

Projeto A3 - Computação Gráfica e Realidade Virtual

Jogo Carmesim Mascarado

Tema:

O jogo possui o tema de Dungeons e ruínas com um aspecto medieval e dark fantasy.

Objetivo:

O jogador sobreviver o máximo de salas (Dungeons) e bosses (chefes) possíveis.

Mecânicas de Jogo:

O jogo possui as mecânicas de RPG com uma mistura de Roguelikes, cada vez que o personagem principal sobe de level, ele ganha acesso a melhorias de habilidades e vida.

Design:

- Gráficos: Possui gráficos em pixel, em 16 bits.
- Sons e Músicas: As músicas temas tem o toque de músicas eletrônicas. Alguns sons de personagem e mobs.
- Interface de Usuário: O jogo possui um menu com as opções de Novo Jogo, controles, som e para sair do game. Em game no pause tem como você começar outra partida, voltar ao menu, sair do game e resumir.

Implementação:

- Engine/Plataforma: Godot
- Linguagens de Programação: GDscript, linguagem com base do C#.
- Bibliotecas e Ferramentas: No geral utilizamos as próprias ferramentas do Godot, Asesprite.

Documentação:

Fizemos um jogo inspirado em alguns RPGs, Rogue-Likes e Souls-like. O objetivo principal do personagem é passar as salas enquanto sobe de level e melhora suas skills. No final ele vai enfrentar um boss e reiniciar esse processo “procedural”. Utilizamos a linguagem GDscript. Pegamos alguns assets e sons da internet sem nenhum direito autoral. Programamos basicamente tudo.

Cronograma de Desenvolvimento:

- [Semana 1]: Começamos a fazer a parte dos sprites que vão ser utilizados (mobs, objetos, personagens. Física e colisão, além de começar a fazer a parte do mapa;
- [Semana 2]: Começamos a fazer parte da interação entre os mobs. Comportamento, e ações em game. Morte e sons tanto do personagem principal quanto dos mobs e boss;
- [Semana 3]: Focamos nessa semana para fazer praticamente todas as interações que o personagem podia ter, como a parte da luta, morte, corrida, habilidades especiais;
- [Semana 4]: Começamos a mexer na transição do mapa. Onde o personagem corre em um corredor com vários mobs e obstáculos;
- [Semana 5]: Mexemos em alguns sprites, sons e colisões de alguns obstáculos, junto com a parte das habilidades do personagem;
- [Semana 6]: Toques finais na parte do pause, menu principal e suas opções. E finalizar as interações do boss.

Referências:

- The Binding of Isaac;
- Dark Souls;
- Chat GPT;
- [Tutoriais do youtube.](#)

Problemas Encontrados e Soluções:

- Tivemos alguns problemas relacionados à parte física do personagem. Ele estava caindo do cenário. Nesse problema conseguimos arrumar fazendo alguns ajustes na colisão do próprio mapa.
- O Boss estava sumindo dependendo de alguma circunstância. Arrumamos em parte esse bug, pois não conseguimos visualizar onde estava o problema em si. De qualquer forma é um bug raro e não afeta a jogabilidade no geral.
- Problemas na hora de linkar a transição do cenário. Estava dando algum problema na hora de passar para a próxima fase. Conseguimos arrumar copiando os dois primeiros levels, e fazendo eles se repetirem.

Lições Aprendidas:

Aprendemos que é bem mais complexo fazer um jogo do que pensávamos. A parte que deixou isso clara, foi na parte de fazer a programação no geral, tem muita coisa que tem que ser programa, como a colisão, animação, sprites, cenário, mobs, etc. É extremamente importante deixar uma copia pré-salva no GitHub.

Conclusão:

Conseguimos implementar de certa forma a nossa ideia inicial, tivemos alguns contratemplos, e bugs que no final, não conseguimos arrumar todos(Não são bugs críticos). O jogo ficou bem completo, o personagem termina uma sala, upa de level, anda para outra sala, derrota o boss e esse ciclo vai se repetindo infinitamente.

