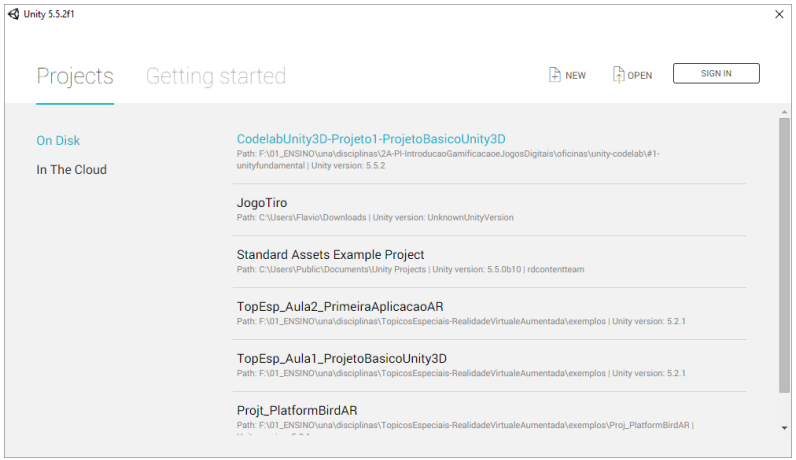
Unity Codelab

Prática 1: Introdução ao Unity3D

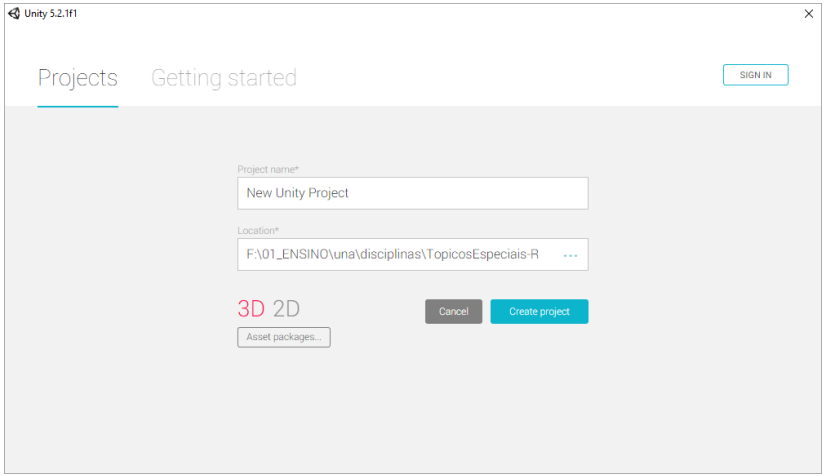
Prof. Flávio A. R. Calado (flavio.calado@prof.una.br)

