

TECNOLOGIA E DESENVOLVIMENTO DE JOGOS -
NP2

Heaven Heights



Heaven Heights

HEAVEN HEIGHTS É UM JOGO DE ESTRATÉGIA E SIMULAÇÃO DO TIPO CITY BUILDING, ONDE O JOGADOR ATUA COMO UM PLANEJADOR E ADMINISTRADOR DE UM BAIRRO, INSPIRADO NA SITUAÇÃO FICTÍCIA DO DOCUMENTO 'O BAIRRO', DA RECEITO FEDERAL.



Gameplay - Ideia Principal

NA IDEIA PRINCIPAL DA GAMEPLAY DO HEAVEN HEIGHTS, O JOGADOR CONTROLA UM PERSONAGEM QUE O REPRESENTE DENTRO DO MUNDO DO JOGO, ANDANDO PELO BAIRRO, INTERAGINDO COM OS MORADORES (NPCS), E FAZENDO MISSÕES QUE RECOMPENSAM O JOGADOR COM DINHEIRO PARA GERENCIAR O DESENVOLVIMENTO E ECONOMIA DO BAIRRO, E A SATISFAÇÃO DOS MORADORES.

Desenvolvimento - Tecnologia

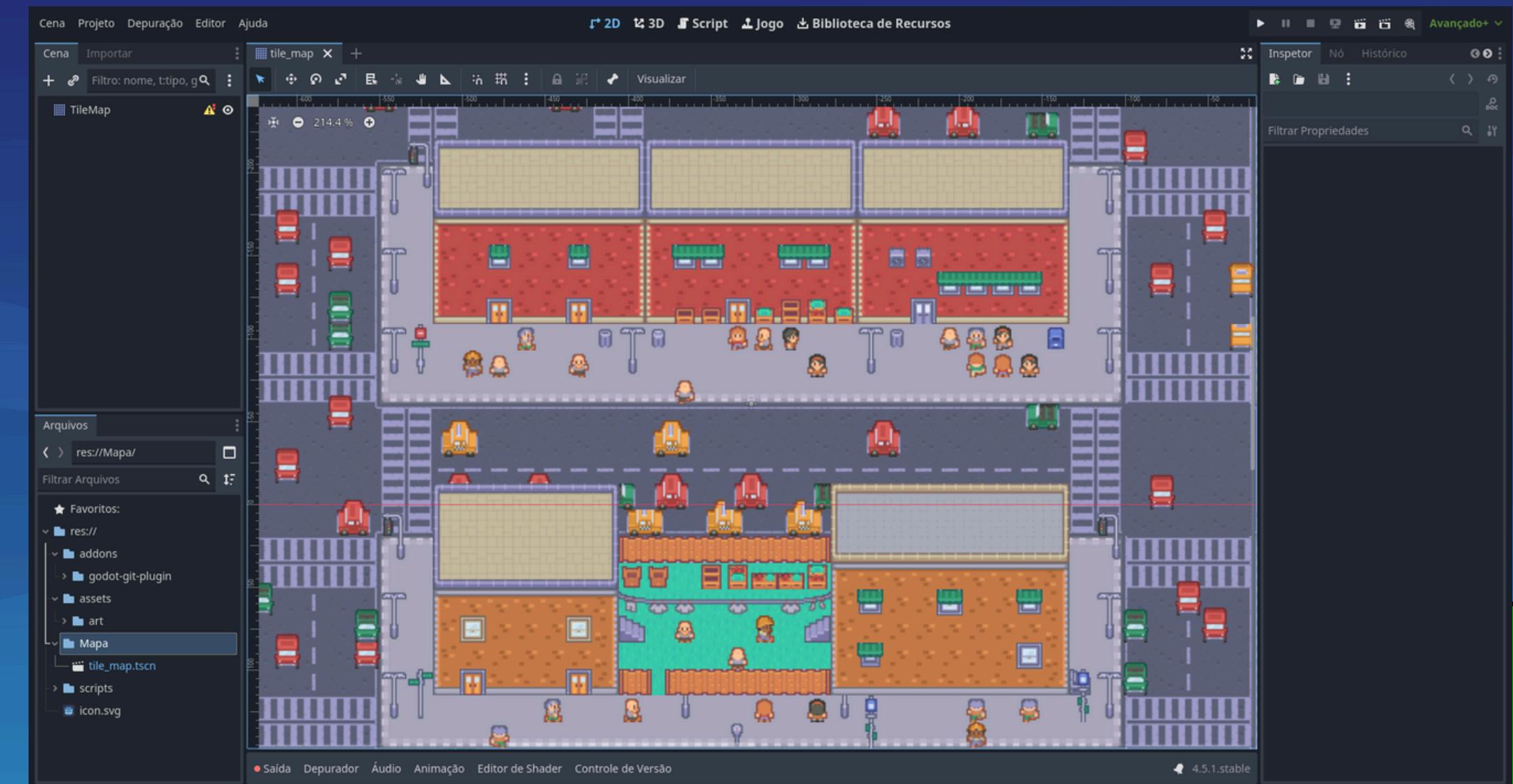
PARA DESENVOLVER O HEAVEN HEIGHTS, UTILIZAMOS A GODOT, UMA GAME ENGINE OPEN-SOURCE QUE UTILIZA GDSCRIPT, SEMELHANTE A PYTHON.