Apêndice:

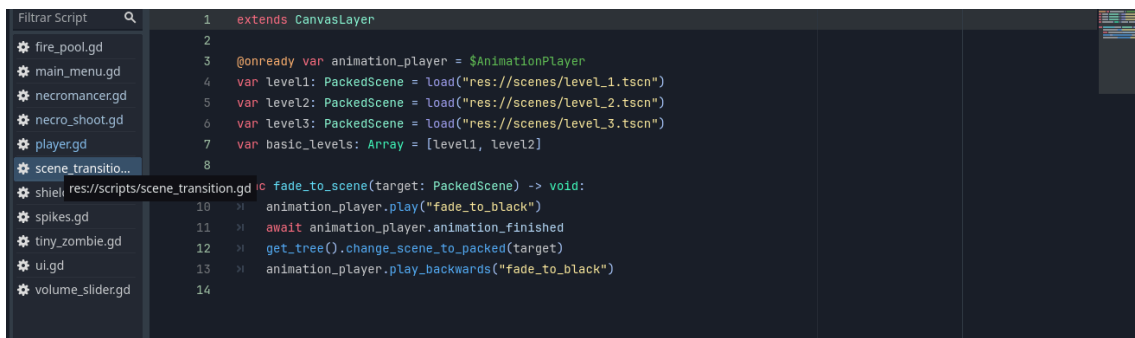
Script dos sons de mobs e personagem:

```
1 extends Enemy
2
3 @onready var burn_audio: AudioStreamPlayer = $BurnAudio
4 @onready var hurtbox: Area2D = $Hurtbox
5
6 func _process(_delta) -> void:
7     for body in hurtbox.get_overlapping_bodies():
8         if "hit" in body and body.is_in_group("player"):
9             var direction_to_player = (player.global_position - global_position).normalized()
10            body.hit(damage, direction_to_player, self)
11        if "hit" in body and body.is_in_group("enemy"):
12            body.hit(damage, self)
13
14
15 func hit(_amount, _attacker: Node = null):
16     pass
17
18 func _on_hurtbox_body_entered(_body):
19     burn_audio.play()
20
```

Menu principal e suas opções:

```
main_menu
Arquivo Editar Pesquisar Ir Para Depuração Documentação Online Pesquisar Ajuda
Filtrar Script
fire_pool.gd
main_menu.gd
necromancer.gd
necro_shoot.gd
player.gd
scene_transitio...
shield.gd
spikes.gd
tiny_zombie.gd
ui.gd
volume_slider.gd
main_menu.gd
Filtrar Métodos
_ready
_on_new_game_pr...
_on_quit_pressed
hide_all
_on_options_pressed
_on_controls_press...
3 @onready var ui_sounds: AudioStreamPlayer = $UISounds
4 @onready var options: Control = $Options
5 @onready var controls: Control = $Controls
6
7
8 # Called when the node enters the scene tree for the first time.
9 func _ready() -> void:
10     UI.hide()
11     hide_all()
12
13
14 func _on_new_game_pressed() -> void:
15     ui_sounds.play()
16     SceneTransition.fade_to_scene(SceneTransition.basic_levels.pick_random())
17
18
19 func _on_quit_pressed() -> void:
20     ui_sounds.play()
21     get_tree().quit()
22
23 func hide_all():
24     controls.hide()
25     options.hide()
26
27 func _on_options_pressed() -> void:
28     hide_all()
29     ui_sounds.play()
30     options.show()
31
32 func _on_controls_pressed() -> void:
33     hide_all()
34     ui_sounds.play()
35     controls.show()
1 : 10 | Abas
```

