

ERBASE – Escola Regional de Computação Bahia-Alagoas-Sergipe
Aracaju, 22 de Agosto de 2018
Laboratório de Entretenimento Digital Aplicado (LEnDA) - UEFS
prof. D.Sc. Victor Sarinho

Minicurso – Introdução ao Godot 3 (3ª parte)

Atividade 1 – Colisão de Sprites;

- 1- Crie um novo projeto;
- 2- Adicione um Node2D na Scene inicial;
- 3- Adicione dois Sprites ao Node2D na Scene inicial;
- 4- Carregue a imagem dos Sprites (propriedade Texture) e posicione-os no mesmo nível do eixo Y mas afastados no eixo X;
- 5- Adicione um script com o seguinte código no primeiro Sprite:

extends Sprite

```
var start_pos = Vector2()
var box1 = null
var box2 = null
```

```
func _ready():
    set_process(true)
    start_pos = get_position()
    box1 = RectangleShape2D.new()
    box2 = RectangleShape2D.new()
```

```
func _process(delta):
    # Get a reference to the other sprite
    var sprite2 = get_node("/root/Node2D/Sprite2")

    # Update our location
    self.move_local_x(0.5)

    # set the boundaries of each RectangleShape2D to those of the texture making up our sprite
    # values are relative to center, thus half width and height
    box1.set_extents(Vector2(self.get_texture().get_size().x/2,self.get_texture().get_size().y/2))

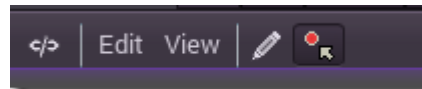
    box2.set_extents(Vector2(sprite2.get_texture().get_size().x/2,sprite2.get_texture().get_size().y/2))

    #Now check to see if box1 at sprite1's pos collided with box2 and sprite2's position
    if(box1.collide(get_transform(),box2,sprite2.get_transform())):
        set_position(start_pos) # it did, so reset position to beginning, what's old is new
again!
```

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

Atividade 2 – Trabalhando com Rigidbody2D, StaticBody2D e KinematicBody2D;

- 1- Crie um novo projeto;
- 2- Adicione um Node2D na Scene inicial;
- 3- Adicione um Rigidbody2D ao Node2D na Scene inicial;
- 4- Adicione um Sprite ao Rigidbody2D e carregue uma imagem ao Sprite;
- 5- Posicione o Rigidbody2D no centro da tela;
- 6- Execute o projeto (salve a Scene antes) e veja o resultado;
- 7- Acesse a opção do menu /Scene/Project Settings, selecione a opção Physics 2D e altere os valores padrões de gravidade (*default_gravity*, *defult_gravity_vector*);
- 8- Execute o projeto (salve a Scene antes) e veja o resultado;
- 9- Selecione Rigidbody2D e altere o valor da propriedade Weight pelo Inspector;
- 10- Execute o projeto (salve a Scene antes) e veja o resultado;
- 11- Adicione um CollisionShape2D ao Rigidbody2D;
- 12- Selecione a propriedade Shape do CollisionShape2D e defina uma forma ao collider (redimensione-o conforme o tamanho do Sprite);
- 13- Adicione um StaticBody2D ao Node2D na Scene inicial;
- 14- Adicione um Sprite ao StaticBody2D e carregue uma imagem ao Sprite;
- 15- Posicione o StaticBody2D uns 100 pixels abaixo do Rigidbody2D;
- 16- Adicione um CollisionShape2D ao StaticBody2D;
- 17- Selecione a propriedade Shape do CollisionShape2D e defina uma forma ao collider (redimensione-o conforme o tamanho do Sprite);
- 18- Execute o projeto (salve a Scene antes) e veja o resultado;
- 19- Substitua um dos CollisionShape2D por um CollisionPolygon2D;
- 20- Selecione o CollisionPolygon2D, selecione o lápis conforme a figura abaixo e desenhe a forma do polígono de colisão desejado;



- 21- Execute o projeto (salve a Scene antes) e veja o resultado;
- 22- Adicione um KinematicBody2D afastado uns 100 pixels no eixo X do StaticBody2D;
- 23- Adicione um Sprite ao KinematicBody2D e carregue uma imagem ao Sprite;
- 24- Adicione um CollisionShape2D ao KinematicBody2D;
- 25- Selecione a propriedade Shape do CollisionShape2D e defina uma forma ao collider (redimensione-o conforme o tamanho do Sprite);
- 26- Adicione o script abaixo ao KinematicBody2D:

```
extends KinematicBody2D
```

```
func _ready():  
    set_physics_process(true)
```

```
func _physics_process(delta):  
    move_and_collide(Vector2(0.5,0))
```

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

Atividade 3 – Inputs & KinematicBody2D;

- 1- Crie um novo projeto;
- 2- Adicione um Node2D na Scene inicial;
- 3- Adicione um KinematicBody2D ao Node2D do projeto;
- 4- Adicione um Sprite ao KinematicBody2D e carregue uma imagem ao Sprite;
- 5- Adicione um CollisionShape2D ao KinematicBody2D;
- 6- Selecione a propriedade Shape do CollisionShape2D e defina uma forma ao collider (redimensione-o conforme o tamanho do Sprite);
- 7- Adicione o script abaixo ao KinematicBody2D:

```
extends KinematicBody2D
```

```
func _ready():  
    set_process_input(true)
```

```
func _input(event):  
    # on keyboard cursor key  
    if(event is InputEventKey):  
        if(event.scancode == KEY_RIGHT):  
            move_and_collide(Vector2(1,0))  
        if(event.scancode == KEY_LEFT):  
            move_and_collide(Vector2(-1,0))  
        if(event.scancode == KEY_UP):  
            move_and_collide(Vector2(0,-1))  
        if(event.scancode == KEY_DOWN):  
            move_and_collide(Vector2(0,1))
```

