

# Unity Crash Course

## 1

EltonCN  
CC BY-NC 4.0

# Preparação

- Juntar grupos com notebook com a Unity
- Checar se instalaram o editor, não apenas o hub

# Introdução

- Slide: **Outline** (2)
- Slide: **Game Engine** (3-7)
  - Por que uma game engine? (5)
    - Doom: divisão entre core e jogo, licenciamento para outros jogos e criação de mods
  - Generalização x Otimização (6)
    - Engines generalistas: Unreal, Unity
  -

# Criando um projeto

- Mostrar onde obter **versão de estudante**
  - <https://unity.com/products/unity-student>
- Slide: **Versões** do editor e **pipelines** (8-10)
- Mostrar como baixar uma versão do Editor e **criar projeto**

# Básicos

- Mostrar **interface do editor** (Scene, Hierarchy, Inspector, Game, Project)
- Mostrar **GameObject** (adicionar, remover, mover, hierarquia, desabilitar)
- Mostrar **componentes** (tipos de componentes, transform, adicionar, remover)
- Mostrar **física**, Rigidbody, collider (tipos),
  - **Exemplo [1]** com gravidade e colisão (com e sem quicar)

# Exemplo [1] - Física

- Criar esfera
- Executar: faz nada
- Adicionar Rigidbody
- Executar: cai infinitamente
- Desativar gravidade e executar
- Adicionar plano
- Executar: cai até bater no plano
- Duplicar esfera e randomizar posição
- Executar: caem e colidem, mas sem quicar
- Criar e configurar PhysicalMaterial
- Inserir material nas esferas
- Executar: quicam

# Código

- Mostrar documentação sobre **script loop**
  - <https://docs.unity3d.com/Manual/ExecutionOrder.html>
- Mostrar **visual script**
  - **Usos:** prototipagem rápida, permitir não-programadores modificarem/estenderem um projeto
  - **Exemplo [2]:** mudar posição (OnStart e OnUpdate)
  - **Exemplo [3]:** mover até bater em parede
- Mostrar **MonoBehavior**
  - **Callbacks:** Start, Update, FixedUpdate, OnEnable/Disable, OnValidate
  - **Referências** e caching
  - **Exemplo [4]:** mudar posição (OnStart e OnUpdate)
  - **Exemplo [5]:** mover até bater em parede (deslocar normal/rigidbody, time)
- Mostrar **Input**
  - **Sistema** antigo x novo
  - **Exemplo [6][7]:** mover quando pressionado botão (antigo e novo)

# Exemplo [2] - Visual Scripting posição

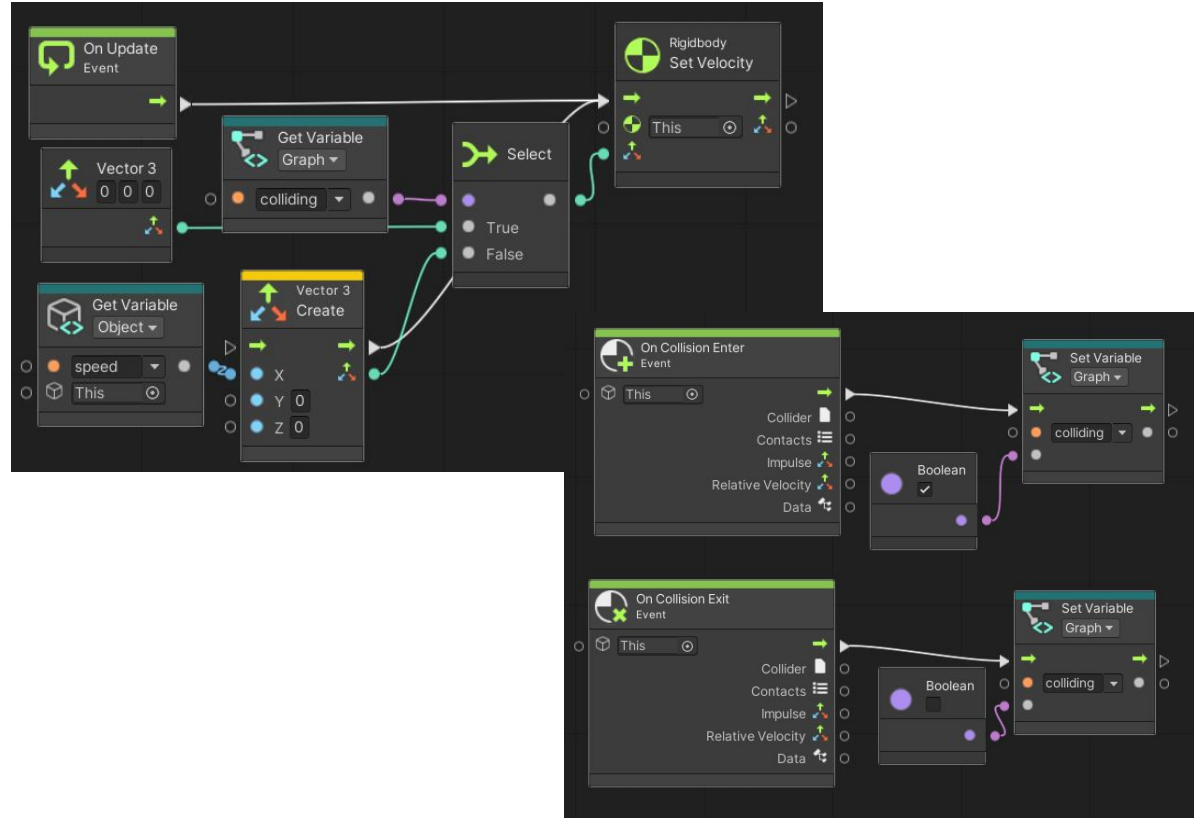
- Adicionar objeto
- Adicionar ScriptMachine
- Criar asset VisualScripting
- Programar OnStart
  - Variável “startPosition”
- Executar: muda posição no início
- Programar OnUpdate
  - Variável “afterTimePosition”
  - Mover após 3 segundos
- Executar: muda posição no início e após 3 segundos





