**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 (4ª parte)**

**Atividade 1 – Sprites Animados;**

1- Crie um novo projeto;

2- Adicione um Node2D na Scene inicial;

3- Adicione um StaticBody2D ao Node2D na Scene inicial;

4- Adicione um Sprite ao StaticBody2D e carregue uma imagem ao Sprite;

5- Posicione o StaticBody2D no centro da tela (viewport);

6- Adicione um CollisionShape2D ao StaticBody2D;

7- Selecione a propriedade Shape do CollisionShape2D e defina uma forma ao collider (redimensione-o conforme o tamanho do Sprite);

8- Adicione um RigidBody2D ao Node2D na Scene inicial;

9- Adicione um AnimatedSprite ao RigidBody2D;

10- Selecione a propriedade frames do AnimatedSprite e clique na opção “new SpriteFrames”;

11- Selecione novamente a propriedade frames do AnimatedSprite e clique na opção “edit”;

12- Carregue quantas imagens você achar necessário em frames para gerar a animação desejada;

13- Adicione o script abaixo no AnimatedSprite:

*extends AnimatedSprite*

*var tempElapsed = 0*

*func \_ready():*

*set\_process(true)*

*func \_process(delta):*

*tempElapsed = tempElapsed + delta*

*if(tempElapsed > 0.5):*

*if(get\_frame() == self.get\_sprite\_frames().get\_frame\_count("default")-1):*

*set\_frame(0)*

*else:*

*self.set\_frame(get\_frame() + 1)*

*tempElapsed = 0*

*print(str(get\_frame() + 1))*

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

**Atividade 2 – Trabalhando com AnimationPlayer;**

1- Continuando a partir do projeto anterior, selecione a propriedade “script” do AnimatedSprite e clique na opção “clear”;

2- Adicione um AnimationPlayer ao RigidBody2D;

3- Crie uma nova animação no AnimationPlayer com o nome “animation1”:

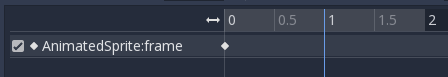


4- Configure a animação para 2 segundos:



5- Selecione o AnimatedSprite e clique na chave do lado direito da propriedade “frame”. Esta ação vai criar uma nova track a ser animada pelo AnimationPlayer;

6- Na barra temporal do AnimationPlayer, selecione o marcador de 1 seg, fazendo com que uma linha azul fique sob a marca de um segundo:



7- Modifique o valor da propriedade “frame” do AnimatedSprite para “1” e pressione novamente a chave do lado direito da propriedade “frame”. Repita este processo para a marca de 2 segundos com o valor da propriedade “frame” em 0;

8- Selecione a propriedade autoplay da animação ou adicione o script abaixo no RigidBody2D:



*extends RigidBody2D*

*func \_ready():*

*get\_node("AnimationPlayer").play("animation1")*

9- Selecione o botão de loop na barra de tarefas de baixo do AnimationPlayer para a animação criada;



10- Execute o projeto (salve a Scene antes) e veja o resultado;

11- Mova o StaticBody2D para a parte mais baixa da Scene;

12- Adicione uma Camera2D ao RigidBody2D e configure a propriedade “current” para “true”;

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

**Atividade 3 – Controle suave do RigidBody2D;**

1- Continuando a partir do projeto anterior, adicione o script abaixo ao RigidBody2D:

*extends RigidBody2D*

*export var player\_speed = 200*

*export var acceleration = 5*

*export var extra\_gravity = 400*

*var current\_speed = Vector2(0,0)*

*func \_ready():*

*set\_physics\_process(true)*

*set\_applied\_force(Vector2(0,extra\_gravity))*

*func move(speed, acc, delta):*

*current\_speed.x =* *lerp(current\_speed.x , speed, acc \* delta)*

*set\_linear\_velocity(Vector2(current\_speed.x,get\_linear\_velocity().y))*

*func \_physics\_process(delta):*

*if (Input.is\_key\_pressed(KEY\_LEFT)):*

*move(-player\_speed, acceleration, delta)*

*elif (Input.is\_key\_pressed(KEY\_RIGHT)):*

*move(player\_speed, acceleration, delta)*

*else:*

*move(0, acceleration, delta)  
  
2*- Execute o projeto (salve a Scene antes) e veja o resultado;

3- Configure as propriedades player\_speed, acceleration e extra\_gravity exportadas para o inspector do RigidBody2D conforme achar devido;

4- Configure as propriedades friction do RigidBody2D e StaticBody2D para 0;

*5*- Execute o projeto (salve a Scene antes) e veja o resultado;

**Atividade 4 – Colisões com RayCast2D;**

1- Continuando a partir do projeto anterior, adicione um RayCast2D ao RigidBody2D;

2- Configure a propriedade “enabled” do RayCast2D para “true”;

3- Substitua o script do RigidBody2D pelo código abaixo:

*extends RigidBody2D  
  
export var player\_speed = 200  
export var acceleration = 5  
export var extra\_gravity = 400  
export var jumpforce = 200  
  
var current\_speed = Vector2(0,0)  
var raycast\_down = null  
  
func \_ready():  
 raycast\_down = get\_node("RayCast2D")  
 raycast\_down.add\_exception(self)  
 set\_ physics\_process(true)  
 set\_applied\_force(Vector2(0,extra\_gravity))  
  
func is\_on\_ground():  
 if raycast\_down.is\_colliding():  
 return true  
 else:  
 return false  
  
func move(speed, acc, delta):  
 current\_speed.x = lerp(current\_speed.x , speed, acc \* delta)  
 set\_linear\_velocity(Vector2(current\_speed.x,get\_linear\_velocity().y))  
  
func \_physics\_process(delta):  
 if (Input.is\_key\_pressed(KEY\_LEFT)):   
 move(-player\_speed, acceleration, delta)  
 elif (Input.is\_key\_pressed(KEY\_RIGHT)):   
 move(player\_speed, acceleration, delta)  
 else:  
 move(0, acceleration, delta)  
 if is\_on\_ground():  
 if (Input.is\_key\_pressed(KEY\_SPACE)):   
 set\_axis\_velocity(Vector2(0,-jumpforce))*4- Execute o projeto (salve a Scene antes) e veja o resultado;

5- Modifique o código de modo a impedir que o jogador possa controlar o RigidBody2D quando este estiver no ar sem contato com o solo;

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

**Atividade 5 – Meu Primeiro Jogo Estilo Plataforma;**

Complemente o projeto com mais plataformas a serem atravessadas pelo jogador. Coloque inicialmente plataformas em posições fixas, para em seguida trabalhar com a geração de plataformas dinâmicas. Procure acrescentar itens a serem coletados em cada plataforma. Ganha o jogo quem chegar na plataforma final. Aplique um HUD de tempo para avaliar o desempenho do jogador. Se o jogador cair da plataforma este deve retornar para a primeira plataforma do jogo. Uma mensagem deve ser exibida para o usuário quando ele concluir o jogo informando o tempo total consumido. Se o jogador demorar mais do que 5 min, o jogo deve informar “Fim de Jogo”.