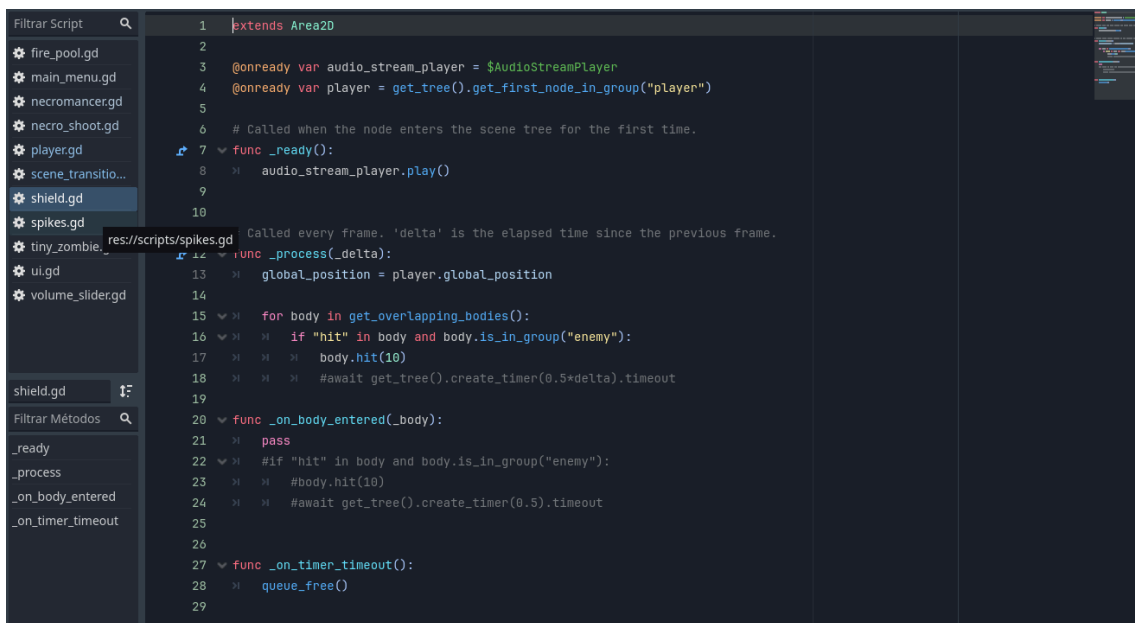
Transição de Áudio:



The screenshot shows the Godot 4.0 script editor with the 'scene_transition.gd' script selected in the left sidebar. The script is a GDScript that extends 'CanvasLayer'. It defines three levels as 'PackedScene' resources and an array of basic levels. A 'fade_to_scene' function is defined, which takes a 'target' scene and calls 'animation_player.play("fade_to_black")', then 'await animation_player.animation_finished', and finally 'get_tree().change_scene_to_packed(target)'. The 'fade_to_scene' function is called from the 'animation_player.play_backwards("fade_to_black")' line.

```
1 extends CanvasLayer
2
3 @onready var animation_player = $AnimationPlayer
4 var level1: PackedScene = load("res://scenes/level_1.tscn")
5 var level2: PackedScene = load("res://scenes/level_2.tscn")
6 var level3: PackedScene = load("res://scenes/level_3.tscn")
7 var basic_levels: Array = [level1, level2]
8
9
10 func fade_to_scene(target: PackedScene) -> void:
11     animation_player.play("fade_to_black")
12     await animation_player.animation_finished
13     get_tree().change_scene_to_packed(target)
14     animation_player.play_backwards("fade_to_black")
```

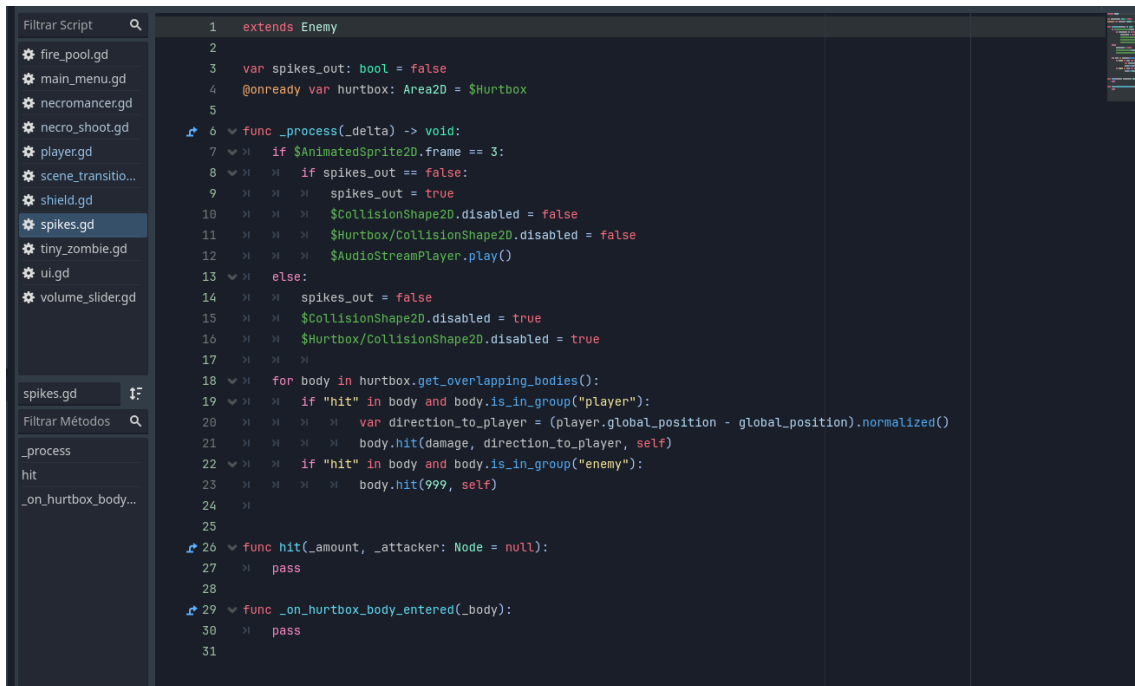
Shield do personagem:



The screenshot shows the Godot 4.0 script editor with the 'shield.gd' script selected in the left sidebar. The script is a GDScript that extends 'Area2D'. It defines an 'audio_stream_player' and a 'player' node. The '_ready()' function calls 'audio_stream_player.play()'. The '_process(_delta)' function is called every frame and checks for overlapping bodies. If a body is found, it checks if it is in the 'enemy' group and calls 'body.hit(10)'. It then calls 'await get_tree().create_timer(0.5*delta).timeout'. The '_on_body_entered(_body)' function is called when a body enters the area and calls 'pass'. The '_on_timer_timeout()' function is called when the timer expires and calls 'queue_free()'.

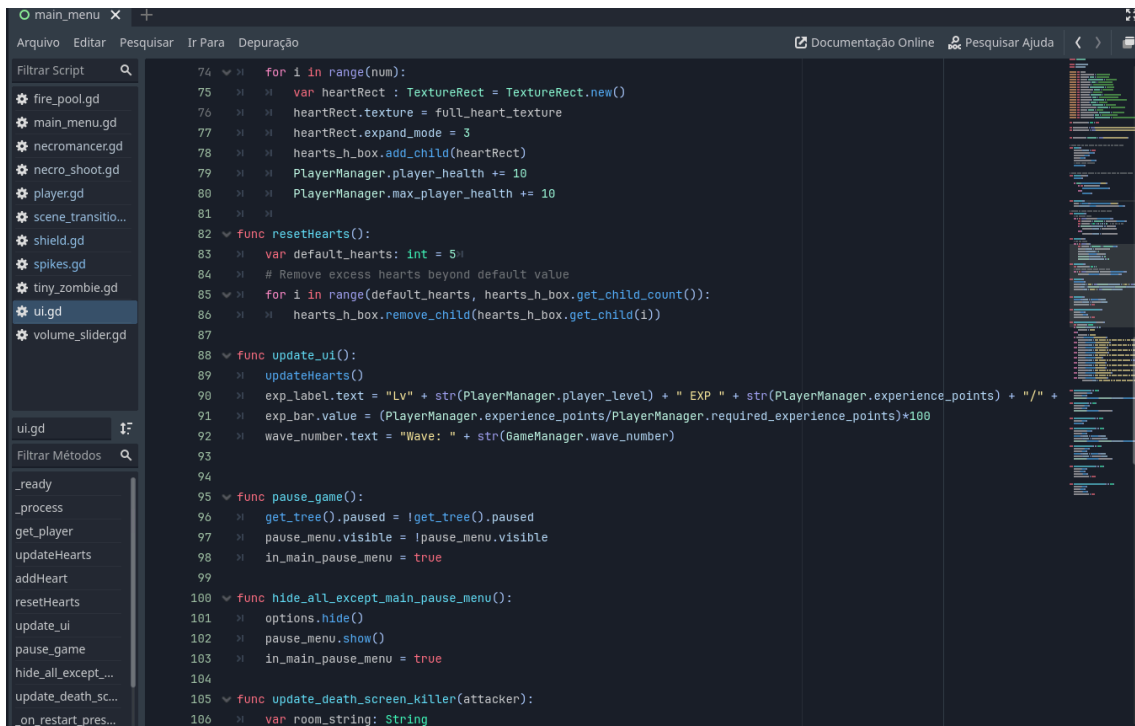
```
1 extends Area2D
2
3 @onready var audio_stream_player = $AudioStreamPlayer
4 @onready var player = get_tree().get_first_node_in_group("player")
5
6 # Called when the node enters the scene tree for the first time.
7 func _ready():
8     audio_stream_player.play()
9
10
11 Called every frame. 'delta' is the elapsed time since the previous frame.
12 func _process(_delta):
13     global_position = player.global_position
14
15     for body in get_overlapping_bodies():
16         if "hit" in body and body.is_in_group("enemy"):
17             body.hit(10)
18             #await get_tree().create_timer(0.5*delta).timeout
19
20 func _on_body_entered(_body):
21     pass
22     #if "hit" in body and body.is_in_group("enemy"):
23         #body.hit(10)
24         #await get_tree().create_timer(0.5).timeout
25
26 func _on_timer_timeout():
27     queue_free()
```

