



# LICENCIATURA ENGENHARIA INFORMÁTICA E MULTIMÉDIA

## DESENVOLVIMENTO DE APLICAÇÕES MÓVEIS

---

### Relatório Final

---

DATA: 4 JULHO 2021

*Docentes:*

Eng. Pedro Fazenda

Eng. Ana Correia

*Realizado por:*

Luís Fonseca - 45125

# Índice de Conteúdos

<b>1</b>	<b>Introdução</b>	<b>4</b>
<b>2</b>	<b>Fase de Concepção</b>	<b>5</b>
2.1	Principais Características . . . . .	5
2.2	Regras . . . . .	5
2.3	Business Model . . . . .	6
2.4	Ferramentas . . . . .	6
2.5	Mockups . . . . .	7
<b>3</b>	<b>Fase Pre-Produção</b>	<b>9</b>
3.1	Feedback do utilizador . . . . .	9
3.1.1	Utilizador 1 . . . . .	10
3.1.2	Utilizador 2 . . . . .	10
3.1.3	Utilizador 3 . . . . .	11
3.1.4	Utilizador 4 . . . . .	11
3.2	Diagrama EA . . . . .	12
3.3	Procedimento inicial . . . . .	12
3.3.1	Diagrama de classes do protótipo . . . . .	13
3.3.2	Resultado do protótipo . . . . .	14
<b>4</b>	<b>Fase de produção e Após-Produção</b>	<b>15</b>
4.1	Após o protótipo inicial . . . . .	15
4.1.1	Construção dos mapas/níveis . . . . .	15
4.1.2	Sprites das Personagens . . . . .	19
4.2	Diagrama de Classes - os diferentes Menus . . . . .	20
4.3	Resultado Menu principal . . . . .	21
4.4	Resultado Menu de opções . . . . .	22
4.5	Resultado ecrã vitória . . . . .	23
4.6	Resultado ecrã Fim do Jogo . . . . .	23
4.7	O jogo . . . . .	24
4.8	Diagrama de Classes - Jogo . . . . .	24
4.9	Inimigos/Inteligência Artificial . . . . .	32
4.10	Personagem Principal . . . . .	33
4.11	Interface do jogador . . . . .	34
4.12	Resultado do Jogo . . . . .	34
4.13	Extras adicionados . . . . .	35
<b>5</b>	<b>Conclusões</b>	<b>36</b>

## **6 Bibliografia**

**37**

## List of Figures

1	MainMenu . . . . .	7
2	Jogo . . . . .	7
3	Menu de opções . . . . .	8
4	Diagrama dos Mockups construídos . . . . .	8
5	Diagrama Entidade Associação . . . . .	12
6	Diagrama de classes do protótipo . . . . .	13
7	Protótipo criado . . . . .	14
8	SpriteSheet, contendo diversas texturas . . . . .	16
9	Construção do mapa usando o tiled . . . . .	17
10	Layers do mapa . . . . .	17
11	Sprites das personagens do jogo . . . . .	19
12	Diagrama de classes do package Screens . . . . .	20
13	Ecrã Menu Principal . . . . .	21
14	Ecrã Menu Opções . . . . .	22
15	Ecrã Vitória . . . . .	23
16	Ecrã Fim do jogo . . . . .	23
17	Diagrama de classes para a interface do jogador . . . . .	24
18	Diagrama de classes do package Utils . . . . .	25
19	Diagrama de classes SuperLuigi . . . . .	26
20	Organização do package Sprites . . . . .	26
21	Diagrama de classes do package Enemies . . . . .	27
22	Diagrama de classes do package TiledObjects . . . . .	28
23	Diagrama de classes do package items . . . . .	29
24	Diagrama de classes da classe Luigi . . . . .	30
25	Organização do projeto . . . . .	31
26	Inimigo gomba com mutações . . . . .	32
27	Inimigo tartura . . . . .	32
28	Inimigo tartaruga, depois do jogador ter saltado por cima . . . . .	32
29	Personagem principal . . . . .	33
30	Personagem principal, após ter comido um cogumelo . . . . .	33
31	Interface do jogador . . . . .	34
32	Ecrã Jogo . . . . .	34
33	Jogo com suporte a computador . . . . .	35

# 1 Introdução

Este relatório tende a apresentar o desenvolvimento do projeto final elaborado para a unidade curricular de Desenvolvimento de Aplicações Móveis. Neste relatório, vai ser apresentado todas as etapas concretizadas para o projeto final, assim como certas implementações adotadas para as diferentes mecânicas.

Visto que o desenvolver deste projeto, tem continuação de tarefas realizadas anteriormente, este relatório irá conter, também, os outros elementos que foram possíveis para o desenvolvimento deste projeto.

Como já referido, o projeto final da unidade curricular da disciplina de Desenvolvimento de Aplicações Móveis optava por escolher em desenvolver uma aplicação ou um jogo.

Foi feita a escolha de um jogo, e com o nome de “Luigi Adventure: A cure for Covid”, que terá como objetivo o jogador concluir um modo história, contendo 2 níveis. Para o desenvolvimento deste projeto, vai ser usada uma base para o desenvolvimento deste jogo, de nome Super Mario, lançado no ano de 1985.

