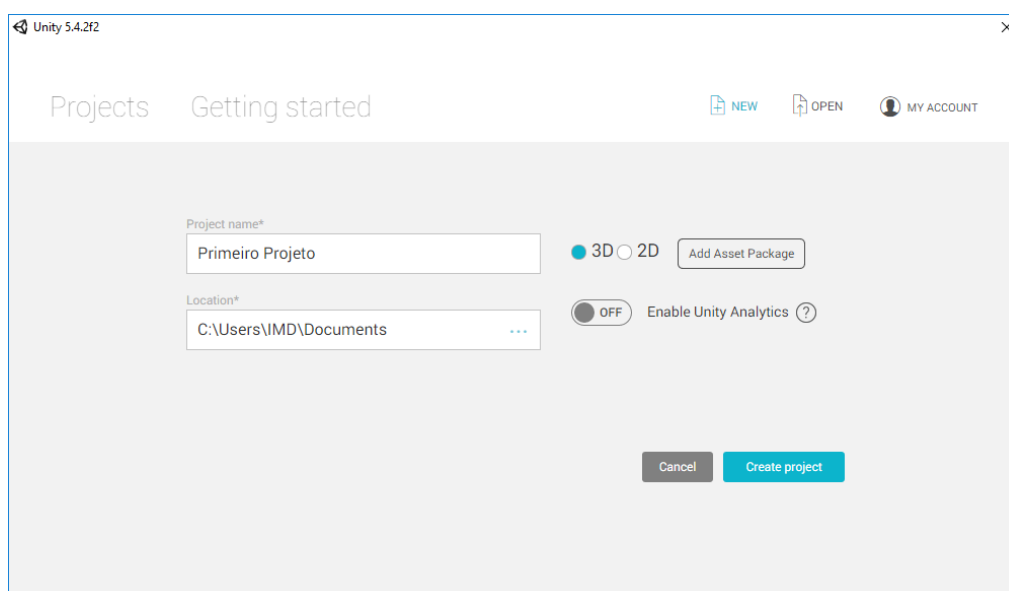
A opção “Projects” aparece selecionada por padrão. Abaixo, você verá as opções “On Disk” e “In the Cloud”. Esses dois filtros permitem listar os seus projetos, os quais estão no seu computador (On Disk) ou armazenados nos servidores do Unity na Internet (In The Cloud). Não se preocupe, falaremos mais sobre os serviços em nuvem do Unity nas aulas seguintes.

Criando um projeto

Com o Unity na tela inicial e já com o seu usuário autenticado, vamos criar um novo projeto 3D? Para isso, siga direitinho os passos necessários.

Primeiro passo: Clique no ícone com o texto “New” para criar um novo projeto. Uma nova tela aparecerá solicitando que você escolha o Project Name (nome do projeto), Location (em qual pasta será criado o seu projeto) e Organization (deixe com o valor por padrão), como demonstra a figura 3:

Figura 03 - Captura de tela do Unity – Game Engine.



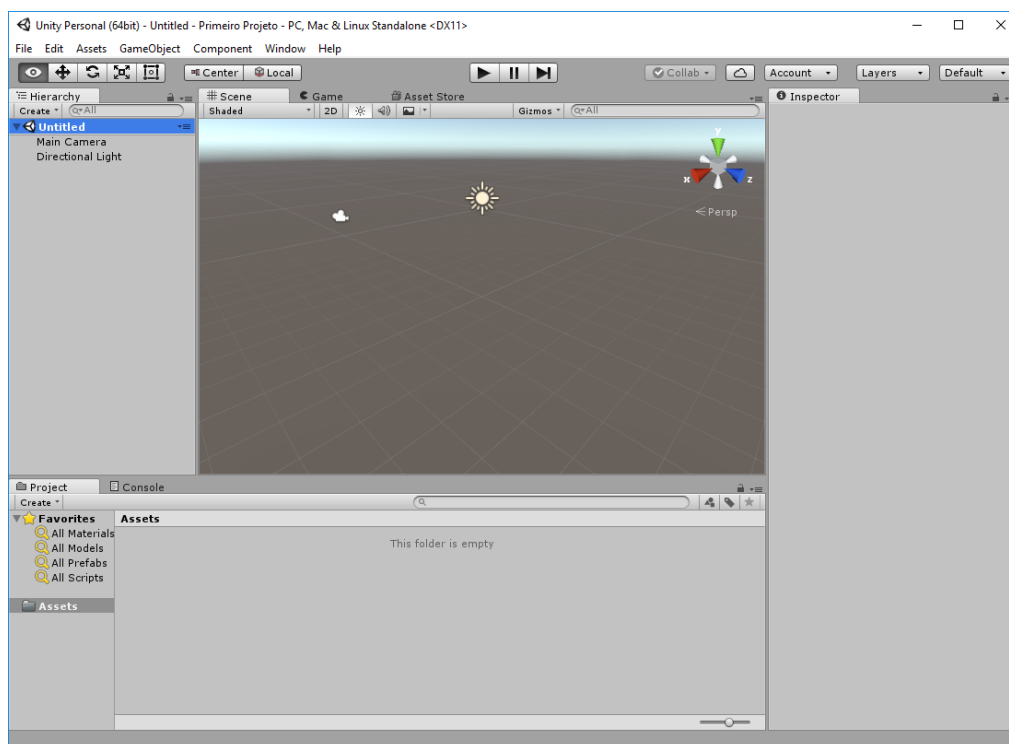
Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

Segundo passo: Crie um novo projeto com o nome “Primeiro Projeto” na sua pasta de Documentos (ou onde preferir) e com o tipo 3D selecionado. Desmarque inicialmente a opção “Enable Unity Analytics”, pois exploraremos esses serviços somente em outras aulas. Clique em “Create Project”.

Finalmente, após clicar em “Create Project”, observe uma tela, a qual chamamos de “editor” do Unity 3D. Essa é a tela que você utilizará durante quase todo o processo de criação dos seus jogos.

Nas últimas versões do Unity é possível notar que a cena (Scene) inicial para novos projetos começa sem uma câmera padrão e também sem uma luz direcional na cena (em breve veremos mais a respeito). Não precisa se preocupar se isso acontecer, basta escolher a opção no menu: File -> New Scene e pronto! Já estará com uma cena nova no seu projeto 3D com os elementos básicos necessários (Câmera e Luz).

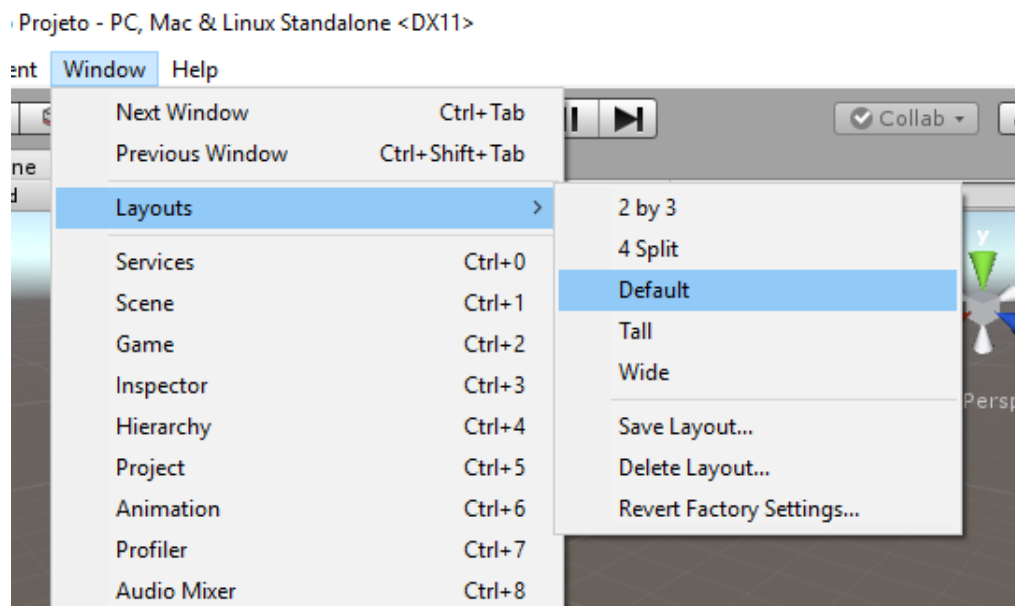
Figura 04 - Primeiro projeto no Unity. Tela inicial após escolher a opção “File -> New Scene”.
Captura de tela do Unity – Game Engine.



Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

Assim como no modo 2D, o Unity tem várias janelas que podem ser posicionadas onde você desejar. Caso deseje voltar para a organização de janelas padrão do Unity, pode a qualquer momento escolher a opção no menu: Window -> Layout -> Defaults, de acordo com a Figura 5.

Figura 05 - Menu com opções de Layouts disponíveis no Unity. Captura de tela do Unity – Game Engine.



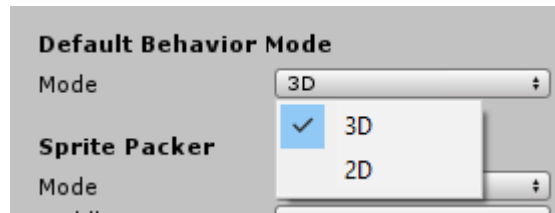
Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

Projetos 2D ou 3D?

Unity suporta a criação tanto de jogos 2D como de jogos 3D. Como visto, ao criar um novo projeto no Unity, é possível fazer essa escolha. Dito isso, é importante saber o que realmente é um jogo 2D ou um jogo 3D. Pode parecer algo simples de se definir mas não é bem assim...