Colisão dos Espinhos:



```
1 extends Enemy
2
3 var spikes_out: bool = false
4 @onready var hurtbox: Area2D = $Hurtbox
5
6 func _process(_delta) -> void:
7     if $AnimatedSprite2D.frame == 3:
8         if spikes_out == false:
9             spikes_out = true
10            $CollisionShape2D.disabled = false
11            $Hurtbox/CollisionShape2D.disabled = false
12            $AudioStreamPlayer.play()
13        else:
14            spikes_out = false
15            $CollisionShape2D.disabled = true
16            $Hurtbox/CollisionShape2D.disabled = true
17
18    for body in hurtbox.get_overlapping_bodies():
19        if "hit" in body and body.is_in_group("player"):
20            var direction_to_player = (player.global_position - global_position).normalized()
21            body.hit(damage, direction_to_player, self)
22        if "hit" in body and body.is_in_group("enemy"):
23            body.hit(999, self)
24
25
26 func hit(_amount, _attacker: Node = null):
27     pass
28
29 func _on_hurtbox_body_entered(_body):
30     pass
31
```

Opções do pause em game:



```
74 for i in range(num):
75     var heartRect: TextureRect = TextureRect.new()
76     heartRect.texture = full_heart_texture
77     heartRect.expand_mode = 3
78     hearts_h_box.add_child(heartRect)
79     PlayerManager.player_health += 10
80     PlayerManager.max_player_health += 10
81
82 func resetHearts():
83     var default_hearts: int = 5
84     # Remove excess hearts beyond default value
85     for i in range(default_hearts, hearts_h_box.get_child_count()):
86         hearts_h_box.remove_child(hearts_h_box.get_child(i))
87
88 func update_ui():
89     updateHearts()
90     exp_label.text = "Lv" + str(PlayerManager.player_level) + " EXP " + str(PlayerManager.experience_points) + "/" +
91     exp_bar.value = (PlayerManager.experience_points/PlayerManager.required_experience_points)*100
92     wave_number.text = "Wave: " + str(GameManager.wave_number)
93
94
95 func pause_game():
96     get_tree().paused = !get_tree().paused
97     pause_menu.visible = !pause_menu.visible
98     in_main_pause_menu = true
99
100 func hide_all_except_main_pause_menu():
101     options.hide()
102     pause_menu.show()
103     in_main_pause_menu = true
104
105 func update_death_screen_killer(attacker):
106     var room_string: String
```

Áudio do Jogo:

```
1 extends HSlider
2
3
4 @export var bus_name: String
5 var bus_index: int
6
7 func _ready() -> void:
8     bus_index = AudioServer.get_bus_index(bus_name)
9
10     # for initializing with CURRENT volume settings
11     value = db_to_linear(AudioServer.get_bus_volume_db(bus_index))
12
13 func _on_value_changed(amount: float) -> void:
14     AudioServer.set_bus_volume_db(bus_index, linear_to_db(amount))
15
```

