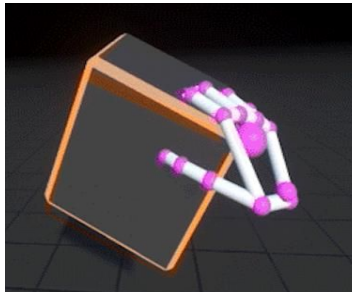
+Scripts

Criando um item ferramenta

Item Ferramenta?



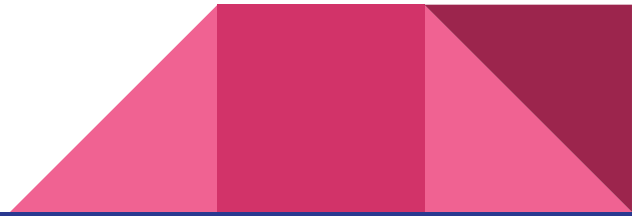
Gravity Gun (Half-life 2)

Etapas lógicas

1 - Ao acionar o gatilho, emitir um raio (p/ descobrir qual é nosso alvo)

2 - Se o alvo atingido for um Rigid Body, atrair para perto

3 - Ao soltar o gatilho, repelir o alvo



Dinâmica da atividade

1 - No Desktop -> Pasta Aula 9 -> abrir o pacote Unity

2 - Puxar o pdf “Tutorial Unity - Aula 9 - Scripts II” do e-disciplinas

3 - Seguir as instruções: quem terminar antes, testará antes

4 - Equipes que terminarem primeiro poderão discutir o trabalho em grupo com os professores / monitores

Script: GravityRay

Primeiro vamos criar um raio que irá colidir com os Obstáculos em cena.
Este raio é uma linha reta contínua, que tem uma **posição inicial**, uma **direção**, um **alcance** e armazena **informações da colisão**.
Se o raio atingir um objeto, iremos armazená-lo em uma variável.

- Carregar o package da aula 9 e abrir a cena de teste
- Criar um novo script e nomear GravityRay
- Arrasta-lo sobre o objeto GravityWandTests
- Abrir o script para edita-lo

Raybeam

```
5  public class GravityRay : MonoBehaviour {
6      private Rigidbody target; //variavel para armazenar objeto capturado pelo raio
7
8      public void RaybeamStart(){
9          RaycastHit hitInfo; // variavel para armazenar informação da colisão do raio
10
11         //cria um raio e SE houver alguma colisão//
12         if (Physics.Raycast (transform.position, transform.forward, out hitInfo, 100)) {
13             //método para desenhar o raio (no editor)
14             Debug.DrawRay (transform.position, transform.forward * hitInfo.distance, Color.green);
15
16             //SE houver um rigidbody no objeto atingido
17             if (hitInfo.collider.attachedRigidbody != null) {
18                 target = hitInfo.collider.attachedRigidbody;
19             }
20         }
21     }
```

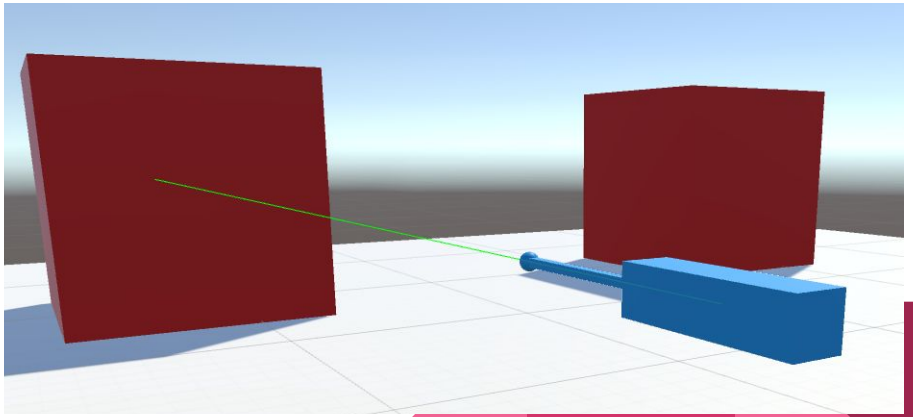
Raybeam (cont.)

```
23      public void RaybeamStop(){
24          target = null;
25      }
26
27      void Update(){
28          // para facilitar os testes, vamos pegar o input do teclado
29          // se a tecla espaço for pressionada, ativar o raio.
30          // se a tecla espaço for solta, desativar o raio.
31          if (Input.GetKeyDown ("space")) {
32              RaybeamStart ();
33          }
34          if (Input.GetKeyUp ("space")) {
35              RaybeamStop ();
36          }
37      }
38  }
```