* Desenvolvimento - Arte *

O ESTILOS QUE ESCOLHEMOS PARA O JOGO FOI O DE PIXEL ART, COM UMA VISÃO DE CIMA DO JOGADOR E DO BAIRRO.

PARA CRIAR O MAPA, UTILIZAMOS TILESETS, UM TILE UM PEDAÇO PEQUENO DE ARTE, COMO GRAMA, RUA, PAREDE, TELHADO, CALÇADA, ETC.

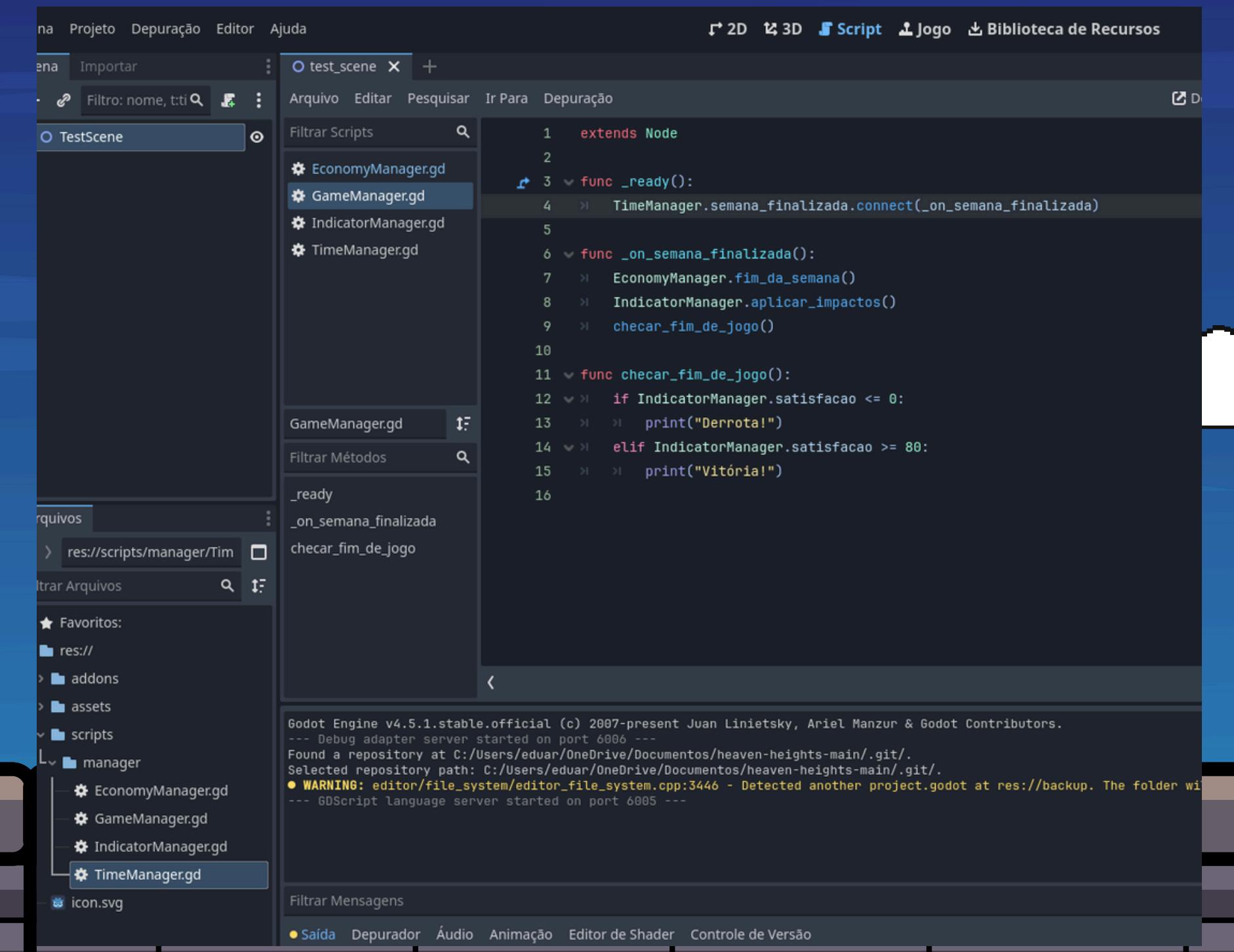
UM TILEMAP É UM “MAPA” QUE DIZ ONDE CADA BLOQUINHO DO TILESET VAI APARECER NA TELA. ELE NÃO GUARDA A ARTE, SÓ AS COORDENADAS. ELE FUNCIONA COMO UMA PLANTA BAIXA DO BAIRRO, MAS USANDO IDS DE TILES EM VEZ DE DESENHAR TUDO MANUALMENTE.