Variáveis do Boss:

```
1 extends CharacterBody2D
2
3 signal necromancer_died
4 signal necromancer_summoned_wave
5
6 @export var stats: CharacterStats
7
8 @onready var health: float = stats.starting_health
9 @onready var max_health: float = stats.max_health
10 @onready var damage: float = stats.damage
11 @onready var speed: float = stats.move_speed
12 @onready var knockback_strength: float = stats.knockback_strength
13 @onready var exp_dropped: float = stats.exp_dropped
14 @onready var hit_sound = stats.hit_sound
15 @onready var hit_sound_pitch = stats.hit_sound_pitch
16 @onready var death_sound = stats.death_sound
17 @onready var death_sound_pitch = stats.death_sound_pitch
18
19 var direction: Vector2 = Vector2.ZERO
20 var knockback: Vector2 = Vector2.ZERO
21 var invuln: bool = false
22 var casting: bool = false
23 var introducing: bool = false
24 var summoned_wave: bool = false
25 var blood_splatter_scene: PackedScene = preload("res://scenes/blood_splatter.tscn")
26 var blood_hit_scene: PackedScene = preload("res://scenes/blood_hit.tscn")
27 var damage_numbers_scene: PackedScene = preload("res://scenes/damage_numbers.tscn")
28 var skull_scene: PackedScene = preload("res://scenes/red_skull.tscn")
29 var lightning_scene: PackedScene = preload("res://scenes/unholy_lightning.tscn")
30
31 @onready var player = get_tree().get_first_node_in_group("player")
32 @onready var animated_sprite_2d = $AnimatedSprite2D
33 @onready var sprite_2d = $Sprite2D
```

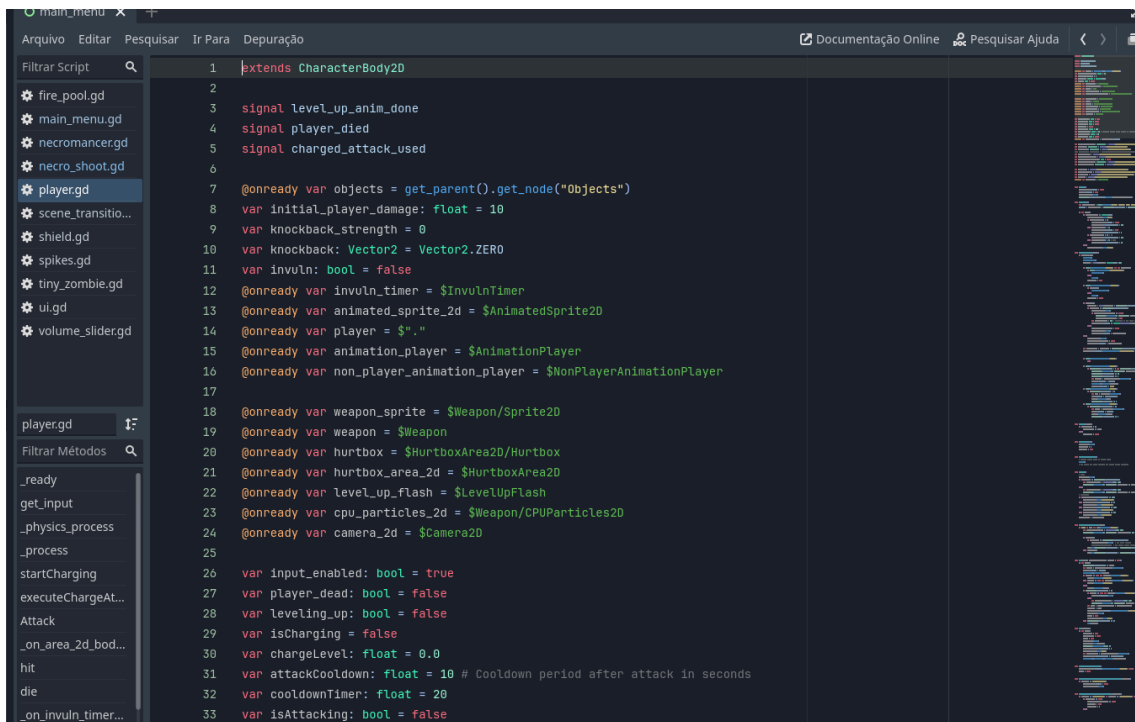
```
Arquivo  Editar  Pesquisar  Ir Para  Depuração  Documentação Online  Pesquisar Ajuda  < >
Filtrar Script  🔍
fire_pool.gd
main_menu.gd
necromancer.gd
necro_shoot.gd
player.gd
scene_transitio...
shield.gd
spikes.gd
tiny_zombie.gd
ui.gd
volume_slider.gd

necromancer.gd  🔍
Filtrar Métodos  🔍
_ready
_physics_process
introduce
summon_wave
hit
_on_invuln_timer_ti...
spawn_blood_hit
die
_on_hurtbox_body...
summon_skull_on_...
summon_lightning...

30
31 @onready var player = get_tree().get_first_node_in_group("player")
32 @onready var animated_sprite_2d = $AnimatedSprite2D
33 @onready var sprite_2d: Sprite2D = $Sprite2D
34 @onready var spell_effect: Sprite2D = $SpellEffect
35 @onready var invuln_timer = $InvulnTimer
36 @onready var dmg_num_location = $DmgNumLocation
37 @onready var roar_audio: AudioStreamPlayer = $RoarAudio
38 @onready var die_audio: AudioStreamPlayer = $DieAudio
39 @onready var skull_marker_node: Node2D = $Sprite2D/SkullMarkerNode/StaffSkullSpawn
40 @onready var staff_skull_spawn_marker: Marker2D = $Sprite2D/SkullMarkerNode/StaffSkullSpawn
41 @onready var necro_anim_player: AnimationPlayer = $NecromancerAnimationPlayer
42 @onready var objects = get_parent().get_parent().get_node("Objects")
43 @onready var necro_shoot_skull_state_node: Node = $StateMachine/NecroShootSkull
44 @onready var necro_lightning_node: Node = $StateMachine/NecroLightning
45 @onready var necro_death_node: Node = $StateMachine/NecroDeath
46 @onready var necro_intro_node: Node = $StateMachine/NecroIntro
47 @onready var necro_summon_node: Node = $StateMachine/NecroSummon
48 @onready var state_machine: Node = $StateMachine
49
50 func _ready():
51     necro_shoot_skull_state_node.skull_casted.connect(summon_skull_on_necro)
52     necro_lightning_node.lightning_casted.connect(summon_lightning_on_necro)
53     necro_death_node.necromancer_died.connect(die)
54     necro_intro_node.intro_started.connect(introduce)
55     necro_summon_node.in_position_to_summon.connect(summon_wave)
56     spell_effect.hide()
57
58 func _physics_process(_delta):
59     move_and_slide()
60     knockback = lerp(knockback, Vector2.ZERO, 0.1)
61
62     if not invuln and not casting and not introducing:
63         if velocity.length() > 0:
```

```
2
3 signal skull_casted
4
5 @onready var enemy: CharacterBody2D = get_parent().get_parent()
6 var player: CharacterBody2D
7 var move_direction: Vector2
8 var casting: bool = false
9 var count: int = 0
10
11 func Enter():
12     player = get_tree().get_first_node_in_group("player")
13     count = 0
14
15
16 func Physics_Update(_delta: float):
17     if enemy and not casting:
18         enemy.velocity = Vector2.ZERO
19         var distance = (player.global_position - enemy.global_position).length()
20         if distance < 40:
21             Transitioned.emit(self, "NecroLightning")
22         elif count < 50:
23             summon_skull()
24         else:
25             Transitioned.emit(self, "EnemyChase")
26
27
28 func summon_skull():
29     casting = true
30     skull_casted.emit()
31     await get_tree().create_timer(0.1).timeout # cooldown
32     count += 1
33     casting = false
34
```