A escolha do projeto 2D ou 3D no momento da criação determina algumas configurações do editor do Unity, entretanto você pode variar entre o modo 2D e 3D a qualquer momento, independentemente da escolha inicial que fez ao criar o projeto. Para optar entre os modos 2D e 3D, inicialmente escolha a opção no menu "Edit > Project Settings > Editor" e na janela "*Inspector*" exibirá diversas configurações do editor. Entre elas está a opção "Default Behavior Mode" e logo abaixo a opção "Mode", que deve estar em 2D ou 3D. Para trocar basta selecionar o modo desejado, conforme a Figura 6.

Figura 06 - Escolha do modo de jogo entre 2D e 3D. Captura de tela do Unity – Game Engine.



Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

As diferenças nas configurações no editor de acordo com cada modo de jogo selecionado (2D ou 3D) são as seguintes:

No modo 2D:

- Qualquer imagem importada no projeto é considerada um *Sprite* e fica configurada como *Sprite Mode*;
- O *Sprite Packer* é habilitado;
- A *Scene View* é configurada por padrão para 2D;
- Os objetos do jogo por padrão não têm luz direcional em tempo real;
- A posição padrão da câmera é 0,0,-10 (No modo 3D é 0,1,-10);
- A câmera é configurada para o modo *Orthographic* (no modo 3D é *Perspective*);
- Na janela *Lighting* (iluminação):
 - O *Skybox* é desabilitado para novas cenas;
 - O parâmetro *Ambient Source* é configurado para Color com a cor configurada para *dark grey*. RGB: 54, 58, 66;
 - *Precomputed Realtime GI* é configurado para off (desligado);
 - *Baked GI* é desativado;
 - *Auto-Building* desativado.

No modo 3D:

- Novas imagens importadas NÃO são consideradas automaticamente como imagens 2D (Sprites);
- O *Sprite Packer* é desativado;
- A janela Scene View é configurada para 3D;
- Os GameObjects padrão têm luz direcional em tempo real;
- A posição padrão da câmera é 0,1,-10 (no modo 2D é 0,0,-10);
- A câmera é configurada para o modo Perspective (em 2D é *Orthographic*);
- Na janela *Lighting*:
 - O Skybox é o Material padrão para *Skybox*;
 - *Ambient Source* é configurado como *Skybox*;
 - *Precomputed Realtime GI* é ativado;
 - *Baked GI* é ativado;
 - *Auto-Building* é ativado.

Sim, eu sei, são várias opções que ainda não vimos. Não se preocupe, exploraremos algumas delas nas próximas aulas. Por enquanto é importante saber que a mudança do modo do jogo depois de criado não afeta somente a câmera e a visualização, mas também várias opções de importação de imagem e de iluminação.

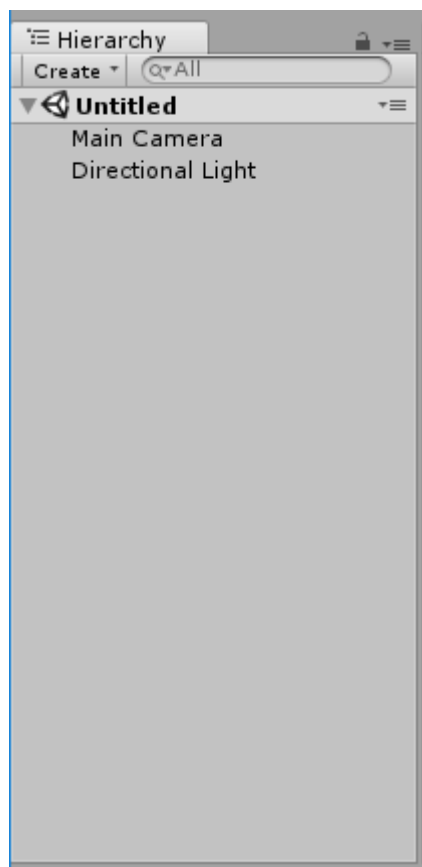
Conhecendo a Interface do Unity

Voltando ao nosso projeto, faremos agora uma rápida revisão sobre a interface do Unity, já que na disciplina de Motores de Jogos I você já teve contato com esse assunto.

A janela principal do Unity é composta por diversos grupos de outras janelas com abas (tabs) que podem ser organizadas como você preferir. A organização padrão (Layout Default) exibe as seguintes janelas:

A janela ***Hierarchy*** lista os nomes dos objetos da cena atual de forma hierárquica. Recentemente o Unity passou a suportar o conceito de edição de múltiplas cenas, o que facilita a passagem de objetos entre cenas sem precisar fechar uma cena e abrir outra. Portanto, conforme pode ser observado na Figura 7, a *Hierarchy* exibe inicialmente a cena atual “Untitled” e logo abaixo dois objetos nessa cena, a câmera principal (Main Camera) e uma luz direcional (Directional Light), já que são dois os objetos padrão em toda nova cena no modo 3D.

Figura 07 - Hierarchy. Captura de tela do Unity – Game Engine.

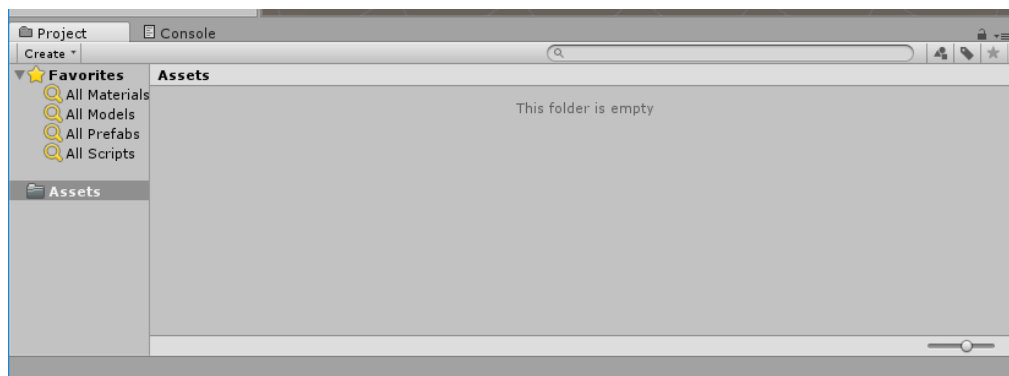


Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

A janela Project mostra os **Assets** que estão no seu projeto em uma hierarquia de pastas, semelhante ao navegador de arquivos tradicional. No lado esquerdo existe uma lista com os Favoritos e outra chamada Assets. Você pode (e deve!) organizar os seus Assets em pastas de acordo com uma estrutura que facilite sua localização futura e todas essas pastas devem ser criadas abaixo da pasta “Assets”

que vem por padrão no seu projeto. Ao clicar em uma pasta o seu conteúdo é exibido do lado esquerdo da janela *Project*. No nosso projeto não temos ainda nenhum asset e, portanto, ela se encontra vazia, observe a Figura 8:

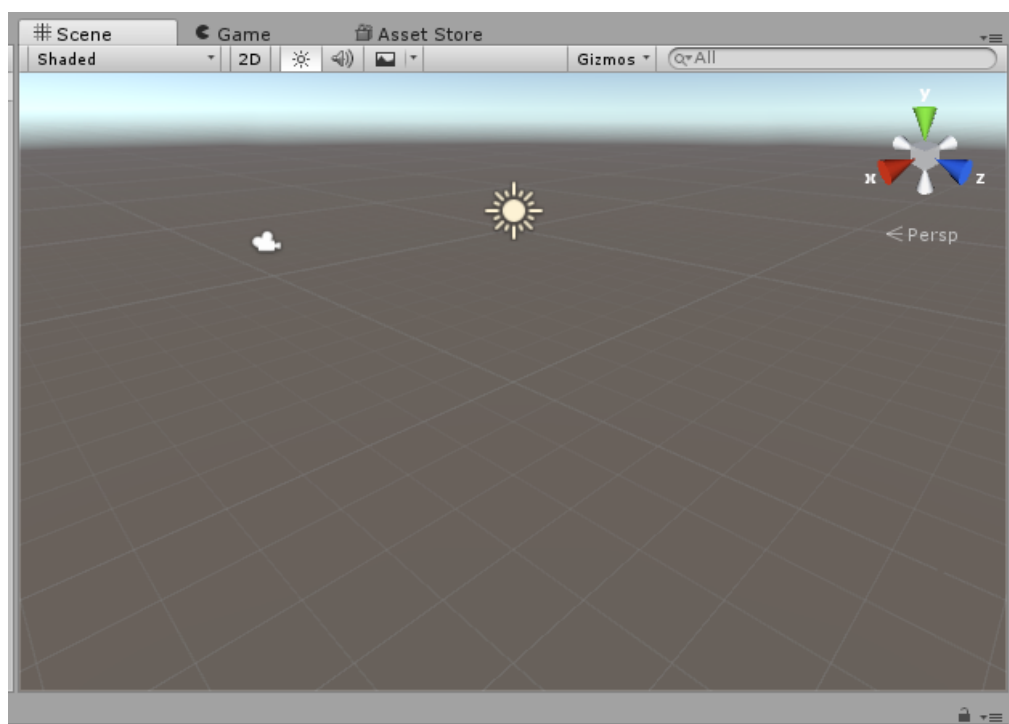
Figura 08 - Janela *Project*. Captura de tela do Unity – Game Engine.



Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

Para adicionar Assets como figuras e arquivos de áudio no projeto, você pode arrastá-los do gerenciador de arquivos diretamente para a janela *Project*, ou utilizar a opção no menu *Assets > Import New Asset*. Veremos mais sobre importação de Assets nas próximas aulas.

Figura 09 - Janela *Scene*. Captura de tela do Unity – Game Engine.

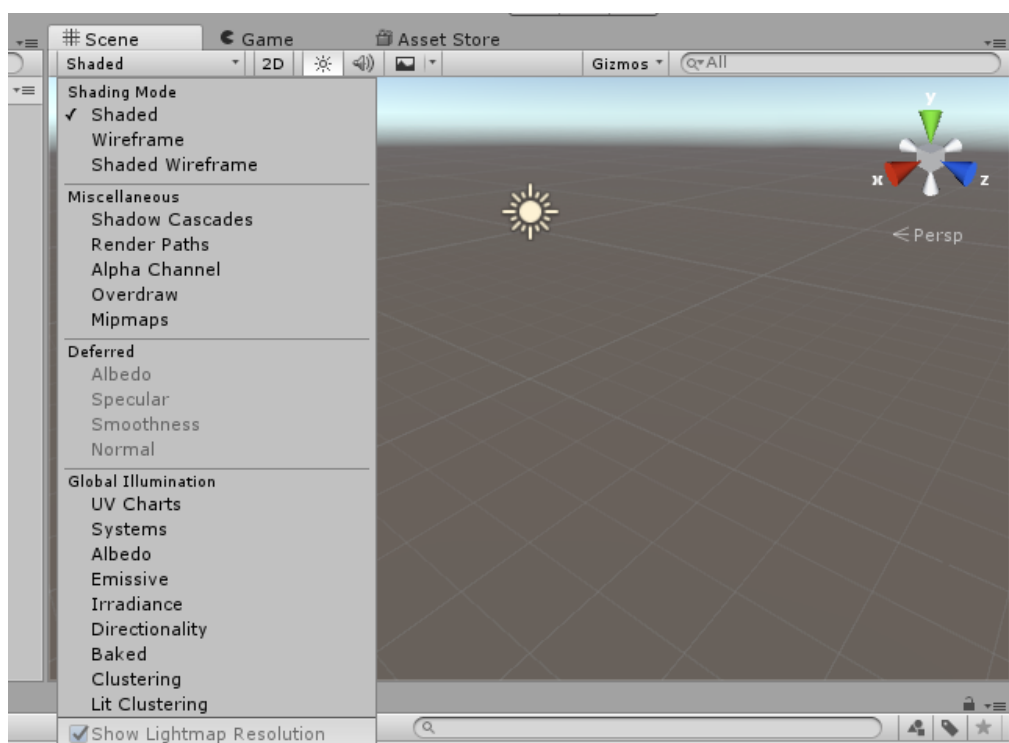


Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

A janela *Scene* exibe visualmente sua cena e permite que você manipule os objetos e navegue na cena em duas ou três dimensões. Também chamado de *Scene View*, ela também mostra ícones dos objetos que não aparecem no jogo como um componente com uma malha 3D. Esses ícones são chamados de **Gizmos** e podem ser vistos na Scene View representando cada elemento a que ele é relacionado. Na Figura 9 é possível ver a cena aberta com um grid 3D em cinza escuro, no canto superior direito um indicador com os três eixos dimensionais (X, Y, e Z) e qual tipo de perspectiva estamos visualizando (veremos mais sobre perspectivas nas próximas aulas), chamado de *Scene Gizmo*, e também 2 *Gizmos*, em que o primeiro representa a posição da câmera e o segundo a posição de uma luz (parecido com um sol).

A **Scene View** tem alguns controles os quais permitem a alternância entre os modos de visualização. São muitos modos, os dois mais utilizados são “Shaded”, que exibe todas as superfícies dos objetos assim como os efeitos de iluminação e sombra, e “Wireframe”, este exibe somente as arestas dos objetos, podendo ser útil quando se deseja visualizar objetos que estão atrás de outros do seu ponto de vista. Veremos sobre modos de visualização mais adiante, aguenta firme aí!

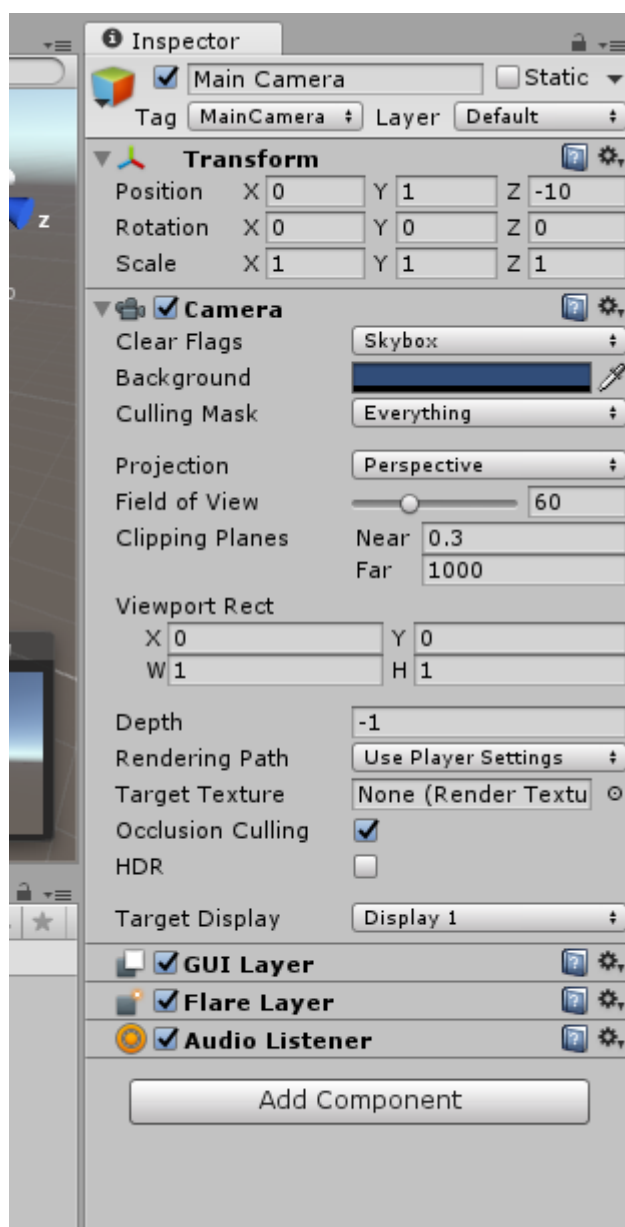
Figura 10 - Modos de visualização. Captura de tela do Unity – Game Engine.



Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

O **Inspector** é a janela onde as propriedades de um objeto selecionado podem ser visualizadas e manipuladas. No *Inspector* também é possível adicionar mais componentes em um determinado objeto (GameObject) para que ele tenha mais funcionalidades e comportamentos. Algumas propriedades, como rotação, posição e escala, podem ser modificadas tanto no *Inspector* como diretamente no *Scene View*; outras podem ser modificadas somente no *Inspector*.

Figura 11 - *Inspector*.. Captura de tela do Unity – Game Engine.

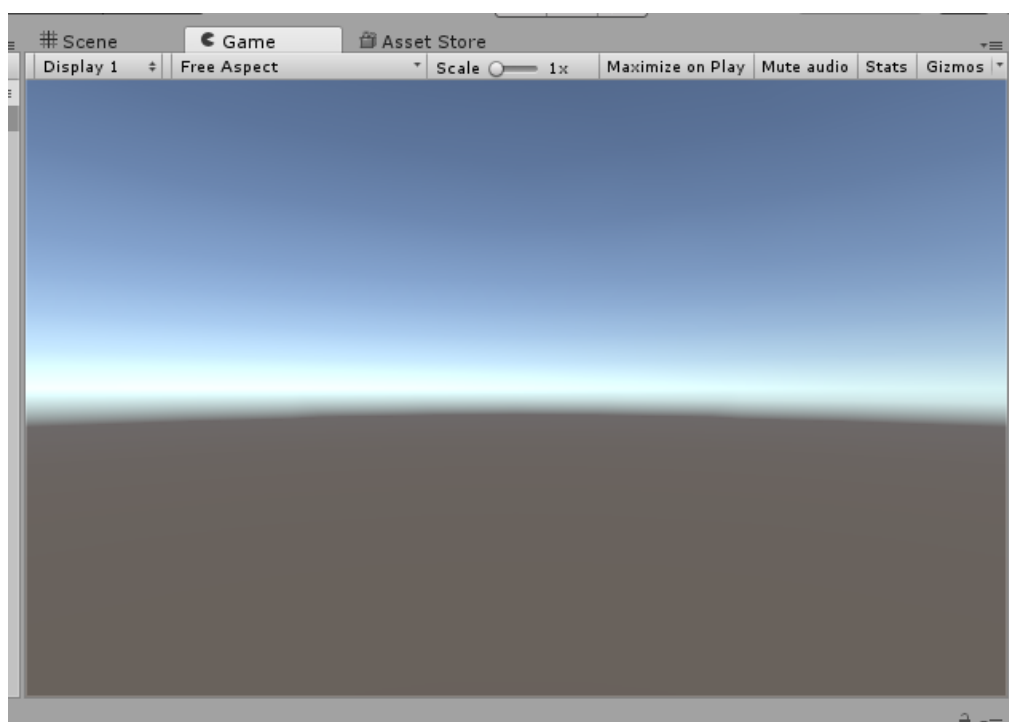


Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

A janela chamada de **Game View** exibe em tempo real a visão da sua câmera principal enquanto está alterando a cena, e também é o local onde você pode testar o seu jogo quando ele for executado em modo (Através do botão “Play”). Na Figura 12 você pode ver que a nossa cena já não exibe os Gizmos vistos na *Scene View*, além disso o Grid do cenário sumiu, ficando somente o céu azul, o qual chamamos de *Sky Box*, ou seja, o *Game View* mostra exatamente o que o jogador irá observar ao executar seu jogo.