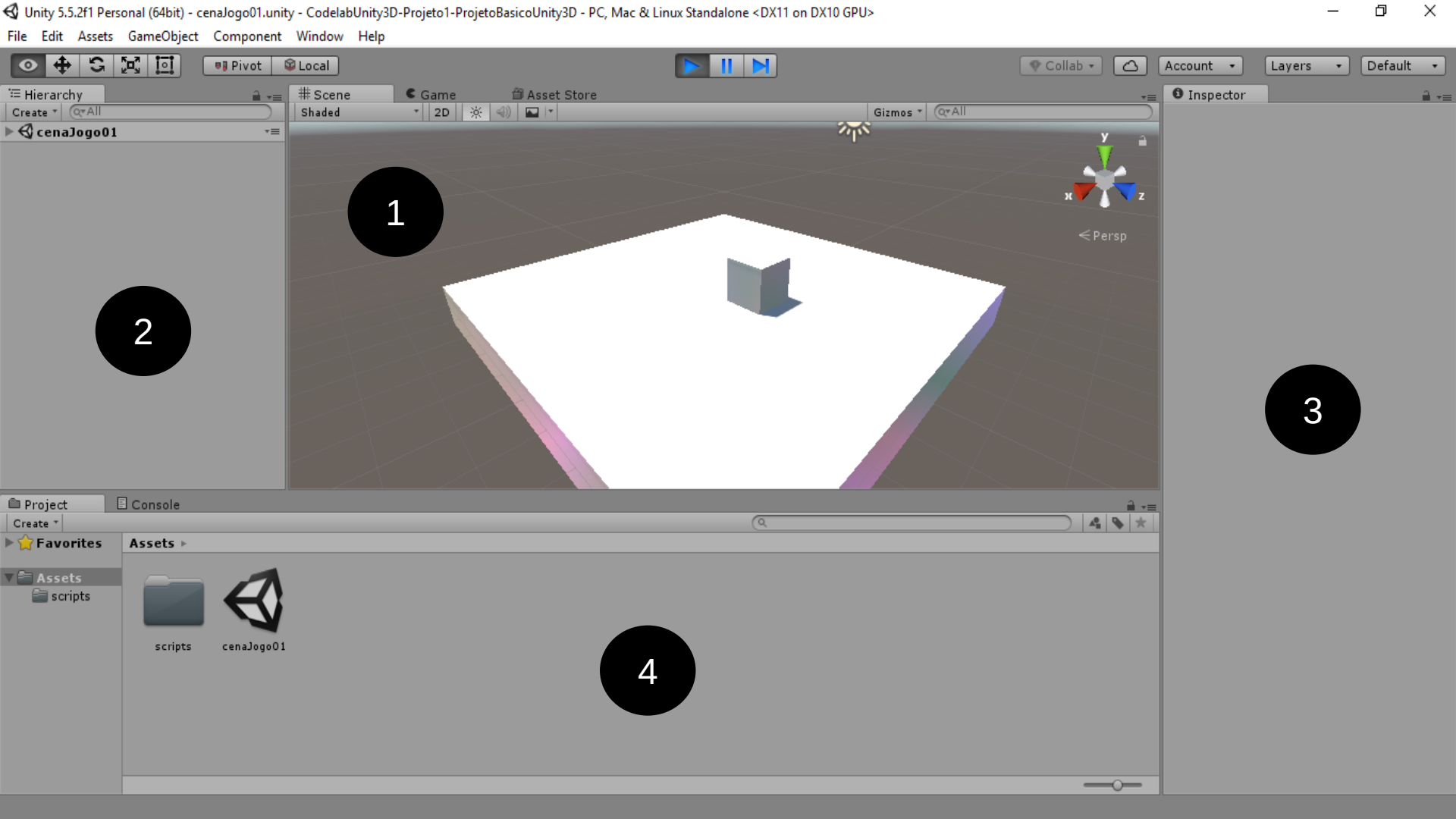
Criando um Novo Projeto

Unity > New



Projects > Projeto 3D > Create Project




Unity 5.5.2f1 Personal (64bit) - cenaJogo01.unity - CodelabUnity3D-Projeto1-ProjetoBasicoUnity3D - PC, Mac & Linux Standalone <DX11 on DX10 GPU>
File
Edit
Assets
GameObject
Component
Window
Help
Pivot
Local
Collab
Account
Layers
Default
Hierarchy
Create
All
cenaJogo01
Scene
Game
Asset Store
Shaded
2D
Gizmos
All
y
x
z
Persp
Inspector
1
2
3
4
Project
Console
Create
Favorites
Assets
Assets
scripts
scripts
cenaJogo01

Interface do Unity

- **Scene** [1]—onde os objetos do jogo são colocados o jogo é construido
- **Hierarchy** [2]—lista de todos os elementos que se encontram na cena do jogo
- **Inspector** [3]—as propriedades do objeto selecionado durante a construção do jogo
- **Project** [4]— Árvore de pastas e arquivos do jogo

