

UEFS - Universidade Estadual de Feira de Santana
Jogos Digitais – prof. D. Sc. Victor Sarinho

Aula Prática – Introdução ao Godot (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_fixed_process(true)
```

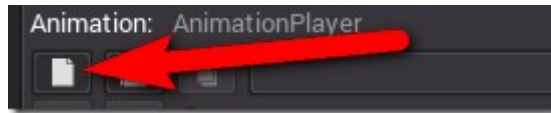
```
func _fixed_process(delta):  
    tempElapsed = tempElapsed + delta
```

```
    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;

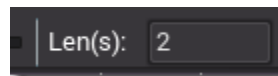
- 1- Continuando a partir do projeto anterior, selecione a propriedade “script” do AnimatedSprite e clique na opção “clear”;
- 2- Adicione um AnimationPlayer ao RigidBody2D;
- 3- Crie uma nova animação no AnimationPlayer com o nome “animation1”;



4- Selecione o ícone de edição do AnimationPlayer:

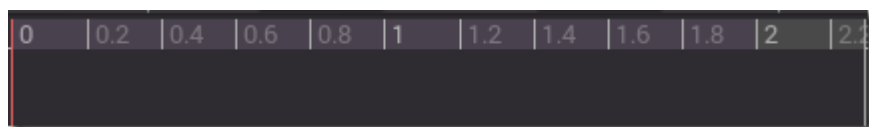


5- Configure a animação para 2 segundos:



6- 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;

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



8- 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;

9- Selecione a propriedade autoplay da animação ou adicione o script abaixo no Rigidbody2D:



extends Rigidbody2D

```
func _ready():
    get_node("AnimationPlayer").play("animation1")
```

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

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

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

13- Adicione uma Camera2D ao Rigidbody2D e configure a propriedade “current” para “true”;

14- 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_fixed_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 _fixed_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_fixed_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 _fixed_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”.