Desenvolvimento de Código

O NOSO DESENVOLVIMENTO DE CÓDIGO COMEÇA PELO ARQUIVO DE SCRIPT "GAMEMANAGER", NELE CRIAMOS FUNÇÕES QUE FINALIZAM A SEMANA E CHAMAM MAIS OUTRAS DUAS FUNÇÕES EM DOIS ARQUIVOS DIFERENTES DE SCRIPTS.

ESSE ARQUIVO DE CÓDIGO NOS PERMITE FINALIZAR A SEMANA DO GAME E ANALISAR SE O NÍVEL DE SATISFAÇÃO FOI ACIMA OU ABAIXO DE 80, SE FOR ACIMA, SERÁ CONSIDERADA UMA VITÓRIA PARA O PLAYER, CASO CONTRÁRIO, DERROTA.



The screenshot shows the Godot Engine 4.5.1 stable official interface. The top menu bar includes 'Arquivo', 'Projeto', 'Depuração', 'Editor', and 'Ajuda'. The toolbar below has icons for '2D', '3D', 'Script', 'Jogo', and 'Biblioteca de Recursos'. The main window displays a script editor with the following code:

```
1  extends Node
2
3  func _ready():
4      >> TimeManager.semana_finalizada.connect(_on_semana_finalizada)
5
6  func _on_semana_finalizada():
7      >> EconomyManager.fim_da_semana()
8      >> IndicatorManager.aplicar_impactos()
9      >> checar_fim_de_jogo()
10
11 func checar_fim_de_jogo():
12     if IndicatorManager.satisfacao <= 0:
13         >> print("Derrota!")
14     elif IndicatorManager.satisfacao >= 80:
15         >> print("Vitória!")
```

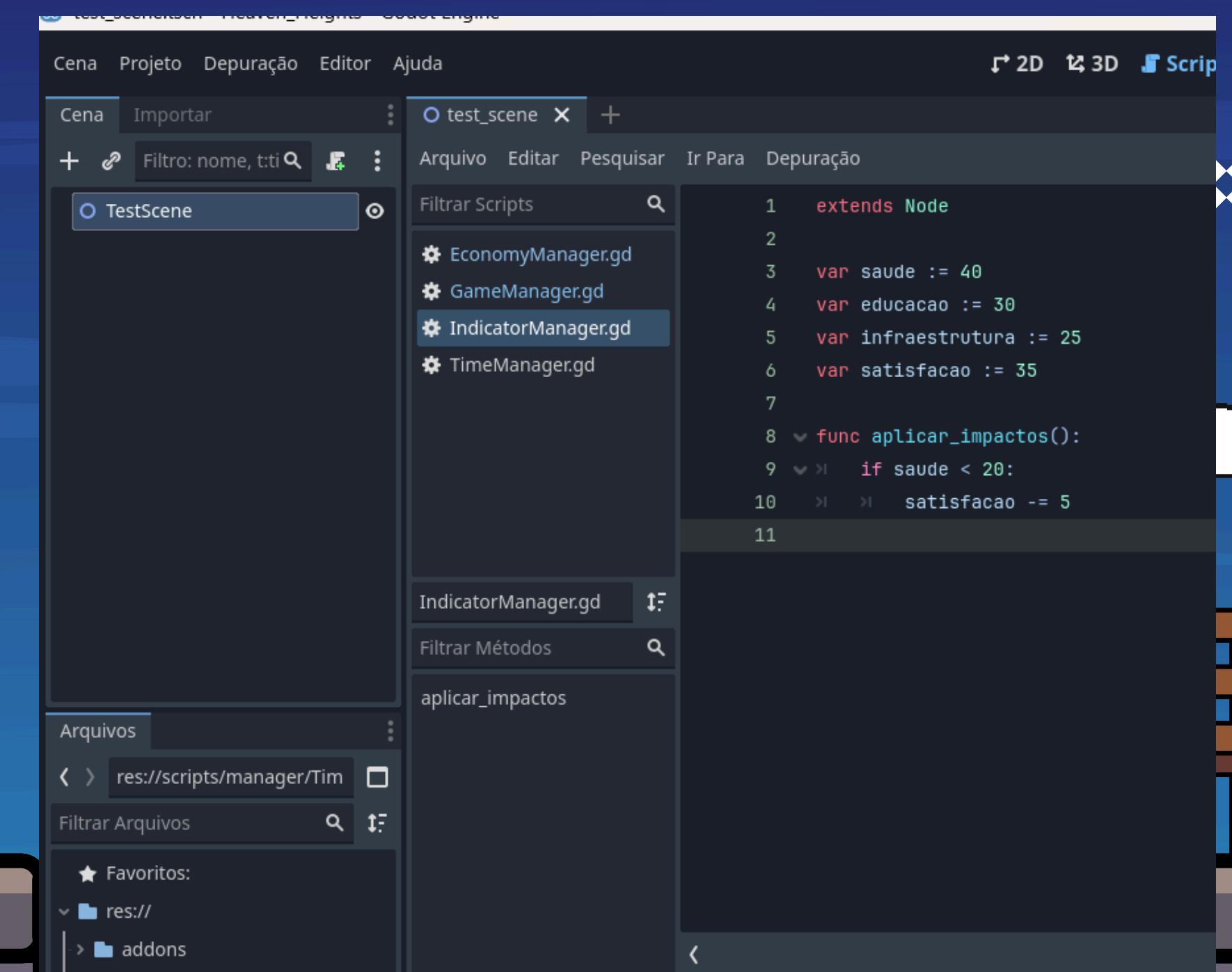
The left side features a file browser with a tree view showing 'res://', 'addons', 'assets', 'scripts', and a 'manager' folder containing 'EconomyManager.gd', 'GameManager.gd', 'IndicatorManager.gd', and 'TimeManager.gd'. The bottom status bar shows 'Godot Engine v4.5.1.stable.official (c) 2007-present Juan Linietsky, Ariel Manzur & Godot Contributors.', 'Selected repository path: C:/Users/eduar/OneDrive/Documentos/heaven-heights-main/.git/', and a warning message: 'WARNING: editor/file_system/editor_file_system.cpp:3446 - Detected another project.godot at res://backup. The folder will be deleted.' The bottom tabs include 'Filtrar Mensagens', 'Saída', 'Depurador', 'Áudio', 'Animação', 'Editor de Shader', and 'Controle de Versão'.