Falamos do Sky Box, que é uma grande imagem exibida como plano de fundo do jogo, movida de acordo com a posição da câmera. Nas próximas aulas aprenderemos a modificar o Sky Box para adicionar imagens com diferentes aspectos para que seu jogo fique com o aspecto e o clima desejados.

Figura 12 - *Game View* em uma cena vazia. Captura de tela do Unity – Game Engine.










Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017


O **Toolbar** do Unity sempre contém esses mesmos botões vistos na Figura 13, mantendo-os sempre à vista para acesso rápido.


Figura 13 - Toolbar (Barra de ferramentas). Captura de tela do Unity – Game Engine.



Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

No lado esquerdo da Toolbar temos o botão  de navegação da Scene View,  para mover objetos,  rotacionar objetos,  mudar escala de objetos e  também para mudar a escala somente em um plano. Além disso podemos alternar os modos de manipulação dos objetos selecionados como pivot/centro e alternar entre o modo de rotação de global para local com os botões  , respectivamente. Veremos mais detalhes sobre esses recursos em outras aulas.

Os botões no centro da Toolbar  permitem iniciar/pausar/avançar seu jogo (que será exibido no *Game View*). Pausar a execução do jogo no meio da execução ajuda a analisar como ele está funcionando em um determinado ponto no tempo.

Os botões da direita  permitem que você acesse facilmente os serviços na nuvem do Unity, tenha acesso à sua conta e também possa escolher quais Layers devem ser exibidos (por padrão são todos. Veremos mais sobre Layers nas próximas aulas) e também qual tipo de Layout de janelas você deseja utilizar.



Tudo bem até aqui? Já conhecemos diversas ferramentas do Unity para o desenvolvimento de jogos em 3D, mas ainda temos muito a conhecer!

É importante que você acompanhe as aulas e coloque em prática as orientações para diminuir possíveis dificuldades!

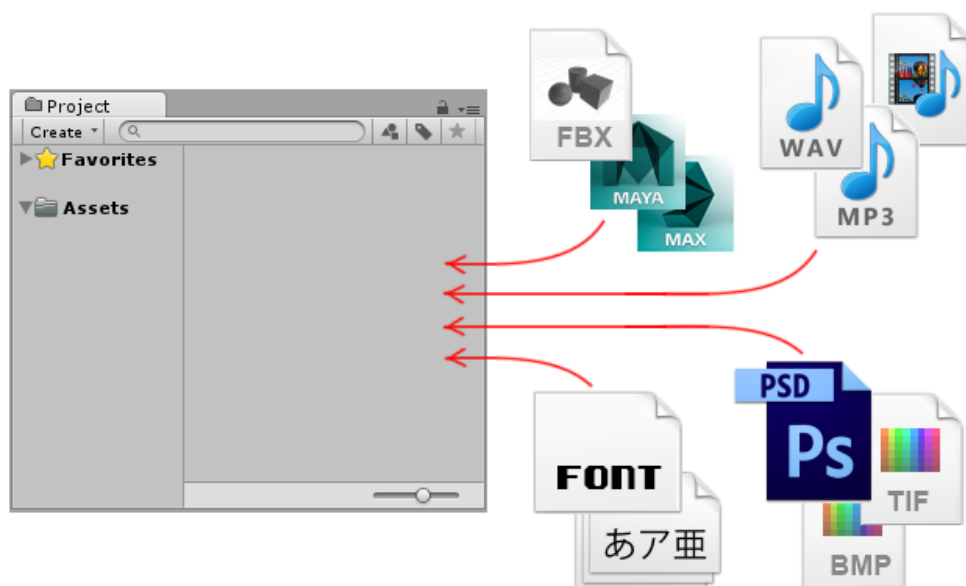
Assets e objetos primitivos

Importação de Assets

O Unity permite que você adicione diversos tipos de objetos em seu jogo, sejam eles imagens, sons, modelos 3D, etc. Esses objetos podem ser adicionados no seu projeto e na sua cena. Quando falamos que um objeto é adicionado no seu projeto, por exemplo um modelo 3D de um navio, isso não significa que ele automaticamente estará em uma de suas cenas e sim que esse modelo foi copiado para a pasta do seu projeto, ficará disponível para uso na janela *Project* e pode então ser adicionado em qualquer cena se você arrastá-lo da *Project* para a *Scene View* ou para o *Hierarchy*.

Você já deve ter visto como adicionar novos objetos no seu projeto na disciplina de Motores de Jogos I. Para projetos 3D não existe nenhuma diferença, basta arrastar o arquivo do objeto para a janela Project e, se o Unity suportar o tipo do arquivo, ele será importado e ficará disponível, então passará a ser um recurso chamado de **Asset**.

Figura 14 - Importação de arquivos para o projeto. Captura de tela do Unity – Game Engine.



Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

Criação de objetos na cena

Antes de trabalhar com Assets criados utilizando outros programas, vamos explorar o conjunto de objetos primitivos disponibilizados pelo Unity. Esses objetos primitivos 3D são uma boa forma de você criar o protótipo de um game sem perder muito tempo modelando seus personagens, paredes, pedras, piso, etc.

É uma boa prática sempre criar um protótipo da ideia do game que você está pensando, para depois investir o seu tempo adicionando os assets finais ao projeto. Essa forma de trabalhar lhe permite ter uma versão funcional do seu jogo em pouco tempo, de modo que possa testar várias ideias antes de trabalhar no acabamento do jogo.

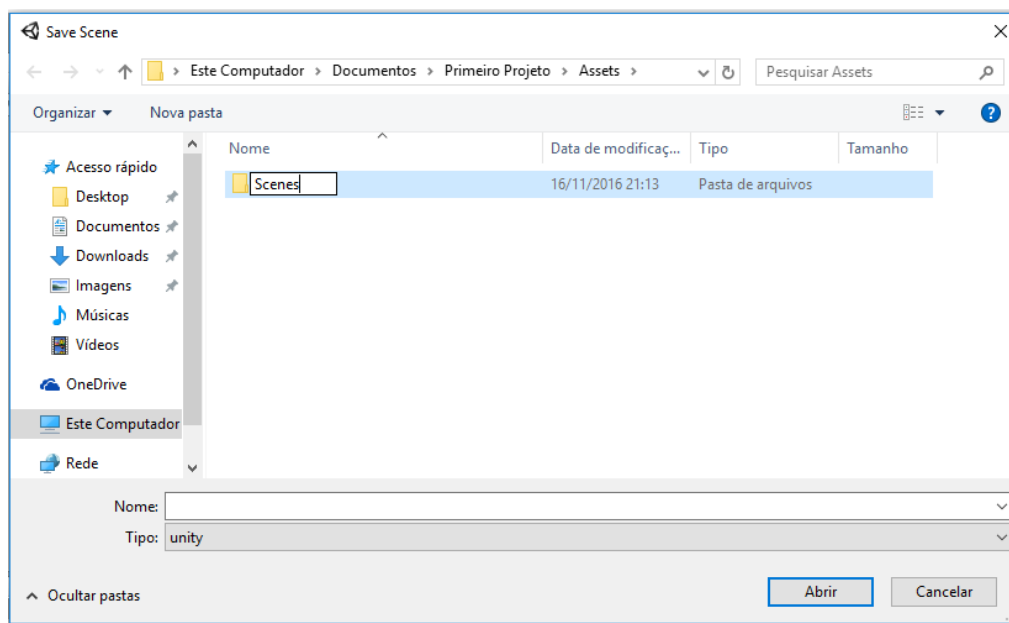
Os objetos primitivos que o Unity disponibiliza em 3D são:

- Cube (Um simples cubo)
 - Sphere (Uma esfera)
 - Capsule (Uma cápsula, similar ao cilindro com as extremidades esféricas)
 - Cylinder (Um cilindro com extremidades achatadas)
 - Plane (Um plano de tamanho 10x10 unidades)
 - Quad (Um plano de tamanho 1x1 unidades). Quads são preferidos aos Planes quando se deseja somente utilizá-lo como se fosse um local para renderizar uma imagem ou um vídeo, já que o mesmo é composto por apenas 2 triângulos e consome menos recursos da máquina. Em todos os outros casos onde você deseja ter uma plane na sua cena utilize a primitiva Plane.
-

Salvando a cena

Antes de adicionar qualquer tipo de objeto primitivo na nossa cena é importante salvá-la. No Unity, uma cena (Scene) é somente um arquivo e precisa ser constantemente salvo na medida em que você a modifica.

Figura 15 - Criando uma pasta para as cenas. Captura de tela do Unity – Game Engine.