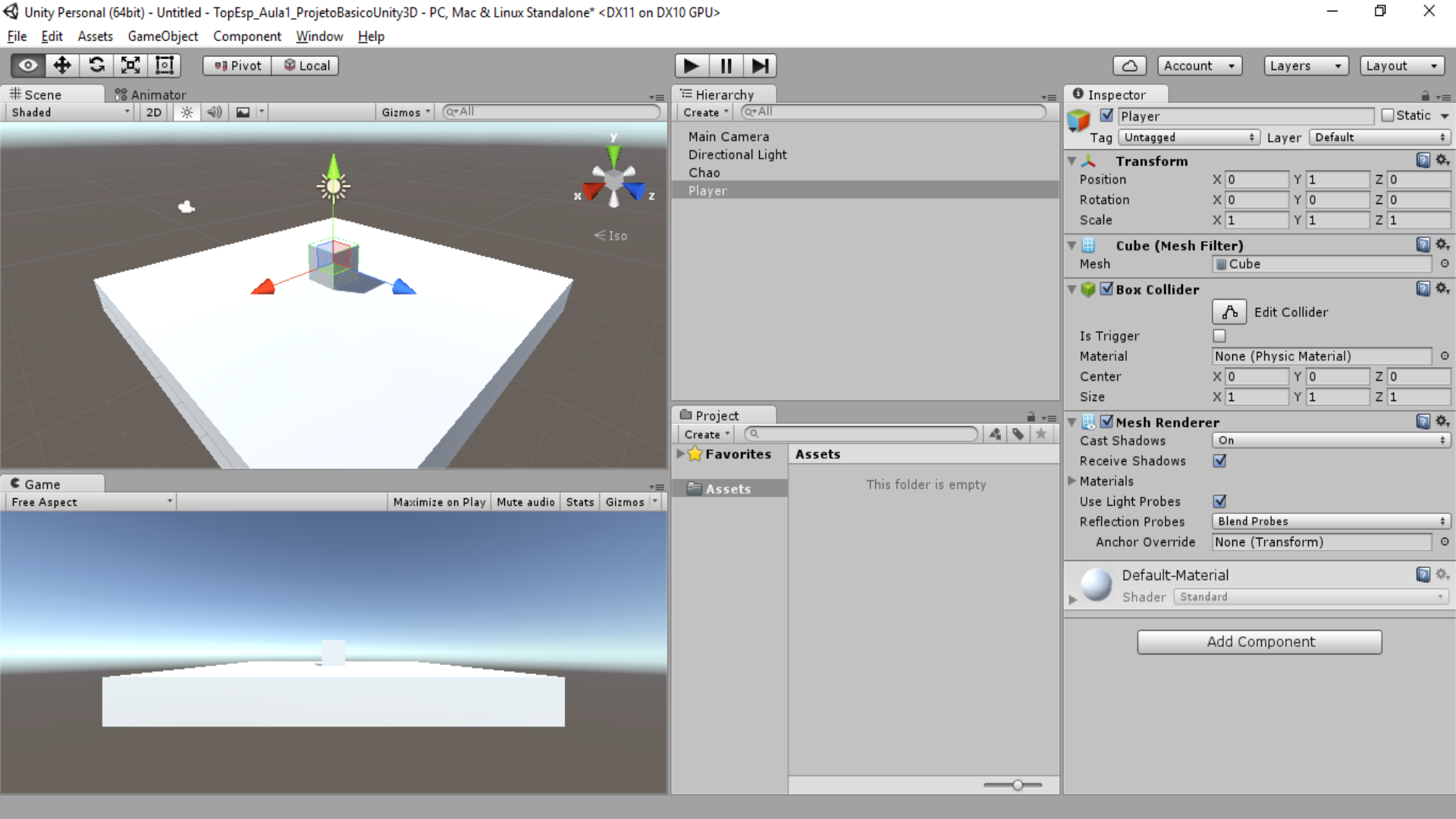
Criando nossa primeira Cena

Cena = tudo que está na aplicação e como aparecerá para o usuário

1. Objetos em Cena
2. Câmera
3. Iluminação
4. Comportamento dos Objetos
5. Interação com o usuário

Objetos e suas Características

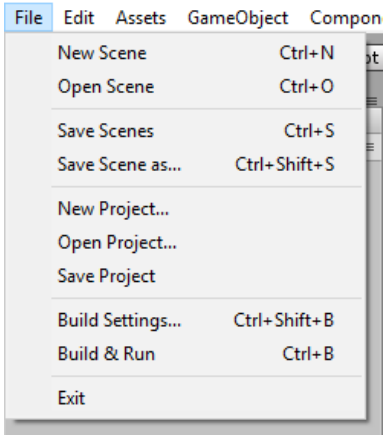
1. Criando o terreno da aplicação
   - GameObject > 3D Object > Cube
     1. Alterar o nome do objeto: Hierarchy > Cube > (F2) “Terreno”
     2. Altera o tamanho do objeto: Inspector > Transform > Scale (X: 10, Y: 1, Z: 10)
2. Criando o objeto de interação com o usuário
   - GameObject > 3D Object > Cube
     1. Alterar o nome: Hierarchy > Cube > (F2) “Player”
     2. Altera a posição: Inspector > Transform > Position (X: 0, Y: 1, Z: 0)


Unity Personal (64bit) - Untitled - TopEsp_Aula1_ProjetoBasicoUnity3D - PC, Mac & Linux Standalone* <DX11 on DX10 GPU>
File Edit Assets GameObject Component Window Help
Pivot Local
Account Layers Layout
Scene Animator
Shaded 2D Gizmos All
Iso
Hierarchy
Create All
Main Camera
Directional Light
Chao
Player
Inspector
Player
Static
Tag Untagged Layer Default
Transform
Position X 0 Y 1 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Cube (Mesh Filter)
Mesh Cube
Box Collider
Edit Collider
Is Trigger
Material None (Physic Material)
Center X 0 Y 0 Z 0
Size X 1 Y 1 Z 1
Mesh Renderer
Cast Shadows On
Receive Shadows
Materials
Use Light Probes
Reflection Probes Blend Probes
Anchor Override None (Transform)
Default-Material
Shader Standard
Add Component
Project
Create
Favorites
Assets
Assets
This folder is empty
Game
Free Aspect Maximize on Play Mute audio Stats Gizmos

Dica 1:

Nunca se esqueça de salvar o que você já fez

File > Save Scenes

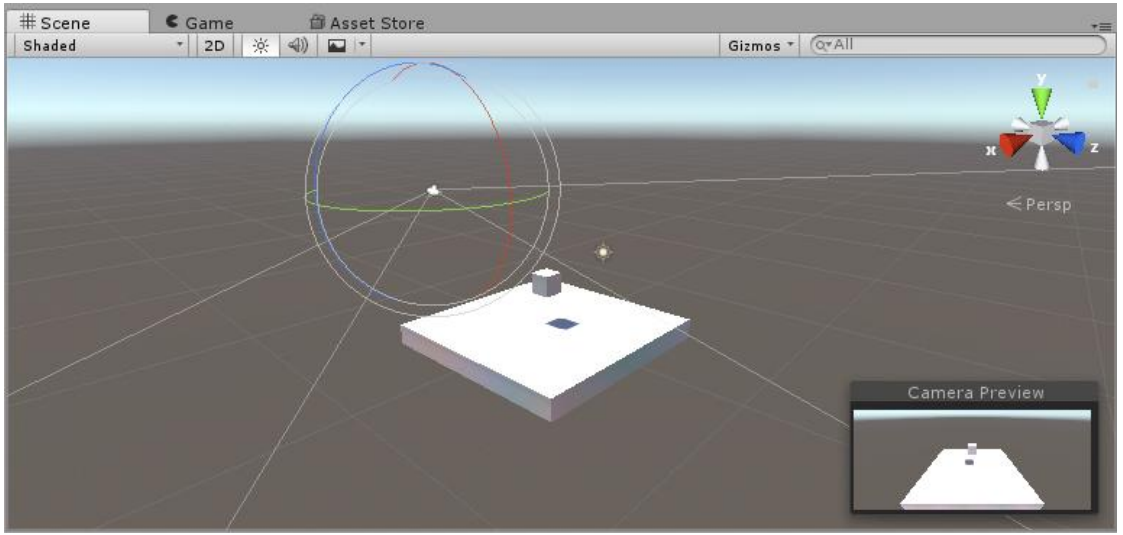


Cinematografia Básica

Posicionando a Câmera

- Selecione: Hierarchy > Main Camera
- Com o mouse

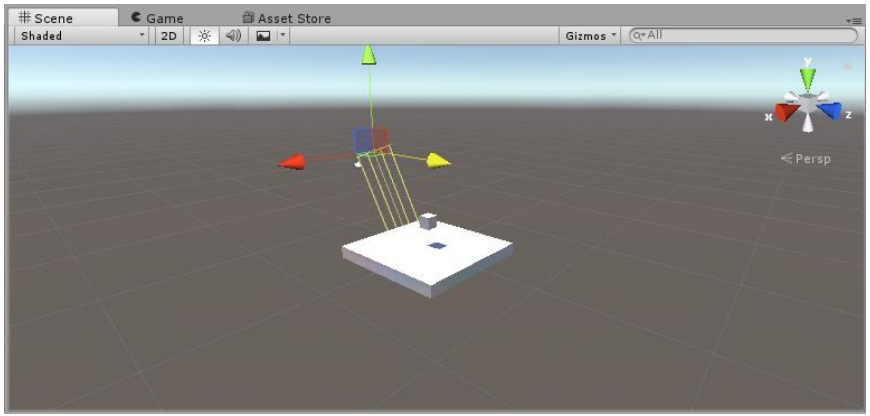
Girar em torno do eixo



Iluminação Básica

Configurando a Iluminação

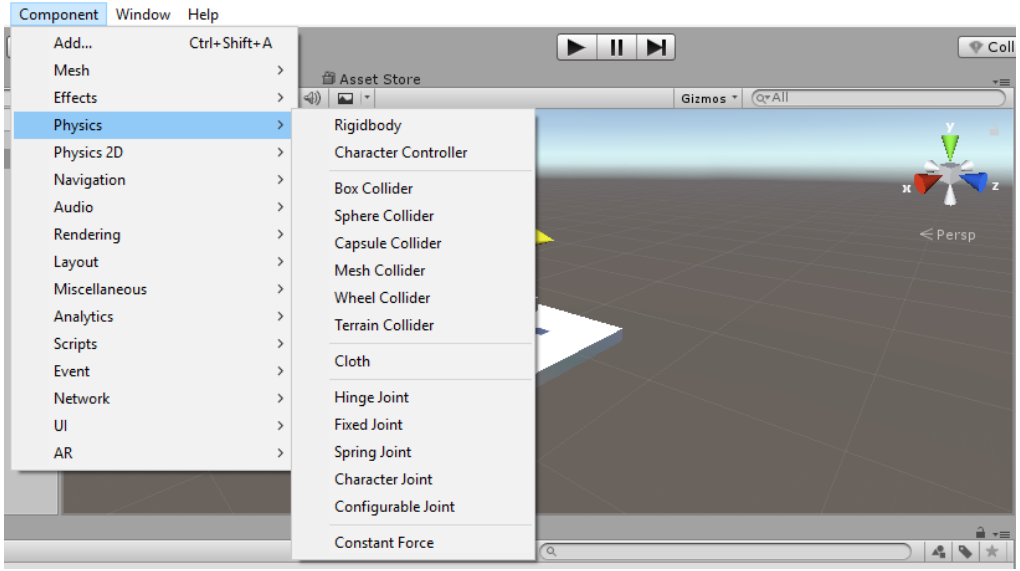
- Hierarchy > Directional Light
- Inspector > Type > Directional



Pratica 1:
Demonstração da Visão do Usuário

Comportamento dos Objetos

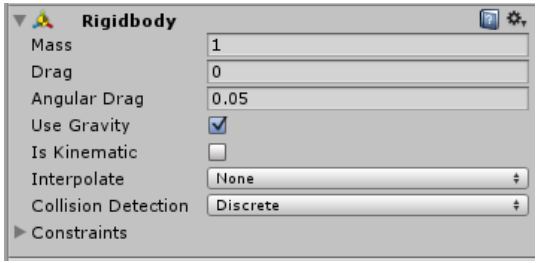
Comportamento como gravidade, detecção de colisões, aimação são módulos pré-fabricados disponíveis