Testes

Play na cena, selecione o GravityWandTests e pressione espaço para disparar

Veja o resultado na janela do editor (o raio dura apenas 1 frame)



GravityRay : Attract

```
5 public class GravityRay : MonoBehaviour {
6     private Rigidbody target; //variavel para armazenar objeto capturado pelo raio
7     private bool isActive = false; //o raio está atraindo um alvo?
```

Criar novo Método:

```
31     private void Attract(){
32         // desliga a fisica para podermos mover o objeto
33         target.isKinematic = true;
34         //move o objeto até a posição da arma com espaçamento de 1 metro
35         target.MovePosition (this.transform.position + transform.forward);
36     }
```

GravityRay: Attract (cont.)

```
9     public void RaybeamStart(){
10        RaycastHit hitInfo; // variavel para armazenar informação da colisão do raio
11
12        //cria um raio e SE houver alguma colisão//
13        if (Physics.Raycast (transform.position, transform.forward, out hitInfo, 100)) {
14            //método para desenhar o raio (no editor)
15            Debug.DrawRay (transform.position, transform.forward * hitInfo.distance, Color.green);
16
17            //SE houver um rigidbody no objeto atingido
18            if (hitInfo.collider.attachedRigidbody != null) {
19                target = hitInfo.collider.attachedRigidbody;
20                isActive = true;
21            }
22        }
23    }
```

```
36    void Update(){
37        if (isActive) {
38            Attract();
39        }
```

```
25    public void RaybeamStop(){
26        isActive = false;
27        target = null;
28    }
```

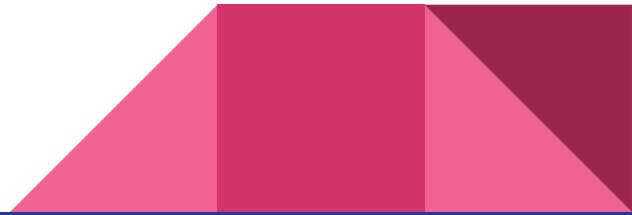
Teste 2

Play na cena, selecione o GravityWandTests e pressione espaço para disparar

Veja o resultado na janela do editor

Experimente atrair diferentes cubos

Observe que a física deles permanece desligada ao solta-los!



GravityRay : Repel

Criar novo Método:

```
37      private void Repel(){
38          // reativa a fisica
39          target.isKinematic = false;
40          // aplica uma força em uma direção
41          target.AddForce (this.transform.forward * 10f, ForceMode.Impulse);
42      }
```

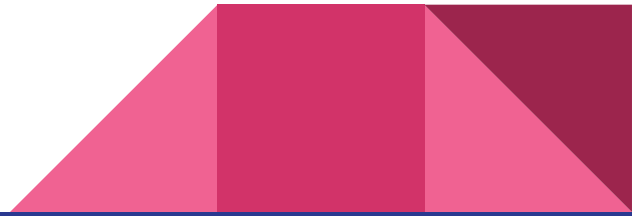
```
25      public void RaybeamStop(){
26          isActive = false;
27          Repel ();
28          target = null;
29      }
```

Teste 3 - final!

Play na cena, selecione o GravityWandTests e pressione espaço para disparar

Veja o resultado na janela do editor

Experimente atrair e repelir diferentes cubos



Criando a ferramenta final

Agora que o script está pronto não vamos mais utilizar o wand de testes. Iremos utilizar um prefab já criado para esta aula.