- 8- Adicione um StaticBody2D ao Node2D na Scene inicial;
- 9- Adicione um Sprite ao StaticBody2D e carregue uma imagem ao Sprite;
- 10- Posicione o StaticBody2D uns 100 pixels abaixo do KinematicBody2D;
- 11- Adicione um CollisionShape2D ao StaticBody2D;
- 12- Selecione a propriedade Shape do CollisionShape2D e defina uma forma ao collider (redimensione-o conforme o tamanho do Sprite);
- 13- Execute o projeto (salve a Scene antes), mova o KinematicBody2D ao StaticBody2D e veja o resultado;

- 14- Substitua o script do KinematicBody2D pelo script abaixo:

```
extends KinematicBody2D
```

```
const speed = 100
```

```
func _ready():  
    set_physics_process(true)
```

```
func _physics_process(delta):
```

```

var direction = Vector2(0,0)
if (Input.is_key_pressed(KEY_UP)):
    direction += Vector2(0,-1)
if (Input.is_key_pressed(KEY_DOWN)):
    direction += Vector2(0,1)
if (Input.is_key_pressed(KEY_LEFT)):
    direction += Vector2(-1,0)
if (Input.is_key_pressed(KEY_RIGHT)):
    direction += Vector2(1,0)

move_and_collide(direction * speed * delta)

```

15- Execute o projeto (salve a Scene antes), mova o KinematicBody2D ao StaticBody2D e veja o resultado;

Atividade 4 – Tratando Colisões;

1- Utilizando o projeto da atividade anterior, substitua o script do KinematicBody2D pelo script abaixo:

```

extends KinematicBody2D

const speed = 100

func _ready():
    set_physics_process(true)

func _physics_process(delta):
    var direction = Vector2(0,0)
    if (Input.is_key_pressed(KEY_UP)):
        direction += Vector2(0,-1)
    if (Input.is_key_pressed(KEY_DOWN)):
        direction += Vector2(0,1)
    if (Input.is_key_pressed(KEY_LEFT)):
        direction += Vector2(-1,0)
    if (Input.is_key_pressed(KEY_RIGHT)):
        direction += Vector2(1,0)

    var collision = move_and_collide(direction * speed * delta)

    if collision:
        print ("Collision with ", collision.collider.get_name())

```

2- Renomeie o KinematicBody2D para **Player** e o StaticBody2D para **Wall**;

3- Execute o projeto (salve a Scene antes), mova o **Player** ao **Wall** e veja o resultado;

- 4- Adicione um Area2D ao Node2D na Scene inicial;
- 5- Adicione um Sprite ao Area2D e carregue uma imagem ao Sprite;
- 6- Posicione o Area2D uns 100 pixels afastado do KinematicBody2D;
- 7- Adicione um CollisionShape2D ao Area2D;
- 8- Selecione a propriedade Shape do CollisionShape2D e defina uma forma ao collider (redimensione-o conforme o tamanho do Sprite);
- 9- Adicione o script abaixo ao Area2D:

```
extends Area2D
```

```
func _ready():
```

```
get_node("/root/Node2D/Area2D").connect("body_entered",self,"_on_Area2D_body_enter")  
get_node("/root/Node2D/Area2D").connect("body_exited",self,"_on_Area2D_body_exit")
```

```
func _on_Area2D_body_enter( body ):  
print("Entered Area2D with body ", body)
```

```
func _on_Area2D_body_exit( body ):  
print("Exited Area2D with body ", body)
```

- 10- Execute o projeto (salve a Scene antes), mova o **Player** ao Area2D e veja o resultado;

- 11- Adicione no final do método `_ready()` do script do KinematicBody2D o trecho de código abaixo:

```
add_to_group("player")
```

- 12- Substitua o método `_on_Area2D_body_enter` do script do Area2D pelo código abaixo:

```
func _on_Area2D_body_enter( body ):  
if body in get_tree().get_nodes_in_group("player"):  
print("Player Entered Area2D with body ", body)  
else:  
print("Entered Area2D with body ", body)
```

- 13- Execute o projeto (salve a Scene antes), mova o **Player** ao Area2D e veja o resultado.

Atividade 5 – Meu Segundo Jogo – Labirinto Físico;

Construa um labirinto formado por vários objetos Area2D. O personagem representado por um KinematicBody2D deve desviar destes objetos e chegar ao final do labirinto no menor tempo. Ao tocar em um destes objetos o personagem deve voltar para o local de origem. Um HUD de tempo deve ser exibido para o jogador. Uma mensagem deve ser exibida para o usuário quando ele concluir o jogo informando o tempo total gasto. Se o jogador demorar mais do que 5 min, o jogo deve informar “Fim de Jogo”.

Atividade 6 – Meu Terceiro Jogo – Flappy Bird cover;

Construa um clone simplificado do jogo Flappy Bird. Adicione apenas 3 fileiras de cano para o pássaro atravessar. O personagem representado por um `RigidBody2D` receberá forças direcionais conforme as entradas do usuário. O jogo finaliza após atravessar os três canos.