

Desenvolvimento de Jogos

UNITY

Profa. Thaiana Pereira dos Anjos Reis, Dra. Eng.
thaiana.anjos@ifsc.edu.br

Prof. Roberval Silva Bett, Me. Eng.
roberval.bett@ifsc.edu.br

Agenda

- Funcionalidades da Unity.

Facilita o desenvolvimento de jogos visto que o desenvolvedor não precisa se preocupar se fará a programação para **OpenGL** ou **DirectX**

1) Game Engine:

É multiplataforma (Mac, iPhone, Xbox, Windows, Android, PS3, Wii, Firefox, Edge, Chrome, Safari...)





2) Interface:

- O Unity possui uma interface simples e amigável;
- Composto de janelas chamadas *views*, definidas a partir de seus propósitos específicos.



TR ☁ 🎯

▶ ⏸ ▶

🔄 🔍 Layers Layout

Hierarchy

Scene

Game

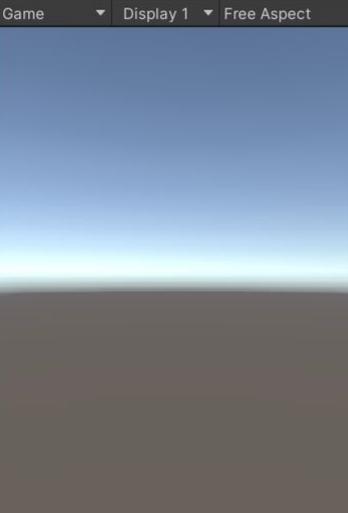
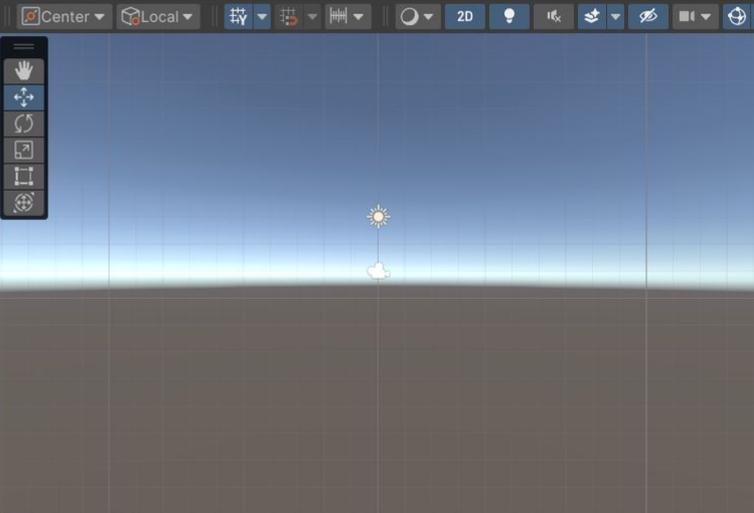
Inspector

+

☰ All

☰ Untitled

- ☰ Main Camera
- ☰ Directional Light



Inspector

Inspector panel showing properties for the selected object.

Project Console

+ 🔍

🔍 📄 🖱️ ⚠️ ⭐ 🗑️ 21

★ Favorites

- 🔍 All Materials
- 🔍 All Models
- 🔍 All Prefabs

📁 Assets

- 📁 Scenes

📁 Packages

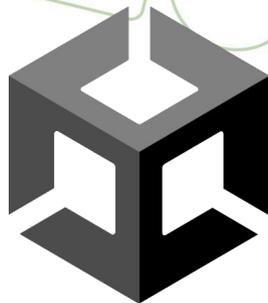
- ▶ 2D Animation
- ▶ 2D Aseprite Importer
- ▶ 2D Common
- ▶ 2D Pixel Perfect
- ▶ 2D PSD Importer
- ▶ 2D Sprite
- ▶ 2D SpriteShape
- ▶ 2D Tilemap Editor

Assets

📁 Scenes

📄 script

Assets panel showing a folder named 'Scenes' and a file named 'script'.



Unity

3) Scene View:

- Ela é utilizada para posicionar e selecionar todos os *Game Objects* (jogador, ambiente, câmeras,...);

Para melhorar a nossa interação com a Scene View vamos analisar alguns atalhos e funcionalidades nos próximos slides.