Desenvolvimento de Código

NOSSO SEGUNDA PARTE DE
DESENVOLVIMENTO DE CÓDIGO É O
SCRIPT "INDICATORMANAGER".

NESSE SCRIPT APRESENTAMOS UMA
FUNÇÃO DE IMPACTO SOCIAL, CASO
A SAÚDE FOR MENOR QUE 20(0
VALOR QUE SATISFAZ A SAÚDE É
IGUAL À 40), O JOGADOR JÁ
COMEÇA A PERDER PONTOS DE
SATISFAÇÃO COM OS NPCS.

ACIMA DA FUNÇÃO CITADA,
POSSUÍMOS VARIÁVEIS DE VALORES
FIXOS, CASO O NÍVEL DE
SATISFAÇÃO FOR ABAIXO DESSES
VALORES, OS NPCS FICAM
INSATISFEITOS COM OS
INVESTIMENTOS.



The screenshot shows the Godot Engine's script editor interface. The title bar indicates the project is named 'test_Scenarios' and the scene is 'Heaven_Rights - Godot Engine'. The menu bar includes 'Cena', 'Projeto', 'Depuração', 'Editor', and 'Ajuda'. The top right has buttons for '2D', '3D', and 'Script'. The main area shows a script titled 'IndicatorManager.gd' with the following code:

```
extends Node

var saude := 40
var educacao := 30
var infraestrutura := 25
var satisfacao := 35

func aplicar_impactos():
    if saude < 20:
        satisfacao -= 5
```

The script also lists other scripts in the project: EconomyManager.gd, GameManager.gd, IndicatorManager.gd (which is selected), and TimeManager.gd. The bottom left shows the file browser with 'res://scripts/manager/Tim' selected. The bottom right shows a preview window with a small tree icon.

Desenvolvimento de Código

NESTE ARQUIVO DE SCRIPT, "ECONOMYMANAGER" QUE FIZEMOS, FAZEMOS COM QUE O PLAYER INICIE SUA JORNADA COM O VALOR DE R\$ 500.000,00 PARA COMEÇAR OS INVESTIMENTOS.

NA PRIMEIRA FUNÇÃO, ELE CHAMA O "INDICATORMANAGER" E CALCULA A RECEITA PELO NÍVEL DE SATISFAÇÃO E MULTIPLICA POR 5000.

NA SEGUNDA E ÚLTIMA FUNÇÃO, APÓS O CÁLCULO DA RECEITA, ELE FAZ O CÁLCULO DO TOTAL DE DINHEIRO QUE O PLAYER POSSUI E ACRESCENTA MAIS A RECEITA, DANDO O TOTAL DE QUANTIA QUE O JOGADOR POSSUI.

The screenshot shows the Godot Engine 4.5.1 stable official IDE interface. The main window displays the script editor for a file named 'EconomyManager.gd'. The code is as follows:

```
extends Node

var dinheiro := 500000.0
var receita := 0.0

func calcular_receita():
    > receita = IndicatorManager.satisfacao * 5000
    > print("Receita calculada:", receita)

func fim_da_semana():
    > calcular_receita()
    > dinheiro += receita
    > print("Dinheiro atual:", dinheiro)
```