# Exemplo [3] - Visual Scripting Colisão

- Criar esfera e cubo
- Adicionar Rigidbody na esfera
  - Desativar gravidade
- Programar OnUpdate
  - Velocidade para frente
  - Variável “speed” = 1
- Executar: movimenta, mas ignora o cubo
- Programar colisões
- Executar: movimenta e para no cubo



## Exemplo [4] - Scripting posição

- Adicionar objeto
- Adicionar script ChangePosition
- Mudar posição no Start
- Executar
- Trocar para variável
  - **Serializar** variável
- Programar timer com o OnUpdate

```
[SerializeField] Vector3 startPosition;  
[SerializeField] Vector3 afterTimePosition;
```

```
void Start()  
{  
    transform.position = startPosition;  
}
```

```
void Update()  
{  
    if (Time.time > 3)  
    {  
        transform.position =  
afterTimePosition;  
    }  
}
```

## Exemplo [5] - Scripting colisão

- Mesmo setup do [2]
- Pegar referência para o Rigidbody no Start
  - Explicar **caching** e custos
  - GetComponent
- Programar ir para frente
- Executar
- Programar colisão
- Executar

```
void Start ()
{
    colliding = false;
    rb = GetComponent<Rigidbody>();
}

void Update ()
{
    if(colliding){
        rb.velocity = Vector3.zero;
    }
    else{
        rb.velocity = new Vector3(velocity, 0, 0);
    }
}

void OnCollisionEnter ()
{
    colliding = true;
}

void OnCollisionExit ()
{
    colliding = false;
}
}
```

# Exemplo [6] - Sistema antigo de input

- Criar cena com esfera
- Criar script para alterar posição da esfera com o teclado
- Adicionar a esfera e executar
  - Esfera muda de posição com a tecla

```
public class Ref_InputOld : MonoBehaviour {  
  
    [SerializeField] Vector3 targetPosition;  
  
    void Update() {  
        if (Input.GetKey(KeyCode.Space)) {  
            transform.position =  
targetPosition;  
        }  
    }  
}
```

# Exemplo [7] - Sistema novo de input

- Criar cena com esfera
- Criar InputActions
  - DefaultMap
  - Action: MovePosition
    - Botão
    - Binding com tecla do teclado
- Criar script e adicionar a esfera
  - Método OnKey para receber tecla e alterar posição
- Criar objeto com Player Input
  - Adicionar map
  - Behavior: Invoke Unity Events
  - Mapear MovePosition -> OnKey
- Executar
  - Esfera muda de posição com a tecla

```
public class Ref_InputNew : MonoBehaviour{  
    [SerializeField] Vector3 targetPosition;  
  
    public void  
    OnKey(InputAction.CallbackContext context)  
    {  
        transform.position = targetPosition;  
    }  
}
```

# Organização de um projeto

- Mostrar: **prefabs** e variants
  - Alterações na cena, sobreescrita do prefab e sobreescrita da variant
- Mostrar: **cenias**
- Mostrar: cenias **aditivas**
- Mostrar: **templates** de cenias

# Versionamento

- Slide: Versionamento (11-13)
  - Orientar sobre materiais online sobre git

# Fim

- Slide: Primeiros passos (14-16)