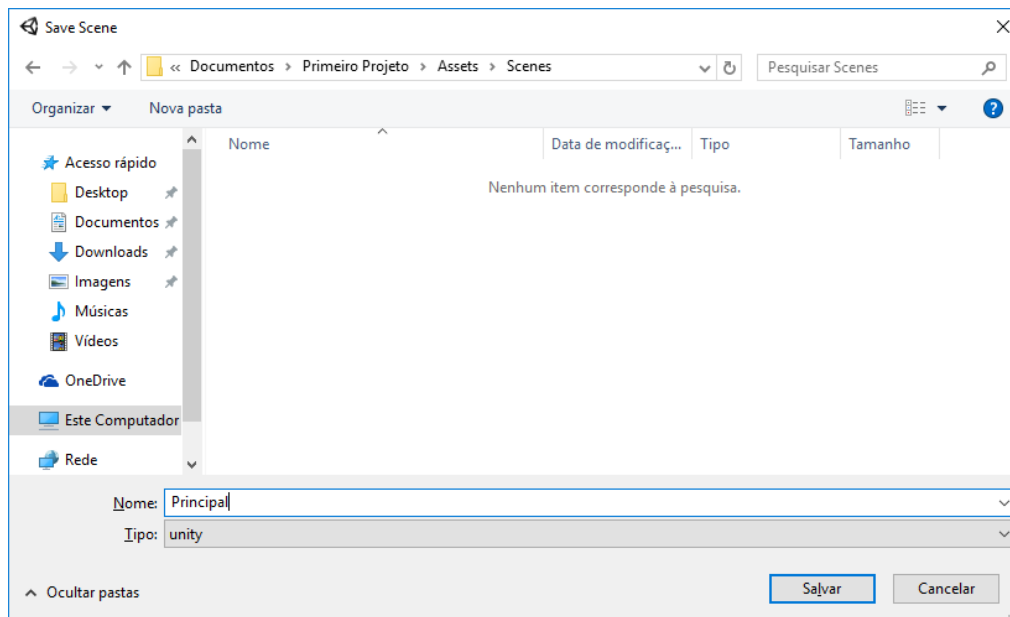


Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

Para salvar a sua cena vazia atual vá no menu “File > Save Scene” ou através da tecla de atalho CTRL+S. Se for a primeira vez que estiver salvando sua cena, aparecerá uma janela de salvamento de arquivo para ser feita a escolha da pasta e do nome da cena. Observe que por padrão você estará dentro da pasta “Assets” do seu projeto. Essa pasta é exibida na janela Project e obrigatoriamente todo Asset (Sim, a cena é um asset também) do seu projeto deve ficar nessa pasta ou em suas subpastas.

Para manter o projeto organizado é recomendável a criação de uma subpasta para cada tipo de Asset. Como estamos salvando uma cena, vamos criar uma subpasta chamada “Scenes”, como na Figura 15, e logo depois faremos um duplo clique nessa pasta para que a cena seja salva dentro dela. No campo “Nome:” digite “Principal” (pode ser outro nome também), já que essa se trata da primeira cena do seu jogo, e clique em Salvar, conforme demonstra a Figura 16.

Figura 16 - Salvando a cena principal dentro da nova subpasta “Scenes”. Captura de tela do Unity – Game Engine.

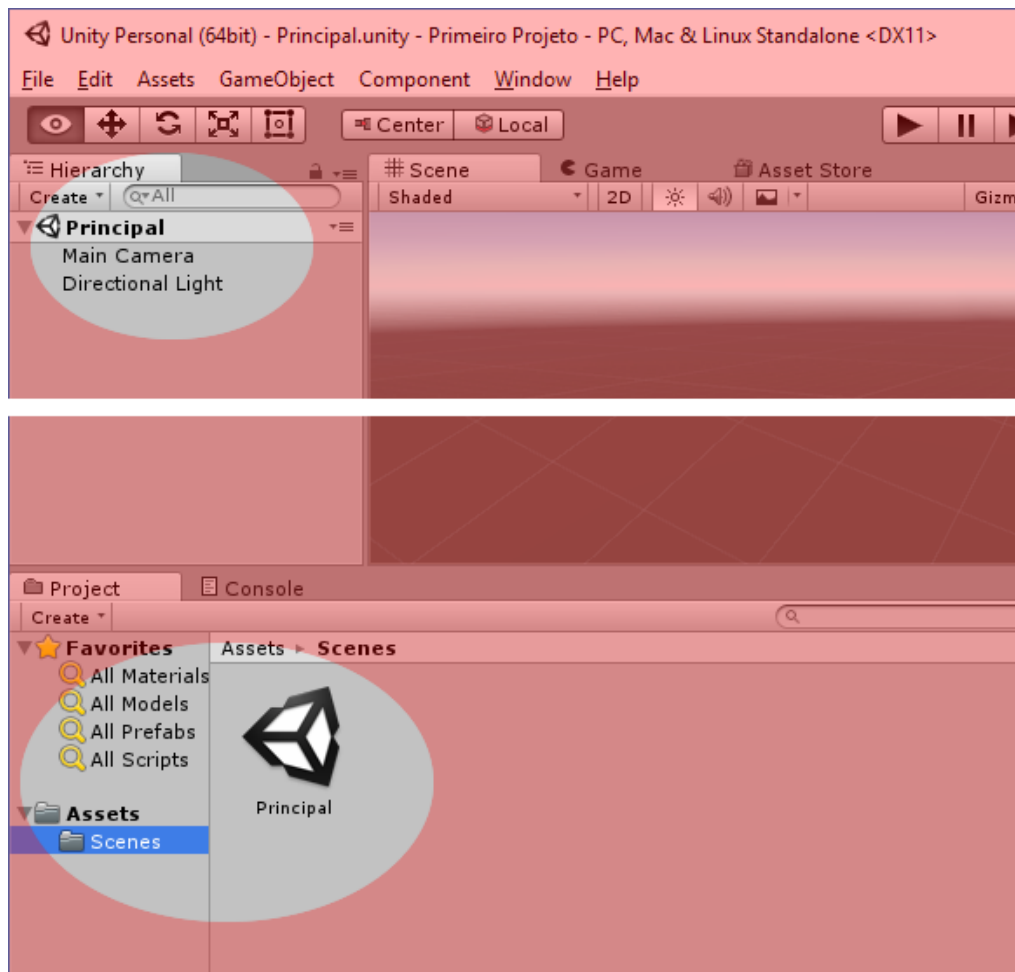


Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

Assim que a cena for salva, percebe-se que na *Hierarchy* a cena antes chamada *Untitled* agora está com o nome “Principal”, na janela *Project* agora existe uma subpasta chamada “Scenes” e dentro dela tem um Asset também chamado “Principal”, que é a sua cena.

Veja na Figura 17 essas mudanças:

Figura 17 - Cena “Principal” salva e sendo exibida no *Hierarchy* e na janela *Project*. Captura de tela do Unity – Game Engine.



Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

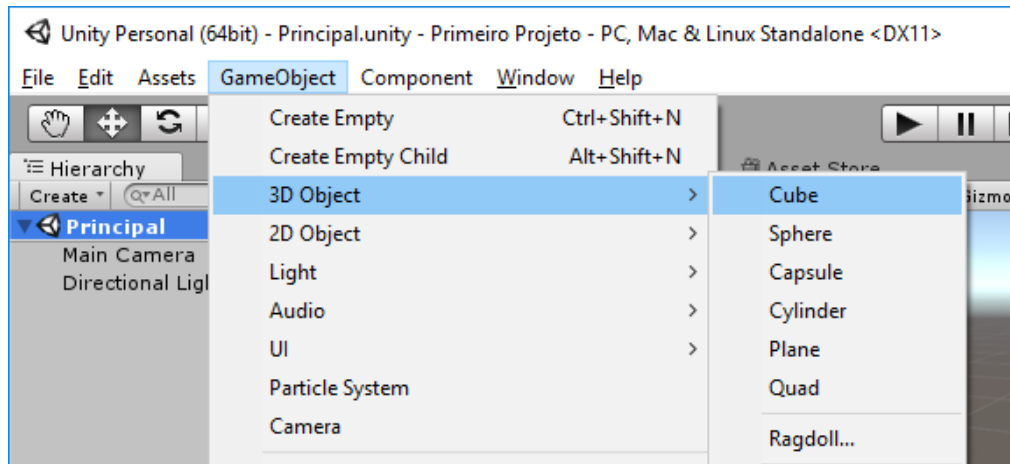
Seu jogo pode conter várias cenas, devendo pensar nelas como algo que pode ser carregado na memória e executado. Você só pode executar no jogo uma cena por vez e veremos nas próximas aulas como alternar entre cenas durante a partida. Para editar uma determinada cena, basta localizá-la na janela Project e dar um duplo clique, no entanto, como já estamos editando a cena principal isso não é necessário agora.

Criando GameObjects na cena

Com nossa cena principal salva, vamos inicialmente criar um simples cubo no centro da cena. Para isso utilizaremos o objeto *Cube*.

Para criar um *Cube*, inicialmente selecione a cena principal na janela *Hierarchy* e depois vá no menu “GameObject > 3D Object > Cube”. Veja na Figura 18:

Figura 18 - Criação de um Cube dentro da cena Principal. Captura de tela do Unity – Game Engine.