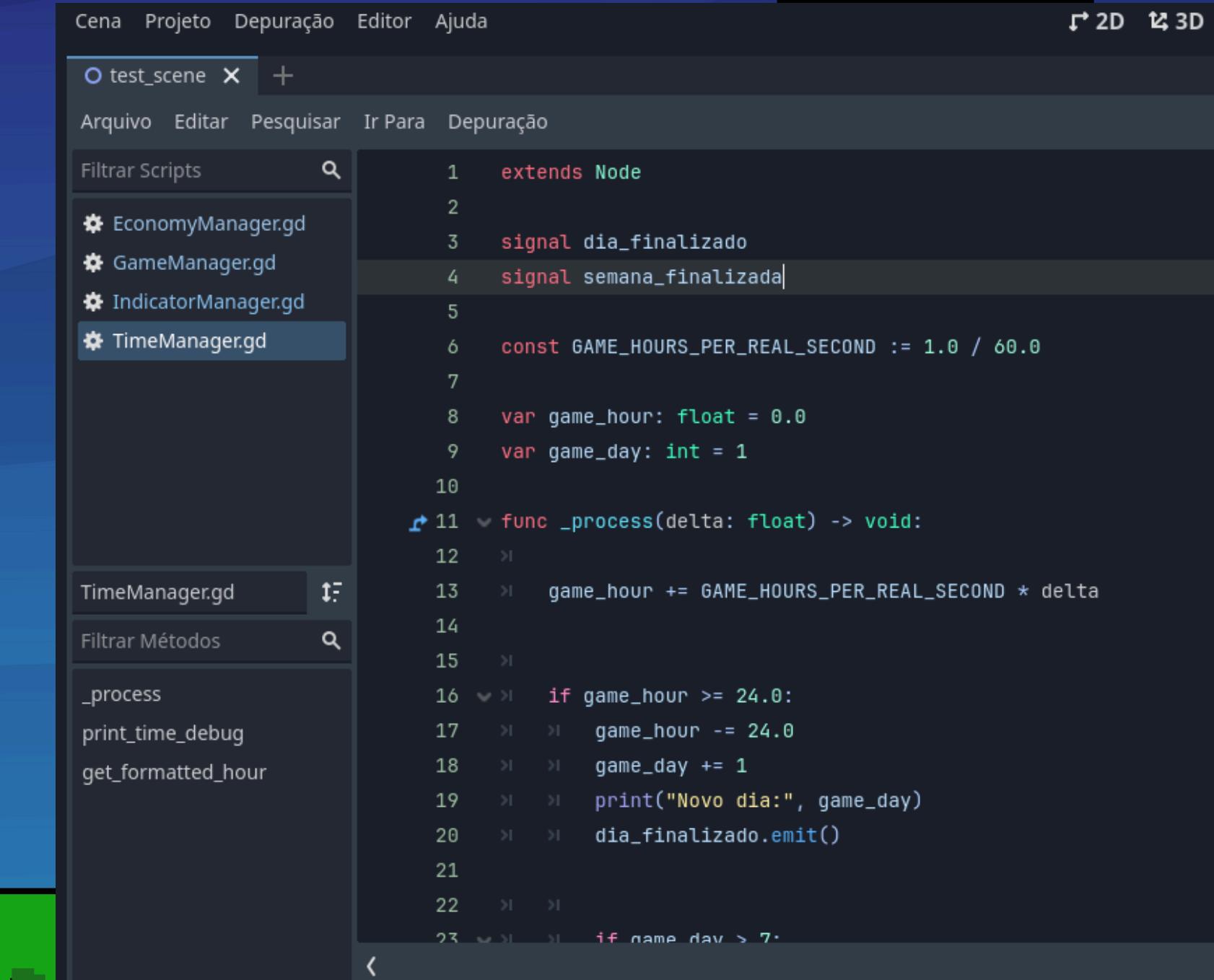
The interface includes a menu bar with 'Cena', 'Projeto', 'Depuração', 'Editor', and 'Ajuda'. The 'Editor' tab is selected. On the left, there are tabs for 'Cena', 'Importar', 'Arquivo', 'Editar', 'Pesquisar', 'Ir Para', and 'Depuração'. Below these are sections for 'Filtrar Scripts', 'Filtrar Métodos', and 'Arquivos'. The 'Arquivos' section shows a tree view with 'res://scripts/manager/TimeManager.gd' selected. The bottom status bar displays the Godot Engine version and repository information.

Desenvolvimento de Código

NOSSO ÚLTIMO ARQUIVO DE CÓDIGO,
"TIMEMANAGER".

FIZEMOS ALGO PARECIDO COM OS DIAS
DURÁVEIS DA FRANQUIA DE
JOGOS "GRAND THEFT AUTO", ONDE O
DIA POSSUI AS 24 HORAS, PORÉM,
CADA HORA EQUIVALE À 1 MINUTO,
SENDO ASSIM, 24 HORAS EQUIVALEM À
24 MINUTOS NA VIDA REAL.

NESSA PRIMEIRA PARTE DO CÓDIGO,
UTILIZAMOS ALGUNS CÁLCULOS DE
CONVERSÃO DE TEMPO PARA DEIXÁ-LO
BEM IDÊNTICO AO DE GTA, E
IMPLEMENTAMOS UMA LÓGICA DE, SE
PASSAR DOS 24 MINUTOS, UM NOVO
DIA SERÁ CONTADO.