Irá também ser incluído o tema do covid-19, que será abordado mais em pormenor ao longo deste relatório.

Em termos de pessoas ”alvo”, o jogo vai ser desenvolvido para todas as idades e para os jogadores que gostam dos jogos clássicos.

## 2 Fase de Concepção

Esta fase consiste em definir quais as funcionalidades que o jogo vai conter. Vai ser proposto um business model, para o jogador pagar uma quantidade para desbloquear mais níveis. No final, é apresentado um diagrama de mockups, funcionando como um esboço, do jogo desenvolvido.

### 2.1 Principais Características

- O jogador irá conter power-ups, que permite aumentar o tamanho da personagem. Esse power-up, consiste num cogumelo, que pode ser encontrado, caso o jogador esmague uma um bloco mistério.
- Os inimigos vão ser sprites do Super Mario, com mutações do vírus da covid-19.
- O nível será concluído quando o jogador chegar ao fim do nível.
- Quando o jogador perde o jogo, surge uma tela de “game over”, onde dá a oportunidade de reiniciar o nível atual.
- O jogador perde o jogo, quando um inimigo entra em contato com o jogador.
- Cada elemento do jogo irá conter som específico.

### 2.2 Regras

- O jogador apenas contém 1 vida, caso chegue ao zero, reinicia o nível;
- O jogador ganha o jogo, quando chega ao final do nível;
- O jogador avança no jogo, partindo blocos (pedra e bloco mistério) colecionando moedas para aumentar a pontuação,e power ups, que ajudam na progressão do jogo.
- O jogador tem um tempo limitado para completar cada nível. Caso esse tempo acabe, perde o jogo, voltando ao início do nível.

## 2.3 Business Model

O Bussiness Model irá consistir num pacote de “master-levels”. Este pacote, consiste em níveis com um maior grau de dificuldade, em relação aos dois níveis fornecidos. Esses níveis apresentam: novos inimigos, sendo mais letais e mais rápidos e um novo mundo. Caso o utilizador efetue a subscrição, é fornecido, todos os meses, um novo nível. A quantia para esta subscrição é de 1€, sendo descontada mensalmente.

## 2.4 Ferramentas

O desenvolvimento deste jogo, irá consistir num conjunto de ferramentas, que iram ajudar na construção deste projeto.

- libgdx: framework em Java, que permite o desenvolvimento de jogos multi-plataforma (2D ou 3D) criado pela companhia BadLogicGames.
- Tiled Map Editor: editor de níveis 2D, que permite ajudar no desenvolvimento do jogo. Uma das suas principais funções é editar mapas de várias formas, mas também suporta a colocação gratuita de imagens, bem como maneiras poderosas de anotar o seu nível com informações extras usadas pelo jogo.

## 2.5 Mockups

Nesta seção é possível de visualizar os diferentes mockups que iram ser construídos para este jogo. É de notar que isto é apenas um esboço e o produto final, pode variar do que está aqui apresentado:

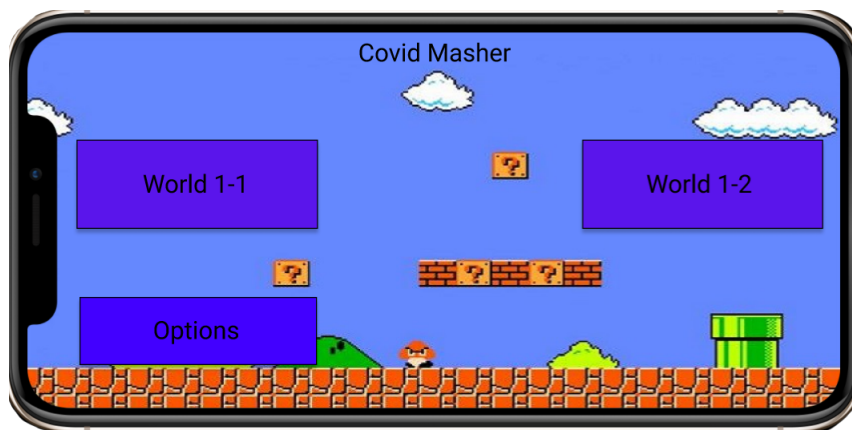


Figure 1: MainMenu

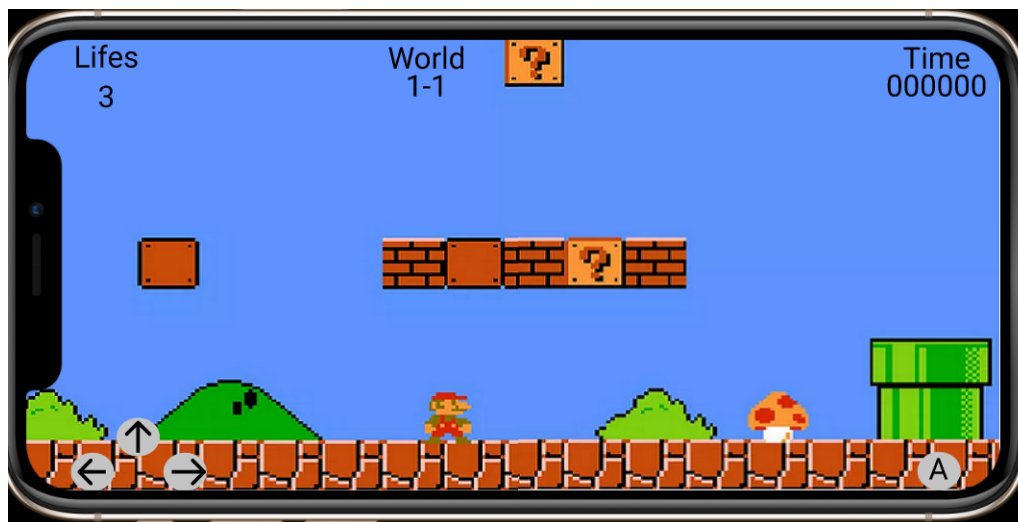


Figure 2: Jogo



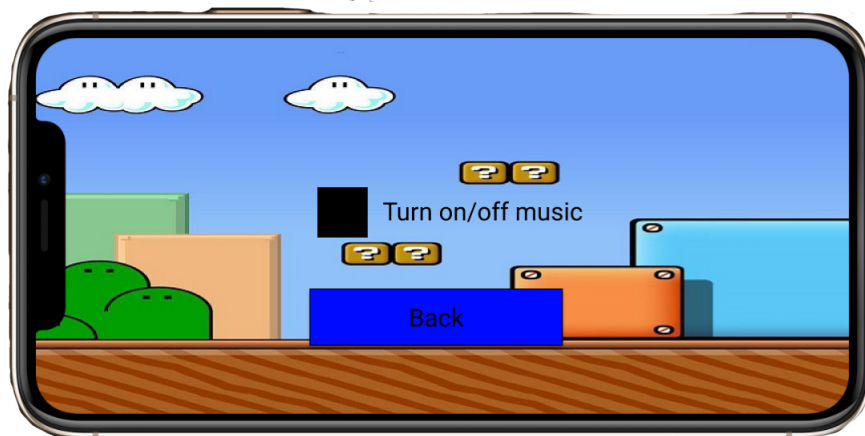


Figure 3: Menu de opções

De seguida, é apresentado o jogo final, com as diferentes associações entre os mockups desenhados

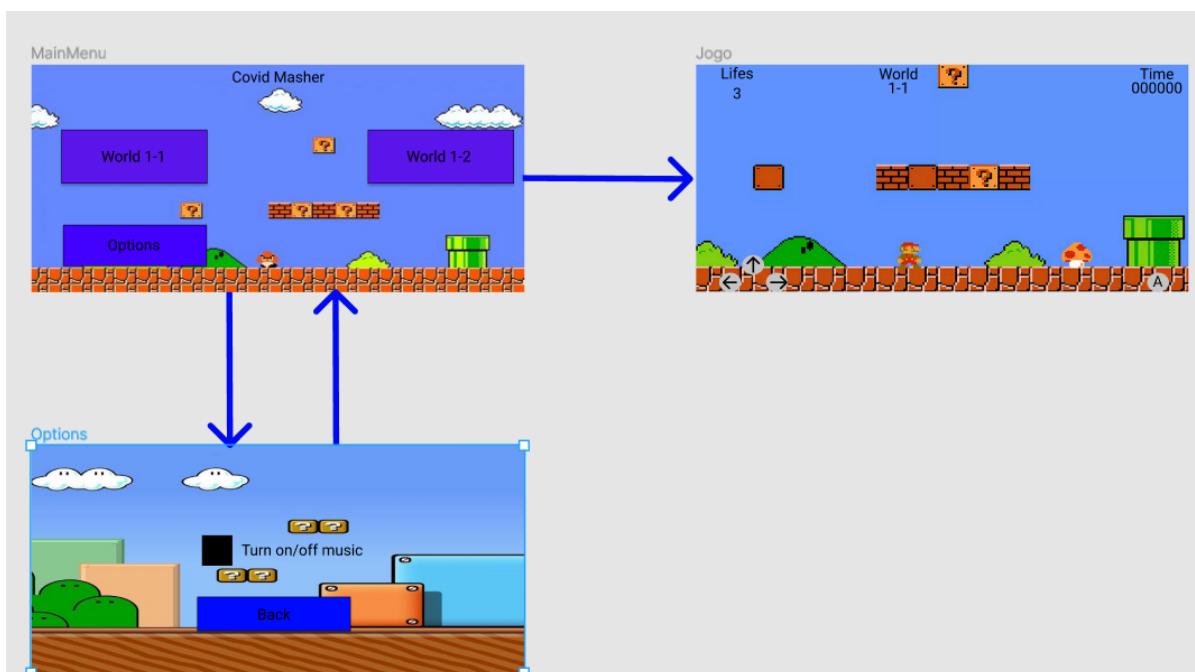


Figure 4: Diagrama dos Mockups construídos

## 3 Fase Pre-Produção

Esta fase centra-se mais no feedback do utilizador. Fornecendo um questionário, vai ser possível de criar este jogo, com base no feedback dos diferentes utilizadores. No final desta fase, é apresentado um protótipo do jogo, mostrando o progresso deste projeto.