- Na pasta Prefabs, localize e selecione o GravityWand
- Adicione o seu componente: GravityRay

Configurando o Wand

Configure (conforme a figura ao lado)

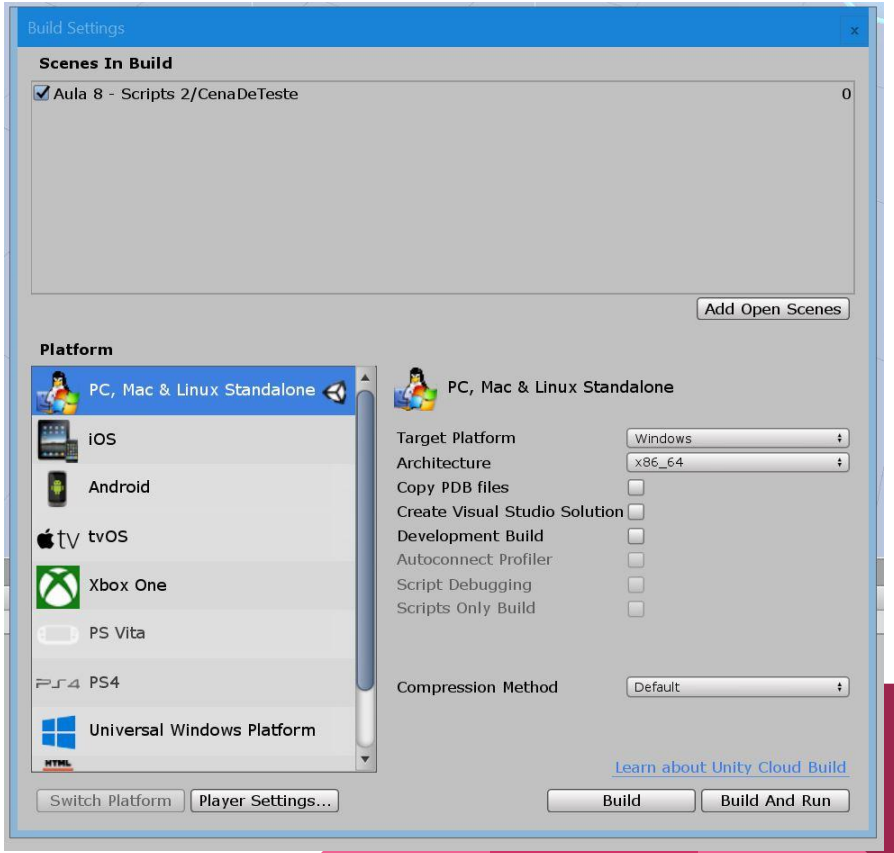
- Criar um evento OnTriggerDown, arrastar o script GravityRay e indicar o método RaybeamStart
- Fazer o mesmo para OnTriggerUp, indicando o método RaybeamStop

Compilando

Para testar é necessário rastrear a mão… Logo, passar para o Rift CV1

File> BuildSettings> Add Open Scene (se necessario) > Build!

Abrir o software Dukto (no desktop) e enviar o arquivo para o computador de testes com Rift CV1 (perguntar o nome do computador a um monitor)



Exercícios p/ casa - próximo abre dia 11

- Óculos anaglifo - devolver
  - Adriano Carvalho e Sousa
  - Augusto Ruy Machado
  - Lincoln Makoto Kawakami
  - Marcos Roberto Franco Filho
  - Rodrigo M. Magaldi
  - Igor Freitas Sym
  - Rodrigo Zanette de Magalhães
  - **Quem ainda não devolveu**
- Pegar - devolver próxima aula
  - Bruno Akio Shirasuna
  - Bruno da Costa Braga
  - Bruno Hisashi Otsuka
  - Bruno Mucha Pasini
  - Clarissa Alves Barreto da Rocha
  - Eric Nozomi Tatsuta
  - Felipe Igai Wang
  - Gabriel de Souza Oliva
  - Helder Seiji Tanaka Costa
  - Henrique Tasaki Imaeda
  - Leonardo Gushiken Yoshitake
  - Matheus Perelmutter
  - Rafael Carvalho Santos
  - Rafael Costa Sales
  - Rafael Szylewicz Levy
  - Rodrigo Rodrigues Gesuatto
  - Sungwon Yoon
  - Vitor Augusto Martin

Projeto - planejamento

- Para semana que vem (entrega pelo representante no e-disciplinas):
  - Setup básico do projeto Unity:
    - Bibliotecas: SteamVR, outras
    - Organização: nomenclatura, pastas e cenas,
    - Player na cena,
  - Planejamento:
    - Listagens e descrição dos prefabs básicos e suas partes
    - Prospecção de recursos: consulta de modelos online (listar, registrar fonte e licença de uso)

Proxima aula

Modelos, texturas, materiais!

Como agregar valor ao seu protótipo
mesmo sem saber modelar