

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

**Aula Prática – Introdução ao Godot (2ª parte)**

**Atividade 1 – Instanciando Objetos Dinamicamente;**

- 1- Crie um novo projeto;
- 2- Adicione um Panel na Scene inicial;
- 3- Adicione um script com o seguinte código no Panel:

```
extends Panel
```

```
var current_index = 0  
var last_index = 0  
var labels = []
```

```
func _ready():  
    self.set_process(true)
```

```
func _process(delta):  
    if(Input.is_mouse_button_pressed(BUTTON_LEFT)):  
        var l = Label.new()  
        l.set_text("Label " + str(current_index))  
        add_child(l) # add it as a child of this node  
        labels.append(l)  
        current_index += 1  
        l.set_pos(get_global_mouse_pos())  
  
    if(Input.is_mouse_button_pressed(BUTTON_RIGHT)):  
        if(last_index < current_index):  
            labels[last_index].queue_free()  
            last_index += 1
```

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

**Atividade 2 – Arrastar e Soltar Imagens;**

- 1- Crie um novo projeto;
- 2- Adicione um Panel e um TextureFrame na Scene inicial;
- 3- Configure o TextureFrame com a imagem desejada e adicione o script abaixo:

```
extends TextureFrame
```

```
var mouse_enter = false  
var object_dragged = false  
var pos = get_global_mouse_pos()
```

```
func _ready():
    self.connect("mouse_enter", self, "_on_mouse_enter")
    self.connect("mouse_exit", self, "_on_mouse_exit")
    set_fixed_process(true)
```

```
func _fixed_process(delta):
    pos = get_global_mouse_pos()
    if mouse_enter == true:
        if Input.is_mouse_button_pressed(BUTTON_LEFT):
            set_global_pos(pos)
            object_dragged = true
        else:
            object_dragged = false
```

```
func _on_mouse_enter():
    mouse_enter = true
```

```
func _on_mouse_exit():
    if object_dragged == false:
        mouse_enter = false
```

- 4- Configure a propriedade Ignore Mouse para False;
- 5- Execute o projeto (salve a Scene antes) e veja o resultado.

### Atividade 3 – Usando Dialogs;

- 1- Crie um novo projeto;
- 2- Adicione um Panel e um Label na Scene inicial;
- 3- Adicione um script com o seguinte código no Panel:

```
extends Panel
```

```
var screen_size
var cd
var current_index = 0
```

```
func _ready():
    self.set_process(true)
    screen_size = get_viewport_rect().size
    cd = get_node("ConfirmationDialog")
```

```
func _process(delta):
    if(Input.is_key_pressed(KEY_SPACE)):
        var new_cd = cd.duplicate()
        new_cd.set_title("Confirm. " + str(current_index))
        add_child(new_cd)
        current_index += 1
        new_cd.popup_centered(Vector2(randf()*screen_size.x, randf()*screen_size.y))
```

- 4- Adicione um ConfirmationDialog na Scene inicial;
- 5- Adicione um script com o seguinte código no ConfirmationDialog:

*extends ConfirmationDialog*

```
func _ready():  
    self.connect("confirmed",self,"_on_confirmed_pressed")  
  
func _on_confirmed_pressed():  
    get_node("/root/Panel/Label").set_text("Confirmed "+self.get_title())
```

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

#### **Atividade 4 – Dados Globais & Múltiplas Scenes;**

- 1- Crie um novo projeto;
- 2- Abra a tela de scripts e adicione um script ao projeto denominado global.gd;
- 3- Adicione o seguinte código ao script global.gd:

*extends Node*

```
var current_scene = null
```

```
func _ready():  
    get_tree().set_auto_accept_quit(false)  
    var root = get_tree().get_root()  
    current_scene = root.get_child( root.get_child_count() -1 )  
  
func goto_scene(path):  
    # This function will usually be called from a signal callback,  
    # or some other function from the running scene.  
    # Deleting the current scene at this point might be  
    # a bad idea, because it may be inside of a callback or function of it.  
    # The worst case will be a crash or unexpected behavior.  
  
    # The way around this is deferring the load to a later time, when  
    # it is ensured that no code from the current scene is running:
```

```
call_deferred("_deferred_goto_scene",path)
```

```
func _deferred_goto_scene(path):  
    # Immediately free the current scene,  
    # there is no risk here.  
    current_scene.free()  
  
    # Load new scene  
    var s = ResourceLoader.load(path)  
  
    # Instance the new scene  
    current_scene = s.instance()  
  
    # Add it to the active scene, as child of root  
    get_tree().get_root().add_child(current_scene)
```

*# optional, to make it compatible with the SceneTree.change\_scene() API*  
*get\_tree().set\_current\_scene(current\_scene)*

- 4- Adicione um Control na Scene inicial e altere seu nome para Tela1 (tela inicial deve ser salva com o nome de tela1.tscn);
- 5- Adicione um Label com o texto “Tela 1” e um Button com o texto “Mudar Tela” na Tela 1;
- 6- Adicione um Script em Tela1, salve como tela1.gd e aplique o seguinte código abaixo no mesmo:

*extends Control*

```
func _ready():  
    get_node("Button").connect("pressed",self,"_on_Button_pressed")
```

```
func _on_Button_pressed():  
    get_node("/root/global").goto_scene("res://tela2.tscn")
```

- 7- Adicione uma nova Scene ao projeto;
- 8- Adicione um Control na Scene inicial e altere seu nome para Tela2 (esta tela deve ser salva com o nome de tela2.tscn);
- 9- Adicione um Label com o texto “Tela 2” e um Button com o texto “Mudar Tela” na Tela 2;
- 10- Adicione um Script em Tela2, salve como tela2.gd e aplique o seguinte código abaixo no mesmo:

*extends Control*

```
func _ready():  
    get_node("Button").connect("pressed",self,"_on_Button_pressed")
```

```
func _on_Button_pressed():  
    get_node("/root/global").goto_scene("res://tela1.tscn")
```

- 11- Selecione /Scene/Project Settings no menu inicial;
- 12- Selecione a aba AutoLoad e adicione o script global.gd como um recurso global ao projeto;
- 13- Execute o projeto e veja o resultado;
- 14- Adicione uma opção de encerrar o jogo na Tela1;
- 15- Execute o projeto e veja o resultado.

## **Atividade 5 – Meu Primeiro Jogo**

Construa um jogo em que Sprites são criados dinamicamente na tela em posições aleatórias. Cada vez que o usuário clicar com o botão esquerdo do mouse em cima de um destes Sprites o jogador ganha um ponto, o Sprite é destruído e um novo Sprite é criado em uma nova posição. Cada partida terá duração de 1 minuto. Um HUD deve ser criado, apresentando os pontos do jogador e o tempo que falta para terminar a partida. Ao final uma mensagem deve ser apresentada informando a pontuação final e perguntando se o jogador quer jogar novamente. Se sim uma nova partida deve ser iniciada, caso contrário o jogo deve ser encerrado.

## **Atividade 6 – Um Menu Inicial para Meu Primeiro Jogo**

Acrescente uma Scene inicial ao jogo contendo o menu inicial do jogo. Ele deve conter a opção de iniciar a partida, a opção Sobre e a opção de encerrar o jogo. Ao final de uma partida o jogador deve retornar ao Menu Inicial construído.

### **Atividade 7 – Uma variante para o Meu Primeiro Jogo**

Acrescente um jogo de arrastar-e-soltar no Meu Primeiro Jogo. Coloque um sprite no meio da tela cada vez que o jogador arrastar um Sprite ao centro da tela ele ganha pontos. Mesma dinâmica de tempo e de encerramento do jogo inicial.

*Sugestão:* use “`Rect2(centro_pos, centro_size).encloses(Rect2(sprite_pos, sprite_size))`” para verificar se houve colisão entre o Sprite arrastado e a área central da tela.