**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 (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\_position(get\_viewport().get\_mouse\_position())*

*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 TextureRect na Scene inicial;

3- Configure o TextureRect com a imagem desejada (propriedade *Texture*) e adicione o script abaixo:

*extends TextureRect*

*var mouse\_enter = false*

*var object\_dragged = false*

*var pos# = get\_viewport().get\_mouse\_position()*

*func \_ready():*

*self.connect("mouse\_entered", self, "\_on\_mouse\_enter")*

*self.connect("mouse\_exited", self, "\_on\_mouse\_exit")*

*set\_process(true)*

*func \_process(delta):*

*pos = get\_viewport().get\_mouse\_position()*

*if mouse\_enter == true:*

*if Input.is\_mouse\_button\_pressed(BUTTON\_LEFT):*

*set\_position(pos)*

*object\_dragged = true*

*print("arrastando: ",pos)*

*else:*

*object\_dragged = false*

*func \_on\_mouse\_enter():*

*mouse\_enter = true*

*print("entrou")*

*func \_on\_mouse\_exit():*

*print("exit")*

*if object\_dragged == false:*

*mouse\_enter = false*

4- 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 joga 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.