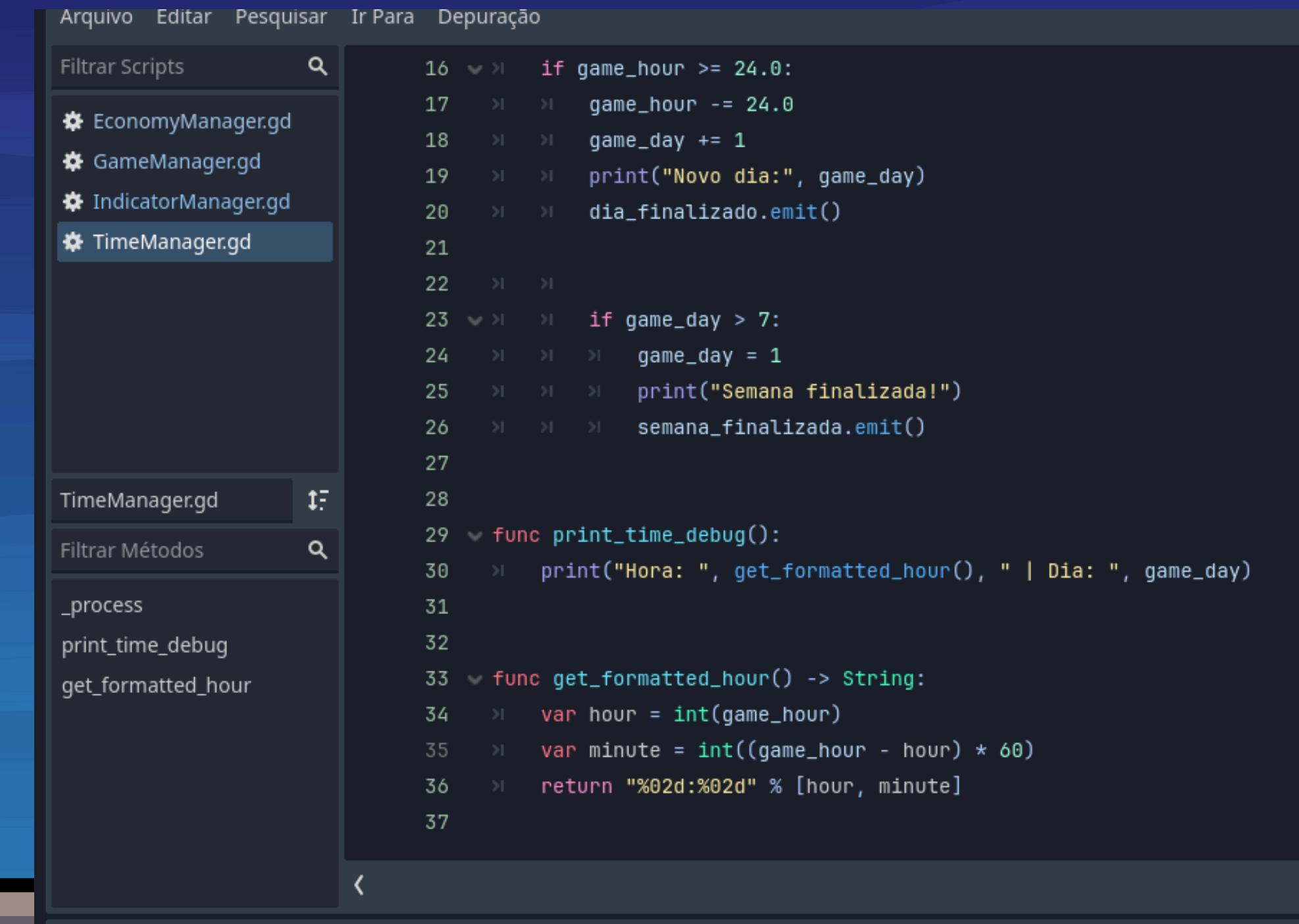
The screenshot shows the Godot Engine's GDScript editor interface. The menu bar includes 'Cena', 'Projeto', 'Depuração', 'Editor', and 'Ajuda'. The tab bar shows 'test_scene' and a '+' button. The toolbar has icons for 'Arquivo', 'Editar', 'Pesquisar', 'Ir Para', and 'Depuração'. A search bar labeled 'Filtrar Scripts' is present. The script list on the left shows 'EconomyManager.gd', 'GameManager.gd', 'IndicatorManager.gd', and 'TimeManager.gd', with 'TimeManager.gd' selected. The code editor on the right contains the following GDScript code:

```
1  extends Node
2
3  signal dia_finalizado
4  signal semana_finalizada
5
6  const GAME_HOURS_PER_REAL_SECOND := 1.0 / 60.0
7
8  var game_hour: float = 0.0
9  var game_day: int = 1
10
11 func _process(delta: float) -> void:
12
13     game_hour += GAME_HOURS_PER_REAL_SECOND * delta
14
15     if game_hour >= 24.0:
16         game_hour -= 24.0
17         game_day += 1
18         print("Novo dia:", game_day)
19         dia_finalizado.emit()
20
21
22     if game_day > 7:
23         semana_finalizada.emit()
```

Desenvolvimento de Código

NESSA ÚLTIMA PARTE DO MESMO CÓDIGO, UTILIZAMOS UMA FUNÇÃO ONDE ELE RETORNA UMA HORA FORMATADA, EXEMPLO: "08:32".

DEIXAMOS NESSE FORMATO PARA SER MAIS COERENTE COM A HORA REAL PARA O PLAYER VISUALIZAR E OBTER MAIOR ENTENDIMENTO DAS HORAS NO GAME.