3) Scene View:

- Segurando o **botão direito do mouse**, entra-se no modo *flythrough* (aéreo), permitindo girar o cenário.
- Ao selecionar qualquer *GameObject* e pressionar a **tecla F**, centralizará o ponto pivô (encontro dos eixos x, y, z) no centro do objeto na *Scene View*.
- Segurando **Alt e clicando com o botão esquerdo do mouse** é possível girar a câmera em torno do pivô do objeto corrente.
- Segurando **Alt e clicando com o botão do meio do mouse** é possível arrastar um *GameObject* pela *Scene View*.
- Segurando **Alt e clicando com o botão direito do mouse** é possível dar zoom na *Scene View*. A mesma ação pode ser feito utilizando o scroll do mouse.



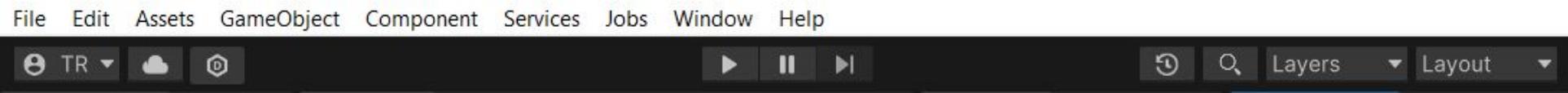
3) Scene View:

- No **canto superior direito** da *Scene View* há o *Scene Gizmo*. Este mostra a atual orientação da câmera da *Scene*, e permite modificar rapidamente o ângulo de visão.



4) Tool Bar:

- Na **parte superior** da interface há um conjunto de ferramentas que nos permite operar a interface, o jogo e seus elementos;



1

2

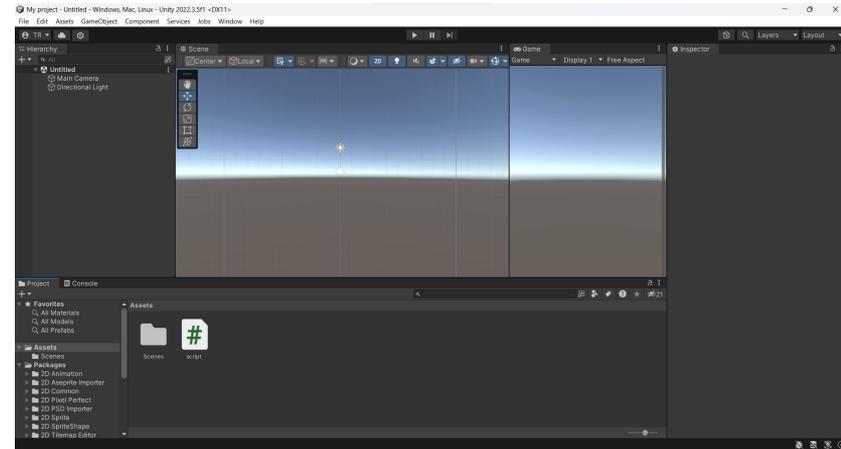
3

4

1. Ferramentas de Usuário e Serviços: conta do usuário, serviços e versão da Unity.
2. Botões *Play/Pause/Step*: utilizados com a *Game View* (Visão do jogo).
3. Caixa suspensa de Camadas: controla quais objetos são mostrados na *Scene View*.
4. Caixa suspensa de Layout: Controles dispostos em todas as *Views*.

5) Warning:

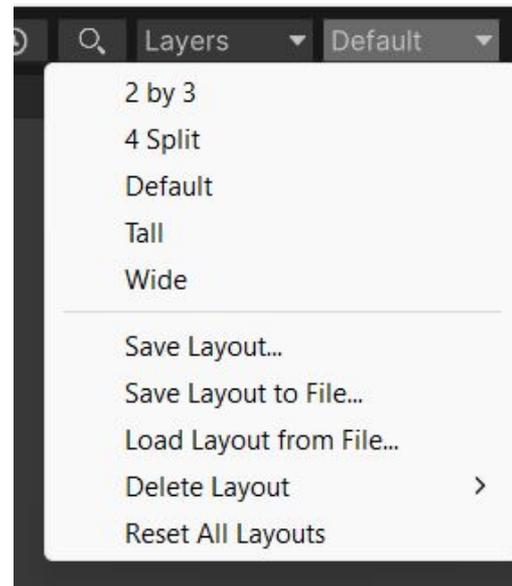
- É muito comum que durante o processo de desenvolvimento de uma aplicação o DEV alterne entre o modo projeto e o modo jogo;
- Deve-se tomar cuidado pois os valores alterados dos atributos dos *Game Objects* durante execução não se mantêm quando retornamos para o modo projeto;
- As alterações feitas em atributos no modo projeto tem efeito também sobre o modo de execução.