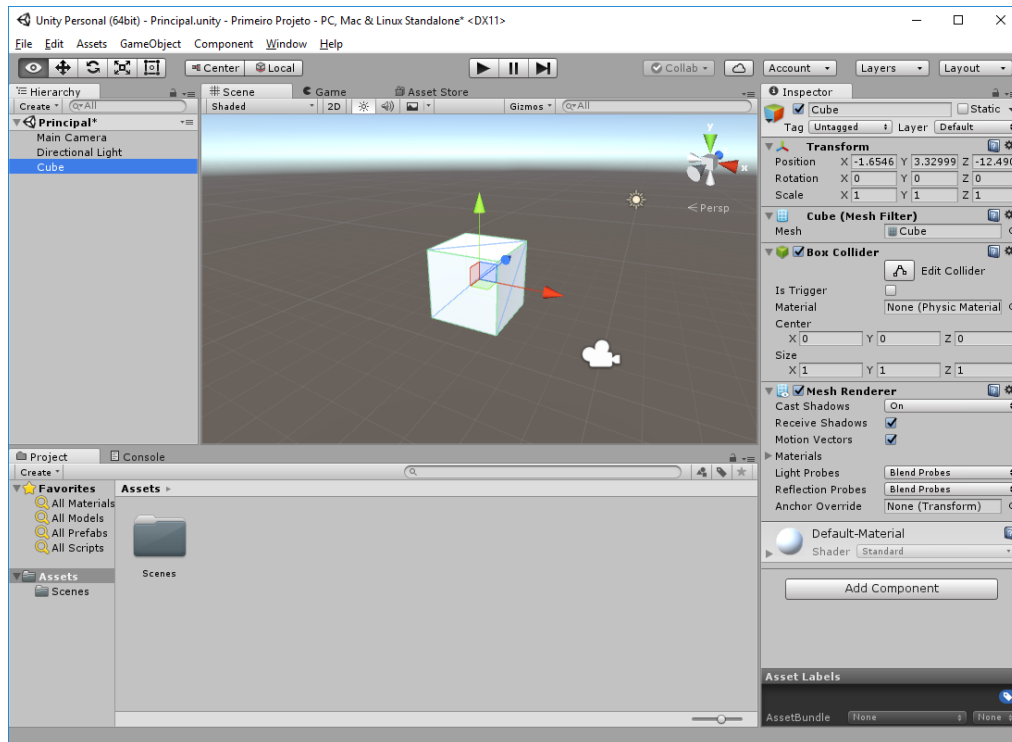


Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

Observe que é importante selecionar a cena “Principal” na janela *Hierarchy*, pois assim o *Cube* será criado imediatamente na raiz da cena. Se você tiver com, por exemplo, a câmera selecionada antes de adicionar o *Cube*, esse *GameObject* será adicionado como “filho” da câmera, mas não queremos isso agora. Se, por acidente, adicionou um *GameObject* como filho de um outro qualquer e deseja mudar isso, basta arrastar esse *GameObject* para cima do nome da cena (*Principal*). Veremos mais sobre hierarquia de objetos nas próximas aulas.

A Figura 19 mostra o *Cube* adicionado no *Hierarchy* e exibido no *Scene View*. Como ele está selecionado no *Hierarchy*, então pode ver seus detalhes no *Inspector*.

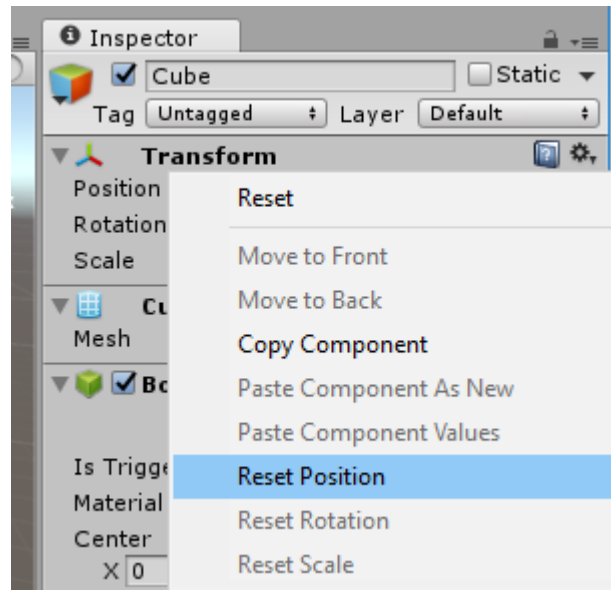
Figura 19 - *Cube* adicionado na cena. Captura de tela do Unity – Game Engine.



Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

Veja que no *Inspector* o Cube não está realmente centralizado na nossa cena. Para isso a sua propriedade *Position* da seção *Transform* deverá ter os valores 0, 0, 0 (eixos X, Y e Z, respectivamente). Para centralizar nosso Cube você pode digitar os valores 0, 0, 0 nos campos indicados ou pode também clicar no ícone representado por uma pequena engrenagem no canto superior direito da seção *Transform*, e no menu que aparecer escolher a opção "Reset Position", como visto na Figura 20. Após essa ação a posição do Cube deverá ficar com o valor 0, 0, 0. O mesmo pode ser feito para Rotação e Escala. Se você escolher a opção com somente o nome "Reset", todas as três serão modificadas deixando rotação e posição com 0, 0, 0 e escala com 1, 1, 1 (o valor padrão).

Figura 20 - Resetar posição de um GameObject. Captura de tela do Unity – Game Engine.



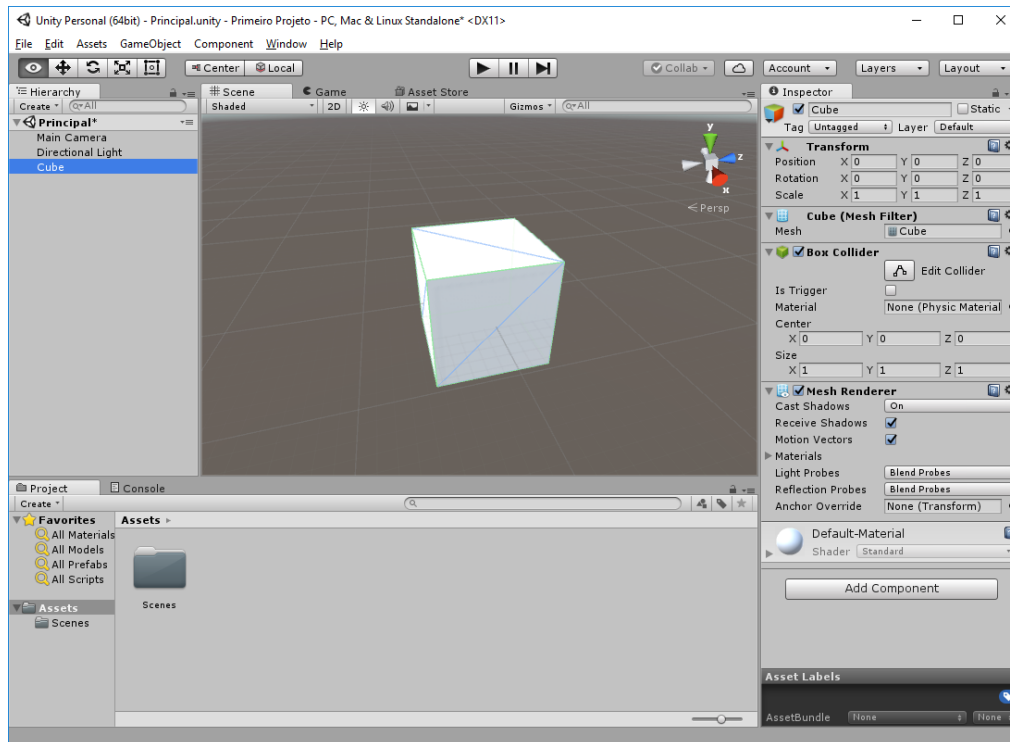
Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

Navegando na cena

Nossa cena agora tem um simples Cube centralizado. Vamos aprender a navegar na cena em 3 dimensões utilizando o Scene View, o teclado e o mouse.

Um dos recursos mais úteis de navegação de cena no Unity é a possibilidade de focar rapidamente a *Scene View* em um *GameObject*. Para isso, basta ir até a janela Hierarchy e dar um duplo clique no *Cube*. Isso fará com que a *Scene View* rapidamente mude seu ponto de vista para deixar o *Cube* no centro e com um zoom que o faça caber por completo na tela, como pode ser visto na Figura 21.

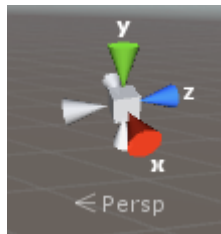
Figura 21 - *Cube* centralizado na *Scene View*. Captura de tela do Unity – Game Engine.



Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

Note que no canto superior direito da *Scene View* existe o que chamamos de *Scene Gizmo* representado pela imagem vista na Figura 22:

Figura 22 - *Scene Gizmo*. Captura de tela do Unity – Game Engine.

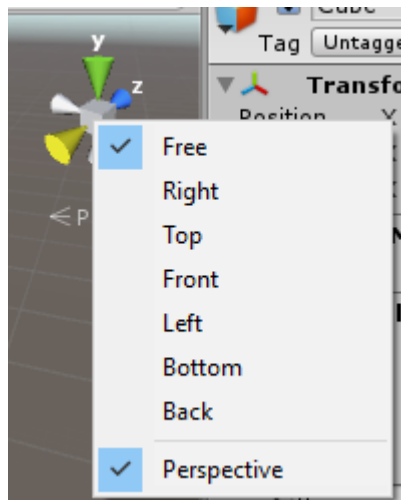


Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

O *Scene Gizmo* mostra a atual orientação do ponto de vista da cena e o tipo de perspectiva que está sendo utilizada. Como visto na Figura 22, você pode alternar entre vários tipos de ponto de vista (Top, Back, Front, etc) e pode alternar entre uma visão perspectiva ou isométrica. Para isso, basta clicar com o botão direito do mouse no *Scene Gizmo* e fazer a escolha no menu. A visão isométrica é muito utilizada no que chamamos de jogos 2D e também em 2.5D, que são jogos construídos em 3D,

mas com a jogabilidade que lembra jogos 2D. Exploraremos mais esse assunto de perspectiva/isométrico em outras aulas, por enquanto mantenha selecionado “Perspective”.

