# Guia de Criação do Jogo Space Shooter – Parte 1

## Criando o Projeto e a Cena Principal (Mundo) *🡺 Main = Cena principal*

### Objetivo desta parte

• Criar o projeto no Godot.  
• Montar a cena principal chamada Mundo.  
• Colocar o script completo do Mundo.  
• Entender para que serve cada parte do código.

### 1. Criar o Projeto

1. Abra o Godot.  
2. Clique em New Project.  
3. Dê o nome: SpaceShooter.  
4. Escolha uma pasta para salvar.  
5. Clique em Create & Edit.  
  
👉 Agora temos o projeto vazio.

### 2. Criar as pastas do Projeto

1. Com o botão direiro, crie as pastas Cenas, Scripts e Assets a dentro da pasta principal res://  
2. Arraste os Sprites e o fundo de tela para dentro da pasta Assets

### 3. Criar a Cena Mundo

1. Clique em + para adicionar um nó.  
2. Escolha o tipo Node2D.  
3. Renomeie para Mundo (aperte F2 para mudar o nome).  
4. Salve a cena como mundo.tscn.

### 4. Adicionar o Script do Mundo

1. Selecione o nó Mundo.  
2. Clique em Attach Script.  
3. Salve como mundo.gd.  
4. Cole exatamente este código:

extends Node2D  
  
@onready var cena\_asteroide = preload("res://Cenas/asteroide.tscn")  
@onready var cena\_inimigo = preload("res://Cenas/inimigo.tscn")  
  
var vidas := 3  
var score := 0  
  
func \_ready():  
 randomize()  
 $HUD.connect("start\_game", Callable(self, "iniciar\_jogo"))  
 $HUD.update\_lives(vidas)  
 $HUD.update\_score(score)  
  
func iniciar\_jogo():  
 vidas = 3  
 score = 0  
 $HUD.update\_lives(vidas)  
 $HUD.update\_score(score)  
   
 # Recria o jogador se ele foi destruído  
 if not has\_node("Jogador"): # se a cena não tem o “Jogador”  
 var cena\_jogador = preload("res://Cenas/jogador.tscn")  
 var j = cena\_jogador.instantiate()  
 j.position = Vector2(240, 400) # posição inicial  
 add\_child(j)  
   
 $Timer\_Asteroide.start()  
 $Timer\_Inimigo.start()  
  
func \_on\_timer\_asteroide\_timeout():  
 var a = cena\_asteroide.instantiate()  
 a.position = Vector2(randi\_range(30, 480), -20)  
 add\_child(a)  
  
func \_on\_timer\_inimigo\_timeout():  
 var i = cena\_inimigo.instantiate()  
 i.position = Vector2(randi\_range(30, 480), -20)  
 add\_child(i)  
  
func add\_score(valor: int) -> void:  
 score += valor  
 $HUD.update\_score(score)  
   
func perder\_vida(qtd := 1):  
 if vidas <=0:  
 return # já acabou, ignora chamadas extars  
   
 vidas -= qtd  
 $HUD.update\_lives(vidas)  
   
 if vidas <= 0:  
 game\_over()  
  
  
func game\_over():  
 # para os spawns  
 $Timer\_Asteroide.stop()  
 print("para o spawn 1")  
 $Timer\_Inimigo.stop()  
 print("para o spawn 2")  
  
 # Limpa inimigos e tiros (coloque tiros em um grupo "Tiros")  
 for n in get\_tree().get\_nodes\_in\_group("Inimigos"):  
 n.queue\_free()  
 for n in get\_tree().get\_nodes\_in\_group("Asteroides"):  
 n.queue\_free()  
 for n in get\_tree().get\_nodes\_in\_group("Tiros"):  
 n.queue\_free()  
  
 if has\_node("Jogador"):  
 $Jogador.queue\_free()  
   
 $HUD.show\_game\_over()

### 5. O que esse script faz

• vidas → começa com 3, é o “coração” do jogador.  
• score → começa com 0, é a pontuação.  
• \_ready() → prepara o HUD (vidas e score).  
• iniciar\_jogo() → reinicia tudo e cria o jogador na posição inicial.  
• \_on\_timer\_asteroide\_timeout() → cria asteroides no topo da tela.  
• \_on\_timer\_inimigo\_timeout() → cria inimigos no topo da tela.  
• add\_score() → aumenta a pontuação.  
• perder\_vida() → tira vidas quando o jogador leva dano.  
• game\_over() → para o jogo, limpa inimigos e tiros, apaga a nave e mostra “Game Over”.

👉 Agora temos o Mundo configurado com todas as regras principais.