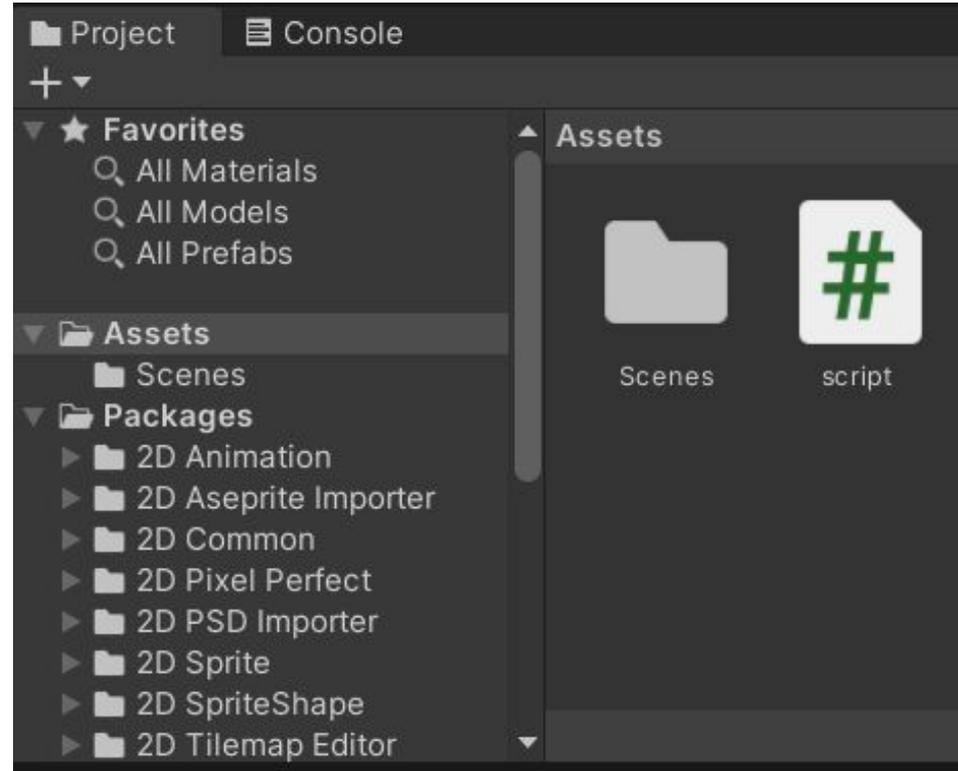
6) Layout:

- No **canto superior direito** é possível encontrar a janela suspensa *Layout*. Nela é possível selecionar outros layouts pré-definidos ou personalizar como você preferir.



7) Project View:

- Todo projeto na Unity possui uma pasta *Assets* (Ativos);
- Ao clicar com o **botão direito do mouse** sobre qualquer *assets* é possível selecionar *“Show In Explorer”*, e visualizar o *asset* no sistema de arquivos do computador.





8) Assets:

- Os *assets* representam todos os ativos para a criação de um jogo, ou seja, as texturas, sons, scripts...;
- Para adicionar *assets* ao projeto basta arrastar qualquer arquivo do computador para dentro do projeto.