### 3.1 Feedback do utilizador

Após ter sido efetuado um protótipo do jogo(e usado uma versão do jogo implementada em diferentes plataformas) foi proposto um questionário, a diferentes utilizadores, acerca da opinião do jogo que irá ser desenvolvido.

Segue-se em baixo, as perguntas que foram enviadas para receber o feedback dos diferentes utilizadores:

- What is the purpose of the game?
- What do you think of the game in general?
- Do you like it?
- What does the player need to do to succeed?
- When does the game end?
- Do you find it challenging?
- Would you play it?
- Do you think there is any unsuitable element?
- How much are you willing to pay (monthly), to have access to payable features?

Na seção a seguir, é possível de encontrar as diferentes respostas que os diferentes utilizadores escreveram, acerca do jogo.

### 3.1.1 Utilizador 1

- Completar um dos níveis fornecidos
- Gostei
- Tem o Luigi por isso 10/10
- Saber os controlos e chegar ao fim do nível
- Quando o nível acaba
- Sim
- Sim
- Não
- 1€

### 3.1.2 Utilizador 2

- O jogo é baseado no jogo Super Mario e consiste em o jogador percorrer determinados níveis, evitando inimigos e colecionando itens que o ajudarão.
- É um bom jogo, principalmente a ideia de incluir os inimigos como imagens do vírus.
- Sim
- O jogador para ser bem sucedido, tem que terminar o nível em que se encontra, dentro do tempo.
- Quando o jogador chega ao final do nível.
- Penso que sim, mas apenas poderei dar uma resposta quando o jogar.
- Sim
- Não existe nenhum elemento inadequado, mas poderia sugerir que em vez da personagem Super Mário, seja outra adequada ao nome do jogo.
- Penso que o valor proposto de 1 euro é adequado.

### 3.1.3 Utilizador 3

- É um jogo que possui 2 níveis, e cujo objetivo é chegar ao fim de cada nível eliminando inimigos no processo, e podendo receber upgrades, como cogumelos.
- É bom.
- Sim.
- Chegar ao fim do nível vivo, conseguindo sobreviver e eliminar inimigos.
- Quando o jogador chega à bandeira do fim do mapa.
- Mais ou menos.
- Sim.
- Não.
- Nada.

### 3.1.4 Utilizador 4

- É um jogo 2d estilo plataforma
- É um jogo muito divertido
- Sim
- Chegar ao fim do mapa sem morrer
- Quando o jogador chega ao fim do mapa com sucesso
- Sim, sem dúvida
- Sim
- Não
- 0.50€

### 3.2 Diagrama EA

Para o projeto escolhido, foi efetuado uma ligação à base de dados, usando o Firebase. Esta mecânica, vai permitir guardar as pontuações dos diferentes jogadores que iram jogar os diferentes níveis. Para isso, foi criado um diagrama EA, onde se pode ver a associação entre o nível e a pontuação. Cada um destes objectos vai conter, um ou mais elementos (ou seja, um ou mais níveis, e um, ou mais, pontuações).

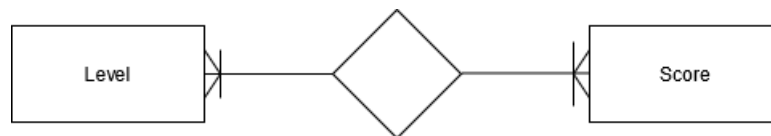


Figure 5: Diagrama Entidade Associação

### 3.3 Procedimento inicial

Para ter uma noção de como será a produção deste projeto, irá ser desenvolvido um protótipo do jogo. O objetivo deste protótipo vai focar em ter, como passo de partida, uma versão já jogável do jogo, e dar a conhecer, os diferentes métodos e a capacidade, que o libgdx contém.

Para isso, procedeu-se à produção do protótipo inicial. Este protótipo, como se pode ver na imagem mais abaixo, apenas contém um quadrado a mover-se num plano, através de setas colocadas como imagens. Foi também implementado uma interface ao utilizador, efetuando uma réplica da interface do jogo "Super Mario", contendo a informação necessária para o utilizador.

### 3.3.1 Diagrama de classes do protótipo

Na imagem abaixo, é possível de verificar o diagrama de classes, para o protótipo criado.