The screenshot shows a game development environment with a menu bar (Arquivo, Editar, Pesquisar, Ir Para, Depuração) and a sidebar for filtering scripts and methods. The main code editor displays the TimeManager.gd script. The script contains logic for handling game time, including day and week calculations, and a function to get a formatted hour string.

```
Arquivo Editar Pesquisar Ir Para Depuração
Filtrar Scripts
EconomyManager.gd
GameManager.gd
IndicatorManager.gd
TimeManager.gd

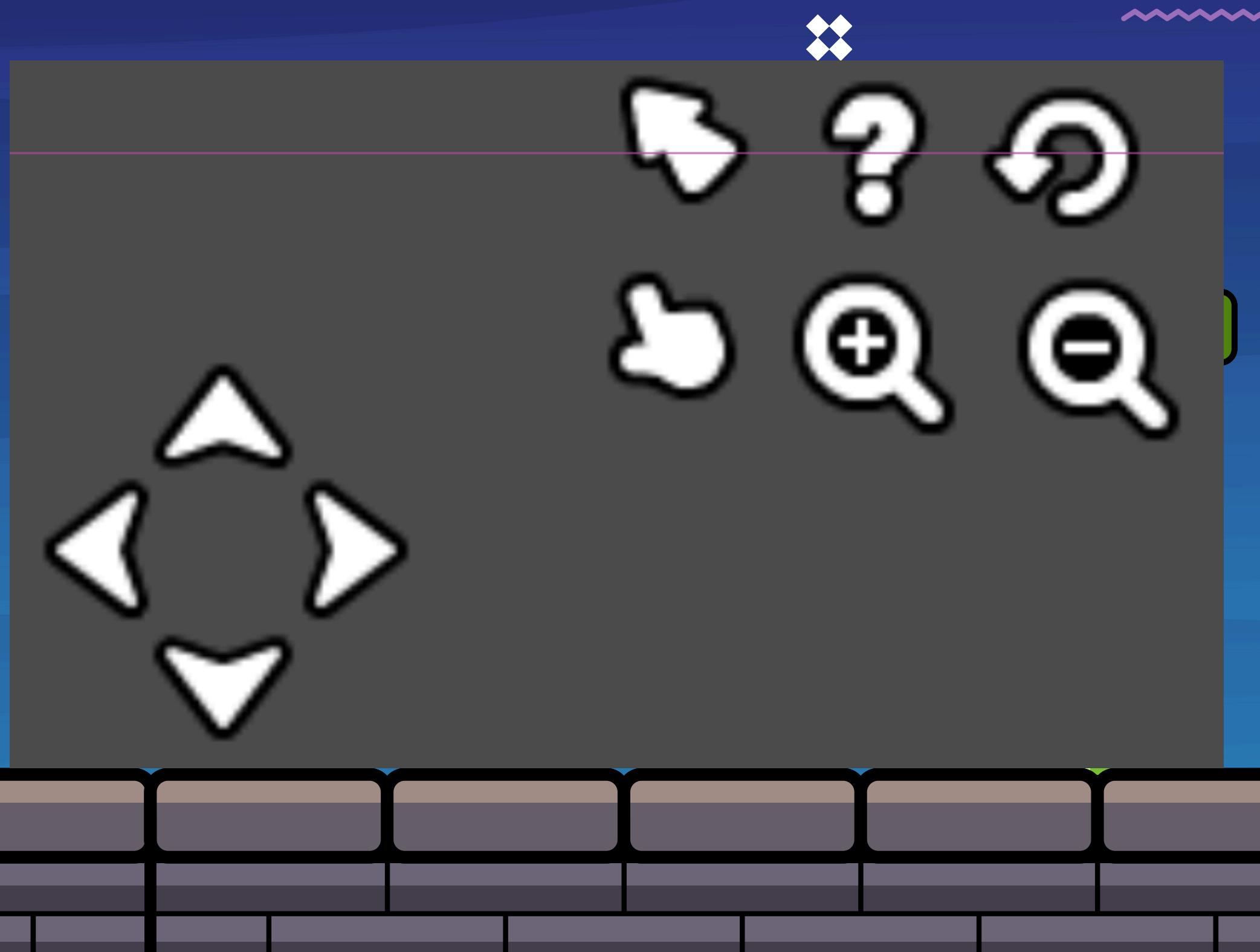
TimeManager.gd
Filtrar Métodos
_process
print_time_debug
get_formatted_hour

16 if game_hour >= 24.0:
17     game_hour -= 24.0
18     game_day += 1
19     print("Novo dia:", game_day)
20     dia_finalizado.emit()
21
22
23 if game_day > 7:
24     game_day = 1
25     print("Semana finalizada!")
26     semana_finalizada.emit()
27
28
29 func print_time_debug():
30     print("Hora: ", get_formatted_hour(), " | Dia: ", game_day)
31
32
33 func get_formatted_hour() -> String:
34     var hour = int(game_hour)
35     var minute = int((game_hour - hour) * 60)
36     return "%02d:%02d" % [hour, minute]
37
```

Desenvolvimento - Interface (UI)

PARA A INTERFACE DE USUÁRIO (UI), REALIZAMOS A CURADORIA DE ASSETS ESPECÍFICOS PARA NAVEGAÇÃO E FEEDBACK, FOCANDO PRINCIPALMENTE EM ÍCONES INTUITIVOS E SETAS INDICATIVAS.