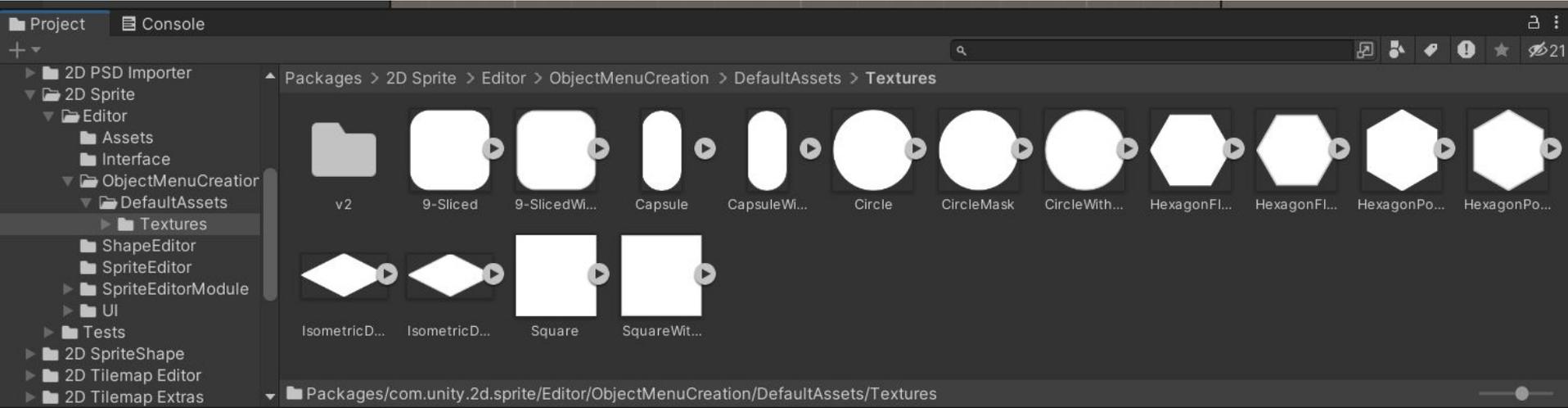
Asset Store

<https://assetstore.unity.com/>



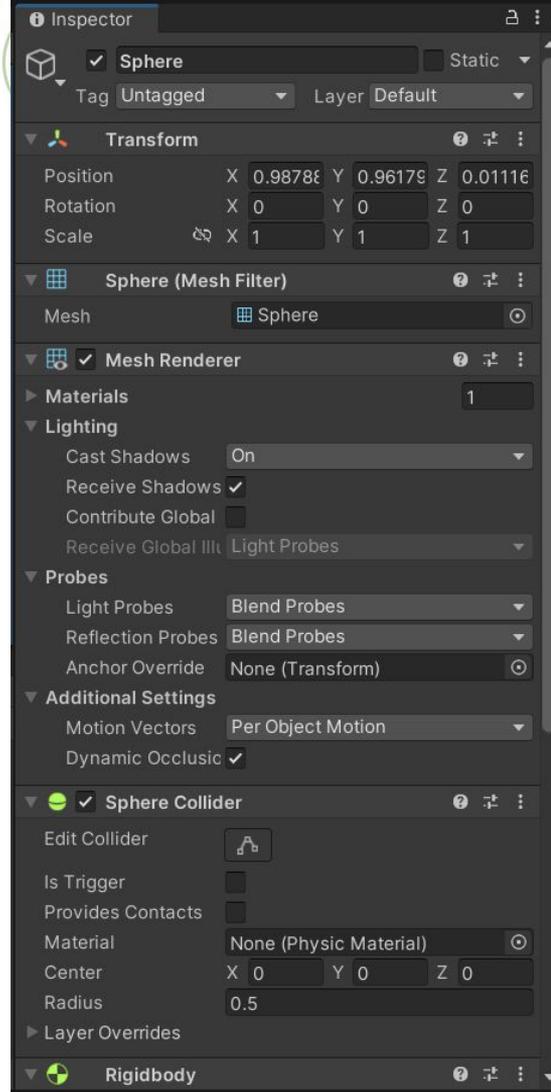
9) Sprites:

- *Sprites* são objetos 2D da Unity.
- Geralmente são texturas ou imagens aplicáveis a *game objects*.



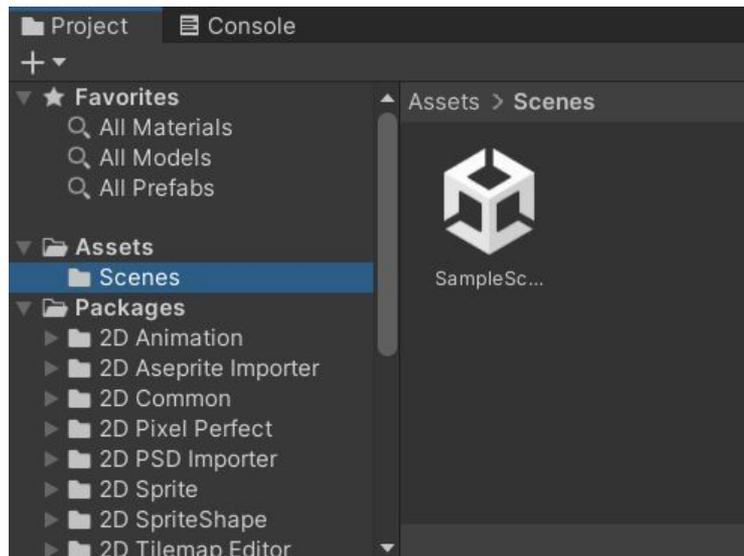
10) Inspector:

- Os jogos são compostos de vários *game objects*;
- Os *game objects* possuem, material, sons, elementos gráficos entre outros;
- O *Inspector* permite mostrar informações detalhadas sobre os *game objects*, desde componentes associados a ele até suas propriedades.



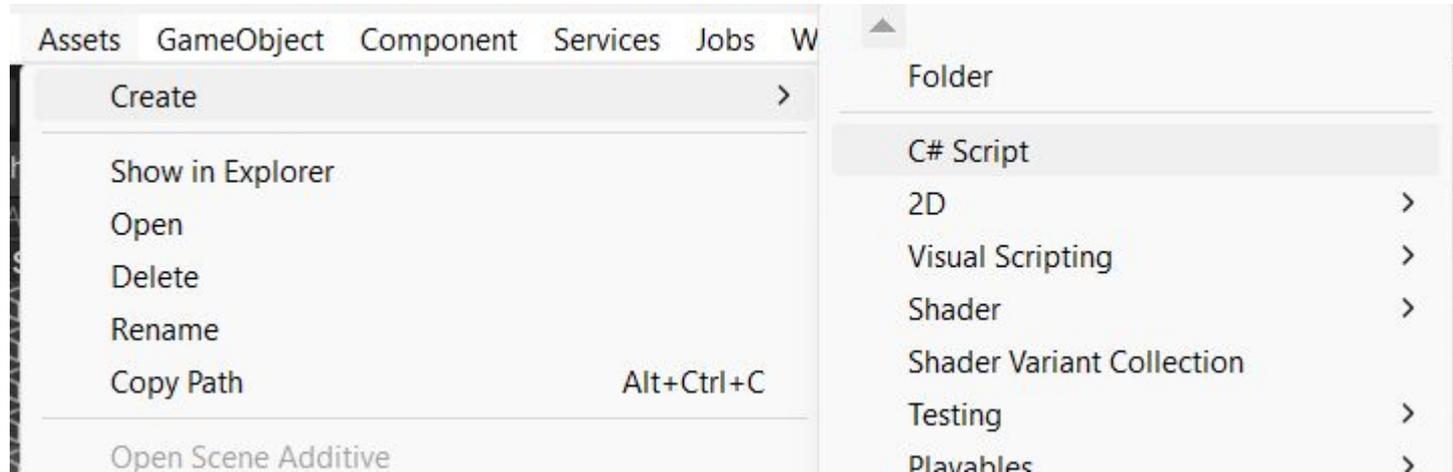
11) Scene:

- As cenas nada mais são do que as telas do jogo;
- Podem ser enxergadas como os níveis do jogo.



12) Scripts:

- Os scripts representam as instruções que podem ser associadas aos *game objects*;
- Podem ser desenvolvidos em UnityScript, C# ou Boo.





12) Scripts:

- Usualmente utilizamos o Visual Studio como IDE para editar os Scripts, mas pode ser utilizado qualquer outro editor;
- Para alterar o editor utilizado basta acessar: “Edit -> Preferences -> External Tools -> External Script Editor”;
- Após criar o Script deve-se adicioná-lo ao *game object* desejado;



12) Scripts:

- **A classe MonoBehaviour:** É a classe de base, da qual todo Script Unity deriva.
- **Método Start():** É chamado quando um Script é habilitado;
- **Método Update():** É chamado a cada quadro;
- **Método FixedUpdate():** Semelhante ao Update, porém com intervalo de tempo fixo.
- **Debug.Log():** Permite apresentar um texto no log do console;
- **Input.GetAxisRaw():** Retorna o valor do eixo virtual identificado pelo parâmetro. Seus valores variam de -1 a 1;

Dúvidas?

Profa. Thaiana Pereira dos Anjos Reis, Dra. Eng.

thaiana.anjos@ifsc.edu.br

Prof. Roberval Silva Bett, Me. Eng.

roberval.bett@ifsc.edu.br