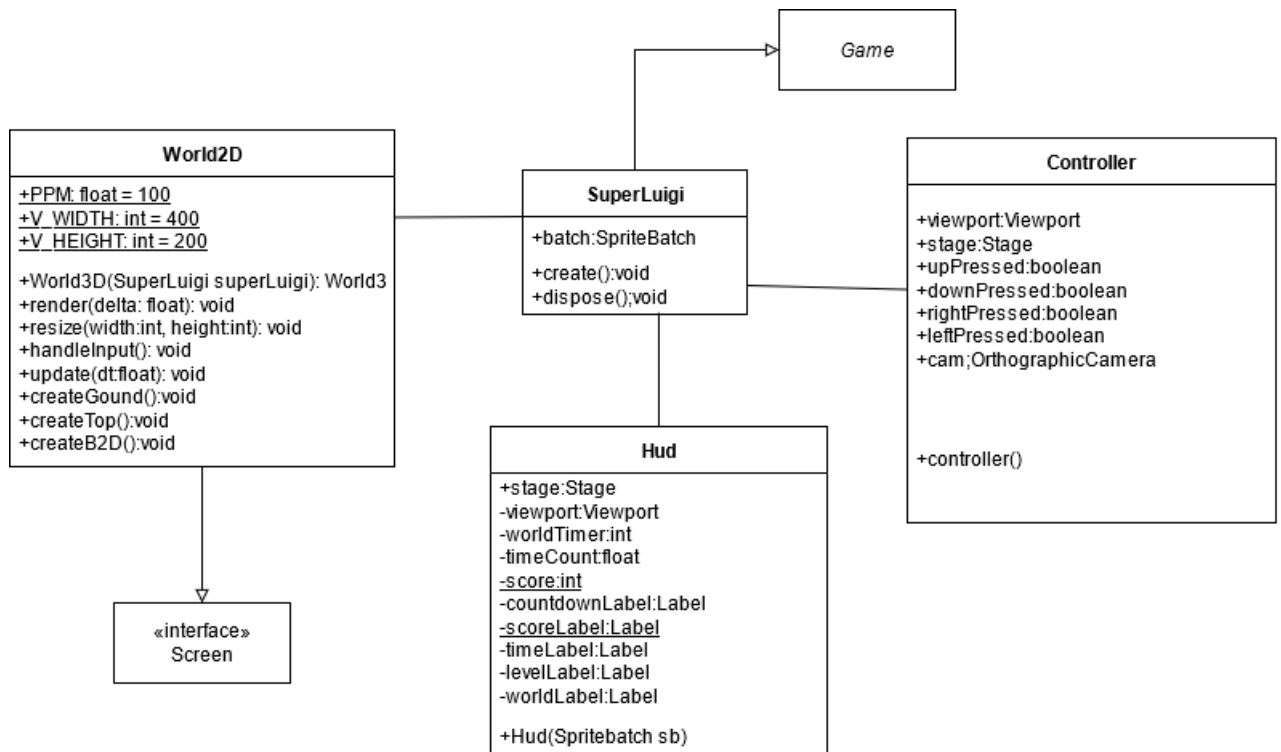


Figure 6: Diagrama de classes do protótipo

### 3.3.2 Resultado do protótipo

Após a elaboração do diagrama de classes, foi feita a implementação do protótipo. Tendo simular. No que diz respeito à parte da interface do utilizador, tentou-se replicar, o melhor possível, a estrutura do jogo Super Mario.

Em termos de jogabilidade, jogador poderá contar com as setas, esquerda e direita, para se movimentar no jogo, assim como, a seta para cima, para saltar no jogo.



Figure 7: Protótipo criado

## **4 Fase de produção e Após-Produção**

Esta seção final do relatório, vai ter em conta a implementação do jogo. Usfruindo da ferramenta "Tiled", foi possível criar os dois níveis que são apresentados. Após a criação dos níveis, passou-se para a implementação do jogo, usando JAVA, através do programa android studio, para a construção do jogo.

### **4.1 Após o protótipo inicial**

Depois de ter sido efetuado um protótipo do jogo, passou-se para a construção do projeto e das suas funcionalidades. Antes de passar para as mecânicas do jogo, foi necessário construir os mapas, usando a ferramenta já mencionada, e um conjunto de sprites, para identificar a personagem principal e os inimigos.

#### **4.1.1 Construção dos mapas/níveis**

Passando então para a construção dos mapas, foram construídos dois níveis, através duma spritesheet, contendo várias texturas. Na imagem a seguir, pode ser possível visualizar spritesheet usada.