Figura 23 - Tipos de ponto de vista e perspectiva. Captura de tela do Unity – Game Engine.

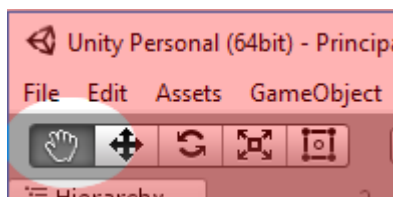


Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

Para uma visão livre, escolha a opção “Free”, assim você poderá manipular o ponto de vista da forma que desejar. Existem diversas formas de manipular o ponto de vista de cena no Scene View. Vamos inicialmente aprender a opção “Move”.

Para utilizar essa opção, habilite a opção “Move” bastando clicar no primeiro botão da Toolbar do Unity com um ícone no formato de uma mão, observe a Figura 24:

Figura 24 - Opção Move. Captura de tela do Unity – Game Engine.

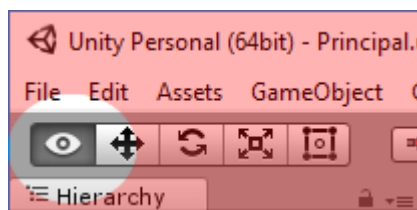


Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

Com a opção Move ativada você pode manter o botão esquerdo pressionado em qualquer local da *Scene View* e arrastar o mouse para navegar na cena para os lados, para cima e para baixo, sem girar a cena.

Outra opção disponível é a *Orbit*, demonstrada na Figura 25. Para ativar essa opção, clique no ícone da Mão, assim como o Move, porém mantenha a tecla ALT pressionada antes de clicar e arrastar o mouse na Scene View. Isso fará com que a câmera orbite a cena modificando o seu ponto de vista.

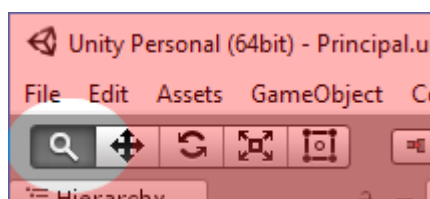
Figura 25 - Opção *Orbit*. Captura de tela do Unity – Game Engine.



Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

A última opção de navegação é a de *Zoom*. Para ativar o Zoom proceda da mesma forma que o Orbit (com a tecla ALT pressionada), porém utilize o botão direito do mouse para clicar e arrastar na *Scene View* e conseguirá aumentar/diminuir o zoom na cena. Quando a opção Zoom está em uso o ícone na Toolbar fica como uma lupa, representada na Figura 26.

Figura 26 - Opção *Zoom*. Captura de tela do Unity – Game Engine.



Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

Atenção, não confunda a câmera da cena com a câmera principal criada no Hierarchy. A câmera da cena não existe no seu projeto, ou seja, ela somente é a câmera virtual que a Scene View está utilizando para exibir a sua imagem no editor do Unity. O que você vê no scene view não é exatamente o que o jogador verá. O jogador vê o que é captado pela câmera principal (Main Camera). Vamos modificar a Main Camera mais adiante, não se preocupe com ela agora!

Basicamente com essas três opções (Move, Orbit e Zoom) você navega na cena para onde desejar. Lembre-se sempre que pode clicar duas vezes em qualquer GameObject no Hierarchy para centralizar a cena nele, facilitando bastante a sua

navegação.

Existe uma quarta opção de navegação chamada de “Flythrough mode”, algo como “modo de voo”, em português. Ela se assemelha ao controle de movimento com teclado e mouse de personagens em jogos em primeira pessoa. Para ativar esse modo é muito simples:

1. Clique e segure o botão direito do mouse em qualquer local na Scene View;
2. "Voe" na cena utilizando as teclas WASD quando quiser ir para esquerda/direita/frente/trás.
3. Utilize as teclas Q e E quando desejar ir para cima e/ou para baixo.

O modo Flythrough pode ser um pouco confuso para quem não está acostumado, porém com a prática é certamente o modo que lhe trará mais produtividade ao se trabalhar no Unity, pois na maioria dos casos permite que você realize a navegação na cena de maneira mais rápida.

Vale lembrar que para todos os modos você pode sempre manter pressionada a tecla SHIFT durante seu uso, fazendo com que a velocidade do movimento seja aumentada. Isso é muito útil quando é necessário percorrer maiores distâncias na cena.

Transform (Posição, Rotação e Escala)

No Unity cada GameObject contém um componente fixo chamado ***Transform***. Esse componente possui três propriedades:

- **Position:** Determina a posição do GameObject na cena nos eixos X, Y e Z com valores que indicam o quanto o centro do GameObject está deslocado da origem da cena em cada eixo.
- **Rotation:** Determina a rotação (em graus) do GameObject em cada eixo.

- **Scale:** Determina a escala do *GameObject* em cada eixo. Cada *GameObject* pode estar relacionado a um modelo 3D que chamamos de “Mesh” (ou malha). Esse modelo pode ser criado por um outro programa e naturalmente tem suas dimensões já determinadas. A propriedade *Scale* modifica essas dimensões antes de exibir o objeto, o “esticando” em qualquer um dos três eixos. O valor “1” determina que naquele eixo a escala do objeto é a original (100%). Se desejar escalonar o objeto em um eixo para que fique com 1,5 vezes o seu tamanho, então você deve adicionar o valor 1.5 no eixo correspondente na propriedade *Scale*. O valor padrão é 1, 1, 1 que exibe o Mesh relacionado ao *GameObject* na escala original sempre.

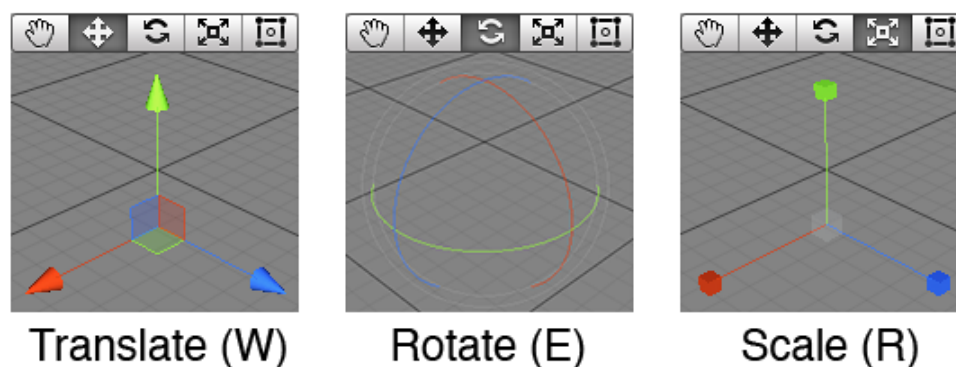
Quando você está posicionando um *GameObject* no Unity é importante ter em mente que cada unidade representa um metro, ou seja: 1 = 1 metro, 15 = 15 metros, 1000 = 1km, 0,1 = 10 cm, 0,01 = 1cm e assim por diante. Isso é importante pois as interações físicas como colisões e gravidade utilizam essa métrica para realizar seus cálculos. Então se deseja posicionar um objeto a 3 metros da origem no eixo X, 5 metros no eixo Y e 1 metro no eixo Z, o valor da propriedade *Position* do *Transform* deve ser: 3.0, 5.0, 1.0.

Translate, Rotate, and Scale

Como vimos, para modificar o *Transform* de um *GameObject* pode-se simplesmente alterar seus valores no *Inspector*. Porém, também é possível realizar essas mudanças utilizando as ferramentas de transformação, acessíveis por meio dos seus ícones na *Toolbar*. Isso permite que você manipule o *GameObject* diretamente na *Scene View* utilizando o Mouse.

São três as principais ferramentas de modificação do *Transform* pela *Toolbar*: o *Translate*, *Rotate* e *Scale*, cada uma modificando a posição, rotação e escala do *GameObject*, respectivamente. Para usá-las, basta clicar no seu ícone ou pressionar a tecla de atalho correspondente.

Figura 27 - Ferramentas de modificação do Transform com suas teclas de atalho e Gizmos..
Captura de tela do Unity – Game Engine.



Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

Quando ativadas, as ferramentas de modificação de *Transform* operam no *GameObject* que se encontra atualmente selecionado no Hierarchy e você observará que um Gizmo será exibido na Scene View sobre esse *GameObject*, indicando que você pode movê-lo, rotacioná-lo ou escaloná-lo, de acordo com a ferramenta selecionada. Também observe na Figura 27 a aparência dos Gizmos para cada operação.

Esses Gizmos são manipuláveis utilizando o mouse, ou seja, você pode interagir com eles clicando e arrastando partes de cada um. Todos eles permitem manipulações com o Mouse em cada eixo representado pela sua cor: Vermelho = X, Verde = Y, Azul = Z.

No Translate (W) você pode mover o *GameObject* em cada eixo de acordo com a cor simplesmente clicando na sua seta e arrastando para a posição desejada. Desse modo o *GameObject* se moverá somente nesse eixo escolhido. Note que no *Gizmo* do *Translate* existe um conjunto com três planos (um verde, um azul e um vermelho), próximo ponto de encontro das três setas. Esses planos também podem ser clicados e arrastados para mover o *GameObject*, porém nesse caso ele será movido em duas dimensões e não somente em um dos eixos. O plano verde move o *GameObject* nos eixos X e Z, o plano azul nos eixos X e Y, o plano vermelho nos eixos Y e Z. Pode parecer um pouco confuso, mas com prática você verá que, na verdade, trata-se de uma maneira muito eficiente de mover *GameObjects* na cena.