Personagem principal:



```
1 extends CharacterBody2D
2
3 signal level_up_anim_done
4 signal player_died
5 signal charged_attack_used
6
7 @onready var objects = get_parent().get_node("Objects")
8 var initial_player_damage: float = 10
9 var knockback_strength = 0
10 var knockback: Vector2 = Vector2.ZERO
11 var invuln: bool = false
12 @onready var invuln_timer = $InvulnTimer
13 @onready var animated_sprite_2d = $AnimatedSprite2D
14 @onready var player = $".."
15 @onready var animation_player = $AnimationPlayer
16 @onready var non_player_animation_player = $NonPlayerAnimationPlayer
17
18 @onready var weapon_sprite = $Weapon/Sprite2D
19 @onready var weapon = $Weapon
20 @onready var hurtbox = $HurtboxArea2D/Hurtbox
21 @onready var hurtbox_area_2d = $HurtboxArea2D
22 @onready var level_up_flash = $LevelUpFlash
23 @onready var cpu_particles_2d = $Weapon/CPUParticles2D
24 @onready var camera_2d = $Camera2D
25
26 var input_enabled: bool = true
27 var player_dead: bool = false
28 var leveling_up: bool = false
29 var isCharging = false
30 var chargeLevel: float = 0.0
31 var attackCooldown: float = 10 # Cooldown period after attack in seconds
32 var cooldownTimer: float = 20
33 var isAttacking: bool = false
```