Comportamento dos Objetos

Para fazer com que um objeto adquira um comportamento adicionamos a ele esse componente

1. Selecione o objeto Player na Hierarquia
    1. Hierarchy > Player
2. Para adicionar a física de gravidade ao Player
    1. Inspector > Add Component > Physics > Rigidbody
    2. Inspector > Rigidbody > Use Gravity = true



Pratica 1:
Demonstração da Física

Programando a aplicação

- **Para programar utilizamos scripts que podem ser em C# ou JavaScript**
- Os scripts são os trechos de código executados pela Unity durante a execução da aplicação

Tarefa: Criar uma pasta para os Scripts

1. Project > Create > Folder : “Assets/scripts”

Interação com o Usuário

Criando um Script para o Player

1. Selecione o Player na Hierarquia
2. Inspector > Add Component > New Script
    1. Define o nome do script: “MovimentaPlayerScript.cs”
    2. Create and Add Script
3. Abra o script na IDE (MonoDevelop ou VisualStudio)

Programando a Interação

Estrutura do Script Unity3D

```
using UnityEngine;
using System.Collections;

public class MovimentaPlayerScript : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }
}
```

Método chamado somente quando a aplicação é iniciada

Método chamado a cada atualização de frames da aplicação

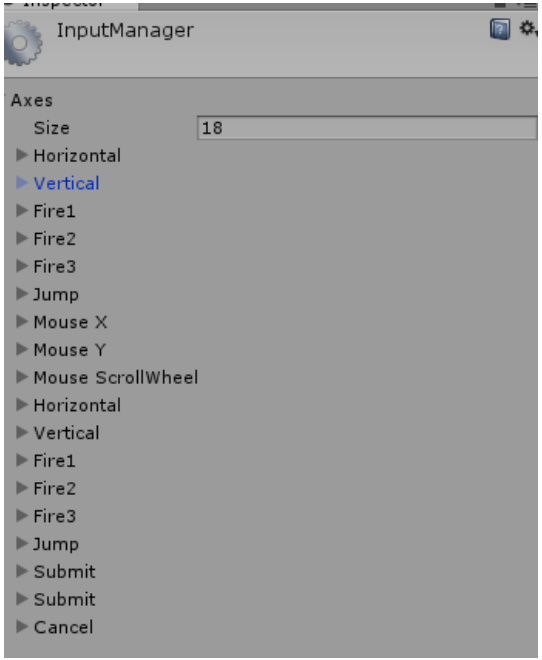
Dica 2:

Nunca se esqueça mover os arquivos para a pasta correta

Interação Básica com o Usuário

Unity > Edit >> Project Settings >> Input

- Setas direcionais do teclado / WASD
- Barras de espaço e botões de disparo
- Movimentação e Botões Mouse
- Controles de console (Joystick)



Programação Unity

É possível ter acesso às propriedades dos objetos através de programação

```
Vector3 transform = transform.position;
```

É possível ter acesso às ações do usuário

```
float setasLaterais = Input.GetAxis("Horizontal");
float setasVerticais = Input.GetAxis("Vertical");
```

Interação Básica com o Usuário

```
// Use this for initialization
void Start () {

}

// Update is called once per frame
void Update () {
    // pega a posição atual do player no jogo
    Vector3 posicao = transform.position;
    float movimentoLateral = Input.GetAxis("Horizontal");
    float movimentoVertical = Input.GetAxis("Vertical");
    // faz a multiplicação do movimento pelo tempo entre os frames
    movimentoLateral = movimentoLateral * Time.deltaTime;
    movimentoVertical = movimentoVertical * Time.deltaTime;
    // aplica os movimentos sobre o player
    transform.Translate(movimentoLateral, 0f, movimentoVertical);
}
```

Pratica 1:
Demonstração Final