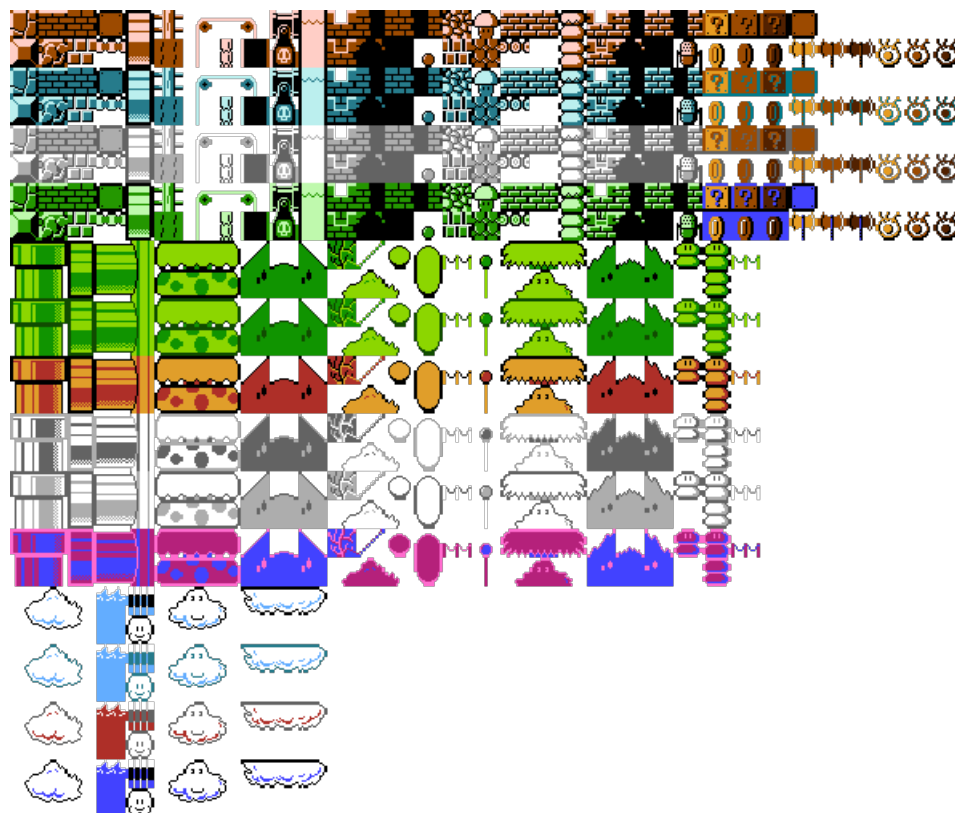


Figure 8: SpriteSheet, contendo diversas texturas

Após ter sido feito o carregamento desta spritesheet para dentro do tiled, passou-se para a construção dos mapas. Na imagem a seguir, é possível de ver um exemplo desta ferramenta a ser usada.

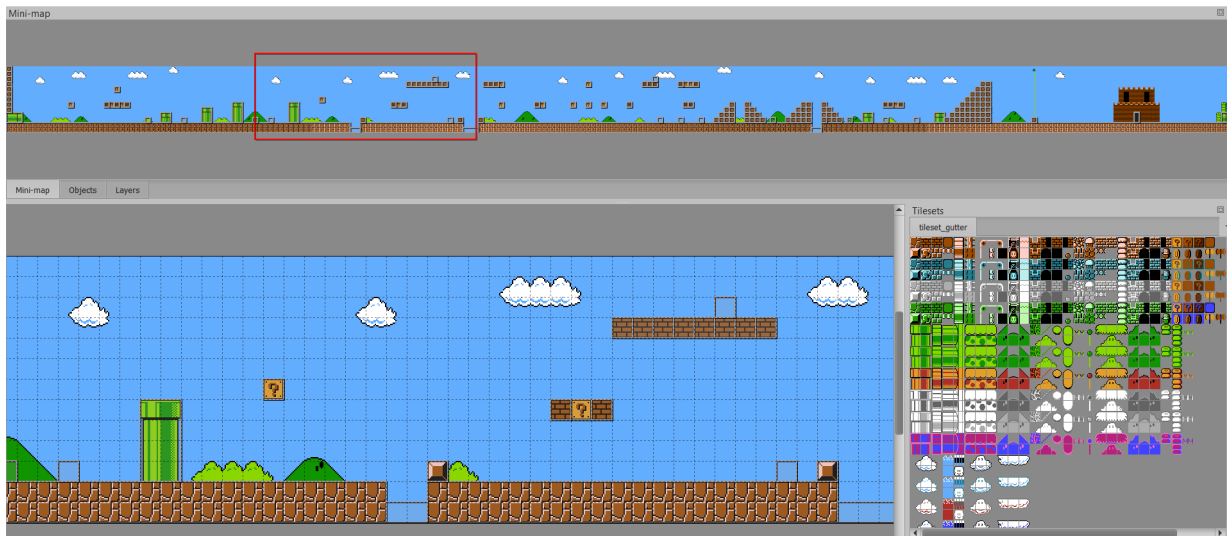


Figure 9: Construção do mapa usando o tiled

Como é de observar na imagem de cima, é possível de ver a spritesheet que está ser usada para a construção do mapa. Mais acima, é possível de ver o comprimento, por completo, do mapa construído.

No entanto, foi necessário criar um conjunto de layers, dentro deste mapa. Estas layers, vão permite que o jogo consiga fazer aparecer e distinguir os diferentes componentes, quer seja, fazer aparecer os inimigos, quer seja, mudar de ecrã.