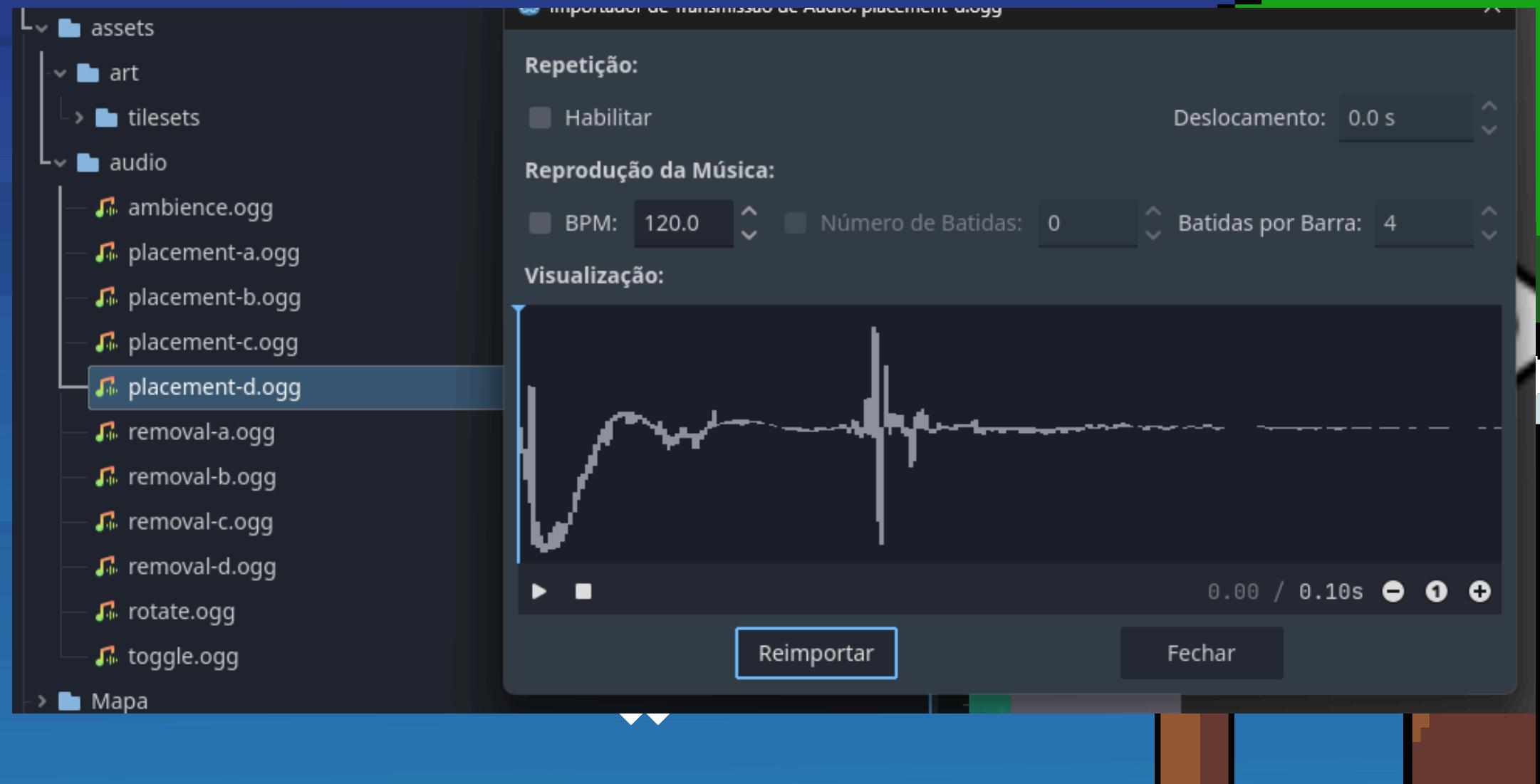
ESSES ASSETS FORAM IMPLEMENTADOS PARA GUIAR O JOGADOR, SERVINDO COMO SINALIZADORES VISUAIS RÁPIDOS PARA AÇÕES DE GERENCIAMENTO E NAVEGAÇÃO PELOS MENUS DO SISTEMA.



Desenvolvimento – Sonoplastia e Áudio

PARA A IMPLEMENTAÇÃO SONORA,
OPTAMOS POR CONVERTER E
UTILIZAR TODOS OS ARQUIVOS
DE ÁUDIO (TRILHAS E EFEITOS)
NO FORMATO .OGG.

ESCOLHEMOS ESSE FORMATO PELA
SUA ALTA COMPATIBILIDADE COM
A GODOT ENGINE E PELA
OTIMIZAÇÃO: ELE OFERECE
EXCELENTE QUALIDADE DE SOM
OCUPANDO POUCO ESPAÇO DE
MEMÓRIA, GARANTINDO QUE O
JOGO RODE DE FORMA FLUIDA.



Oh my god

