

Aula Prática – Godot 3D (3ª parte)

Atividade 1 – Construindo um Ambiente Físico 3D;

- 1- Crie um novo projeto;
- 2- Adicione um Spatial na Scene inicial;
- 3- Adicione 2 objetos Rigidbody, um objeto Camera e um objeto DirectionalLight;
- 4- Adicione um Quad e um CollisionShape no primeiro Rigidbody;
- 5- Configure a propriedade Mode do primeiro Rigidbody para Static;
- 6- Configure a propriedade Axis do Quad para Y;
- 7- Configure a propriedade Shape do CollisionShape para PlaneShape;
- 8- Adicione um TestCube e um CollisionShape no segundo Rigidbody;
- 9- Configure a propriedade Shape do CollisionShape para BoxShape;
- 10- Redimensione o Quad, o TestCube e os CollisionShapes conforme a figura abaixo;
- 11- Execute o projeto e veja o resultado.

Atividade 2 – Rola-Bola;

- 1- Continuando a partir da atividade anterior, remove TestCube e carrega em seu lugar um modelo de bola de futebol (<http://tf3dm.com/3d-model/soccer-ball-80310.html>) usando MeshInstance;
- 2- Configure a propriedade Shape do CollisionShape da bola para SphereShape;
- 3- Selecione o Rigidbody do Quad e mude a propriedade Shape do CollisionShape para BoxShape. Posicione o BoxShape embaixo do Quad;

- 4- Adicione o script abaixo no Rigidbody da bola:

extends Rigidbody

func _ready():

set_process(true)

func _process(delta):

if(Input.is_key_pressed(KEY_UP)):

self.apply_impulse(self.get_translation(), Vector3(0,0,-0.1))

if(Input.is_key_pressed(KEY_DOWN)):

self.apply_impulse(self.get_translation(), Vector3(0,0,0.1))

if(Input.is_key_pressed(KEY_LEFT)):

self.apply_impulse(self.get_translation(), Vector3(-0.1,0,0))

if(Input.is_key_pressed(KEY_RIGHT)):

self.apply_impulse(self.get_translation(), Vector3(0.1,0,0))

- 5- Adicione o script abaixo na Camera:

extends Spatial

var offset

```
func _ready():  
    offset = self.get_translation() - get_node("/root/Spatial/RigidBody 2").get_translation()  
    set_fixed_process(true)  
  
func _fixed_process(delta):  
    self.set_translation(get_node("/root/Spatial/RigidBody 2").get_translation() + offset)
```

6- Execute o projeto e veja o resultado.

Atividade 3 – Alvos Kinemáticos;

- 1- Continuando a partir da atividade anterior, adicione um RigidBody ao Spatial inicial e configure as propriedades Contact Monitor para true , Contacts Report para 1 e Mode para Kinematic;
- 2- Adicione um TestCube e um CollisionShape no novo RigidBody;
- 3- Configure no novo CollisionShape as propriedades Trigger para true e Shape para BoxShape;
- 4- Redimensione e posicione o novo RigidBody conforme a imagem abaixo;

5- Adicione o script abaixo no novo RigidBody:

extends RigidBody

```
func _ready():  
    set_fixed_process(true)  
    connect("body_enter",self,"_on_body_enter")  
  
func _on_body_enter( body ):  
    self.hide()  
  
func _fixed_process(delta):  
    rotate(Vector3(0.1,0.2,-0.1), 0.03)
```

6- Execute o projeto (salve a Scene antes) e veja o resultado.

Atividade 4 – HUD de Pontos + Lógica do Jogo;

Finalize o jogo complementando os recursos que faltam. Procure adicionar um HUD de pontos fixo a Camera adicionando um Quad e uma nova Scene de pontos. A criação de uma Scene contendo uma GUI para perguntar se o jogador quer tentar novamente também é uma opção. Procure adicionar barreiras laterais ou reposicione a bola no início quando ela cair. Adicione mais cubos dinâmicos na tela com valores de rotação definidos randomicamente. Adicione um controle de tempo por partida, exibindo uma mensagem com os pontos obtidos após o fim do jogo. O reaparecimento de cubos após um tempo ou quando todos os cubos forem coletados também é uma

opção a ser adicionada. Ganha o jogo quem fizer mais pontos, ou quem coletar todos os cubos em menos tempo.