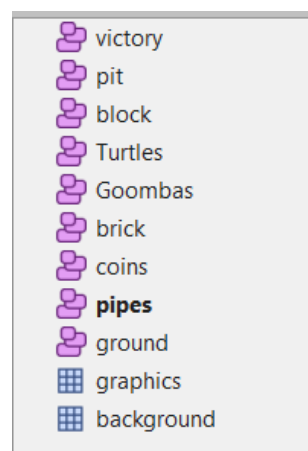


Figure 10: Layers do mapa

Como se pode verificar na imagem, existe um vasto conjunto de layers, pelo que as que correspondem à cor roxa, iram ser tratadas como objetos. As layers de cor azul, correspondem apenas à cor de fundo e os diferentes elementos de desenho, como por exemplo: os canos, as pedras, e as caixas mistério.

- Layer background: corresponde à cor de fundo do mapa.
- Layer graphics: corresponde aos diferentes elementos de desenho colocados no mapa(canos, pedras e caixas mistério).
- Objecto ground: permite desenhar um chão, onde o jogador pode movimentar-se.
- Objecto coins: objeto que corresponde aos blocos mistério. Pode conter moedas, para aumentar a pontuação, ou um cogumelo.
- Objecto brick: consiste em blocos de pedra/tijolo onde o jogador pode andar. Caso tenha comido um cogumelo, o jogador fica maior, e pode partir este tipo de blocos.
- Objecto Goombas: corresponde a um dos inimigos do jogo.
- Objecto Turtles: corresponde a um dos inimigos do jogo.
- Objecto bloco: corresponde a um outro tipo de blocos que existe no jogo. Apenas serve como obstáculo, tendo a necessidade do jogador saltar por cima deste obstáculo.
- Objecto pit: corresponde a partes do mapa, onde não existe chão, ou seja, existe apenas um buraco.
- Objecto victory: caso o jogador chegue ao fim do nível, transita para o "Victory Screen".

### 4.1.2 Sprites das Personagens

O último passo antes de passar para a implementação do jogo, foi precisar de mais uma spritesheet, mas desta vez, contendo as diferentes sprites das personagens. Na imagem abaixo, é mostrada a imagem necessária, que depois, irá dar jeito, para associar estas sprites, aos diferentes componentes do jogo.



Figure 11: Sprites das personagens do jogo

## 4.2 Diagrama de Classes - os diferentes Menus

Na imagem a seguir, pode ser visto o diagrama de classes, que foi construído para os menus, que correspondem aos diferentes ecrãs.