No Rotate (E) a mudança da rotação do *GameObject* é muito simples também. Você pode clicar e arrastar cada um dos três “aros” coloridos para rotacionar o *GameObject* ao redor do eixo que sua cor representa.

No Scale (R) poderá clicar e arrastar os pequenos cubos coloridos presentes nas pontas de cada uma das três barras (uma para cada eixo). Ao arrastar um deles você estará diminuindo/aumentando o seu tamanho naquele eixo. Simples assim! Note que existe um cubo cinza claro no centro do Gizmo (no ponto de encontro das três barras coloridas), podendo clicar e arrastar esse cubo cinza para ampliar/reduzir a escala do objeto nos três eixos ao mesmo tempo. Essa última forma de escalonamento (os três eixos ao mesmo tempo) é muito útil, pois modifica o tamanho do GameObject, mantendo suas proporções, como visto no Vídeo 1.

Note que ao clicar nas partes manipuláveis desses Gizmos, elas ficam temporariamente com a cor amarela, para deixar claro em qual delas vocês está clicando.



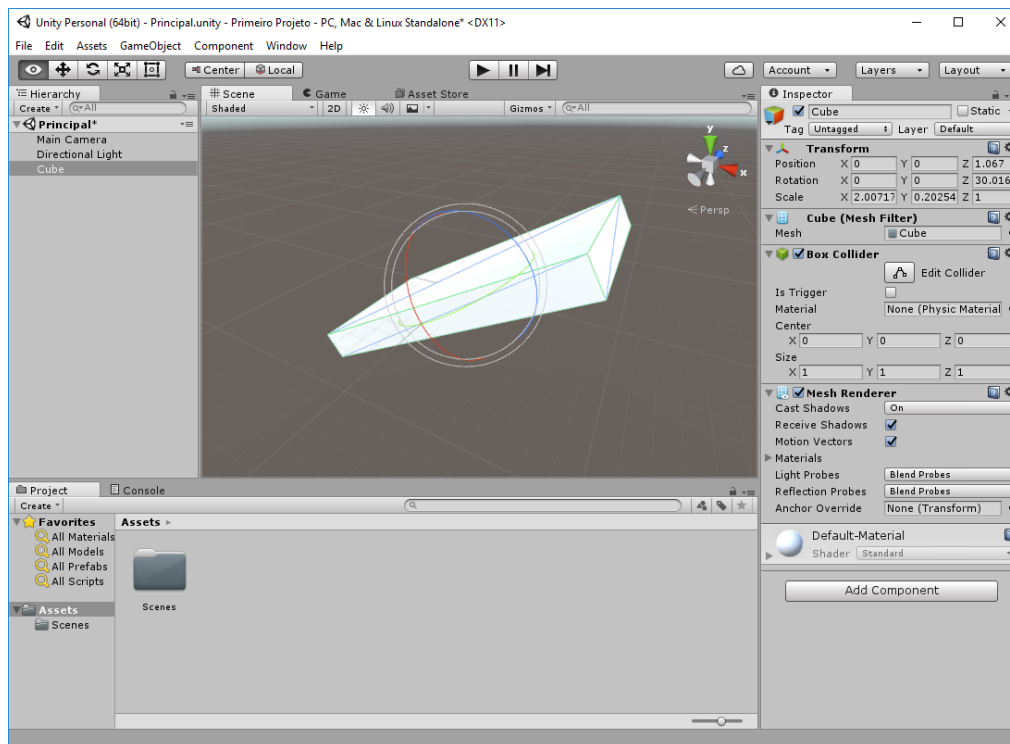
Video 01

Vamos praticar! Sua cena atualmente contém somente um Main Camera, uma Direcional Light e um Cube no posicionado na origem. Utilizando as ferramentas de modificação de Transform do Toolbar, faça as seguintes mudanças no Cube utilizando somente o Mouse:

1. Mova o Cube no eixo Z e deixe-o o mais próximo que conseguir a 1 metro (uma unidade no Unity) da origem nesse eixo;
2. Modifique a escala do Cube no eixo X para 2.0 e em Y para 0.2 (ou o mais próximo disso que conseguir usando o Mouse);
3. Rotacione o Cube no eixo Z em 30 graus (ou o mais próximo disso que conseguir usando o Mouse).

Se você fez as mudanças sugeridas na atividade, deve ter ficado com um Cube que não é mais um cubo perfeito, e sim um objeto, representado abaixo na Figura 28.

Figura 28 - Cube com mudanças na sua posição, rotação e escala. Captura de tela do Unity – Game Engine.

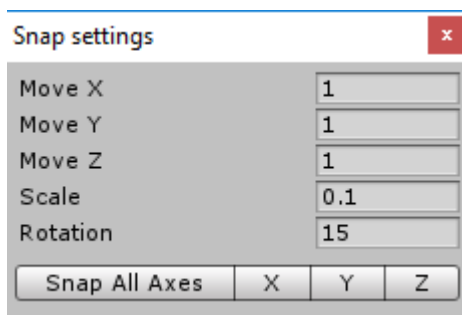


Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

Você deve ter notado que é muito difícil mover/rotacionar/escalar GameObjects com o Mouse para que eles fiquem com valores exatos. Esse é o lado negativo de se usar o mouse para essas mudanças. Para contornar isso, use um recurso chamado *Snap*. O *Snap* faz com que as mudanças no *Transform* de um *GameObject* seja realizada em “passos” de acordo com a configuração do Unity. Para ativar o Snap basta pressionar e segurar a tecla CTRL enquanto realiza a operação. Use a opção Reset do *Transform* para voltar o Cube às suas características originais (Clique na engrenagem próximo ao componente *Transform* e depois em *Reset*) e tente novamente realizar as mudanças descritas anteriormente com o *Snap* ativado. Às vezes o Snap não é perfeito, principalmente para rotação. Esse é um pequeno problema em algumas versões do Unity, mas mesmo assim você pode continuar a usar essa ferramenta para agilizar o seu trabalho.

O Snap pode ser configurado no menu “Edit > Snap Settings...” e na janela exibida você pode configurar o tamanho do passo de movimentação em cada eixo, da escala e da rotação, de acordo com o que desejar, como mostra a Figura 29.

Figura 29 - Snap Settings. Captura de tela do Unity – Game Engine.



Fonte: Disponível em: <https://unity3d.com/pt/>. Acesso em: 20 fev de 2017

Você deve estar se perguntando: -E isso funciona somente para cubos? A resposta é: -Não! As alterações no Transform de um GameObject, tanto no *Inspector* quanto com as ferramentas que vimos, serve para qualquer tipo de objeto 3D, seja com as primitivas Cube, Cylinder, Plane, entre outras, ou modelos 3D importados no Unity.

Que tal “testar” as novas possibilidades e compartilhar suas descobertas com os colegas nos fóruns? É uma oportunidade de também conhecer essas outras ferramentas e ficar cada vez mais hábil no desenvolvimento de jogos com o Unity! ;) Estamos chegando ao fim da nossa primeira aula! Cansados? Que nada, estamos apenas começando! Nas próximas aulas teremos muito mais elementos para conhecer, até lá!

Resumo

Nesta aula você aprendeu a criar projetos 3D no Unity, criar e salvar cenas, adicionar objetos primitivos na cena e modificar suas propriedades utilizando o Transform, com valores exatos e com o Mouse. Nas próximas aulas aprenderemos a criar cenas mais elaboradas e ainda criaremos o nosso primeiro jogo!

Leitura complementar

Como em todo aprendizado de uma ferramenta, avançar nos conhecimentos desta depende muito dos estudos auxiliares que você fará. Abaixo segue uma lista de alguns recursos que você poderá usar para aprender mais sobre o Unity e o desenvolvimento de jogos em geral.

Aprenda com o Unity (oficial): <https://unity3d.com/pt/learn>

Unity 5 Tutorial For Beginners - How To Make A Game:
<https://www.youtube.com/watch?v=G9BdFZ2MCXc&list=PLZ1b66Z1KFKik2g8D4wrmYj4yein4rCk8>

Unity Transform (position, rotation, scale): <https://www.youtube.com/watch?v=EGmXFYSkooo>

Autoavaliação

- Adicione um componente chamado “Physics > Rigidbody” em algumas primitivas na sua cena, pressione o botão “Play” e observe o que ocorre.
- Altere algumas propriedades da Main Camera como Projection, Field of View, Clipping Planes, entre outras, e veja quais mudanças ocorrem ao iniciar o jogo (Play).

Referências

UNITY TECHNOLOGIES. 2016 (C). Unity 3D Online Tutorials [online]. Disponível em: <https://unity3d.com/pt/learn/tutorials> [Acessado em 16 de novembro de 2016].

PASSOS, E. B., SILVA, J., NASCIMENTO, G. T., KOZOVITS, L. CLUA, E. W. G. 2008. Fast and safe prototyping of game objects with dependency injection. Anais do Simpósio Brasileiro de Games e Entretenimento Digital. São Leopoldo, RS. 2008

STOY, C. 2006. Game object component system. In Game Programming Gems 6, Charles River Media, M. Dickheiser, Ed., Páginas 393 a 403.

MARQUES, Paulo; PEDROSO, Hernâni - C# 2.0 . Lisboa: FCA, 2005. ISBN 978-972-722 208-8

UNITY TECHNOLOGIES. 2016 (C). Unity 3D Manual [online]. Disponível em: <https://docs.unity3d.com/Manual/index.html> [Acessado em 16 de novembro de 2016].