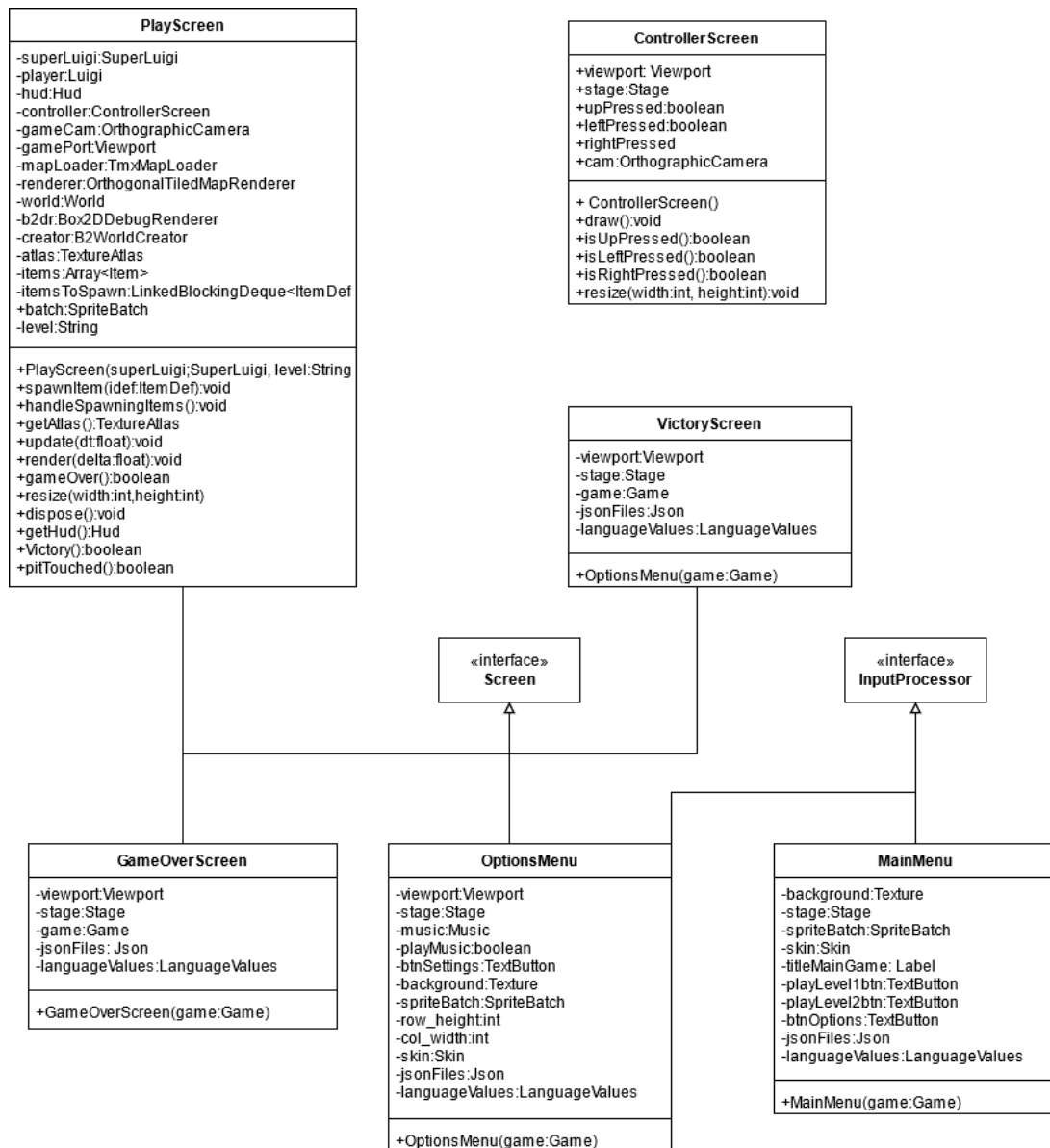


Figure 12: Diagrama de classes do package Screens

### 4.3 Resultado Menu principal

Após ter sido elaborado um diagrama de classes, passou-se para a implementação dos diferentes ecrãs. A primeira implementação consistiu num menu principal, onde pode ser visto:

- O título do jogo.
- Um botão que permite iniciar o primeiro nível.
- Um botão que permite iniciar o segundo nível.
- Um botão, que leva para o menu das opções.

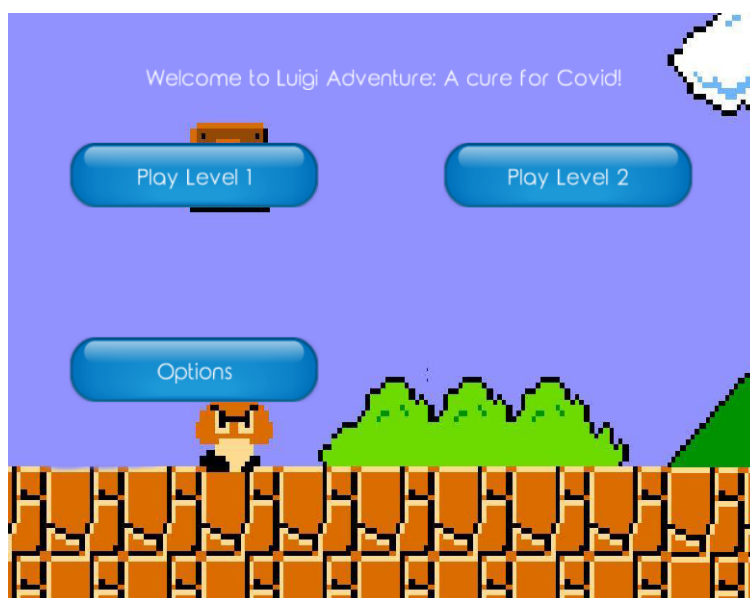


Figure 13: Ecrã Menu Principal

#### 4.4 Resultado Menu de opções

De seguida, foi feita a implementação do menu de opções. Na imagem podemos verificar que este menu contém:

- Um botão que permite ligar e desligar a música do jogo.
- Um botão que permite voltar ao menu principal.



Figure 14: Ecrã Menu Opções

## 4.5 Resultado ecrã vitória

De seguida, foi feita a implementação do ecrã de vitória. Na imagem a seguir, podemos verificar que sempre que um nível é concluído, aparece uma imagem, para clicar no ecrã, para passar po próximo nível(quando termina o primeiro nível) e ir para o menu principal(quando termina o segundo nível).

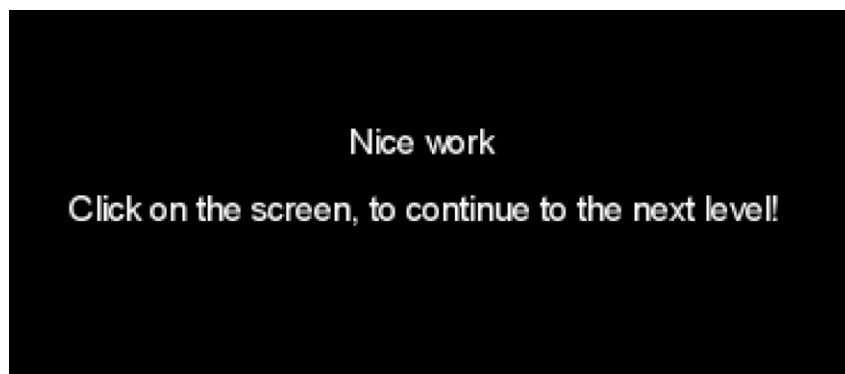


Figure 15: Ecrã Vitória

## 4.6 Resultado ecrã Fim do Jogo

De seguida, foi feita a implementação do ecrã de fim do jogo. Na imagem podemos verificar que sempre que o jogador morre, aparece uma imagem, para clicar no ecrã, permite reiniciar o nível atual.

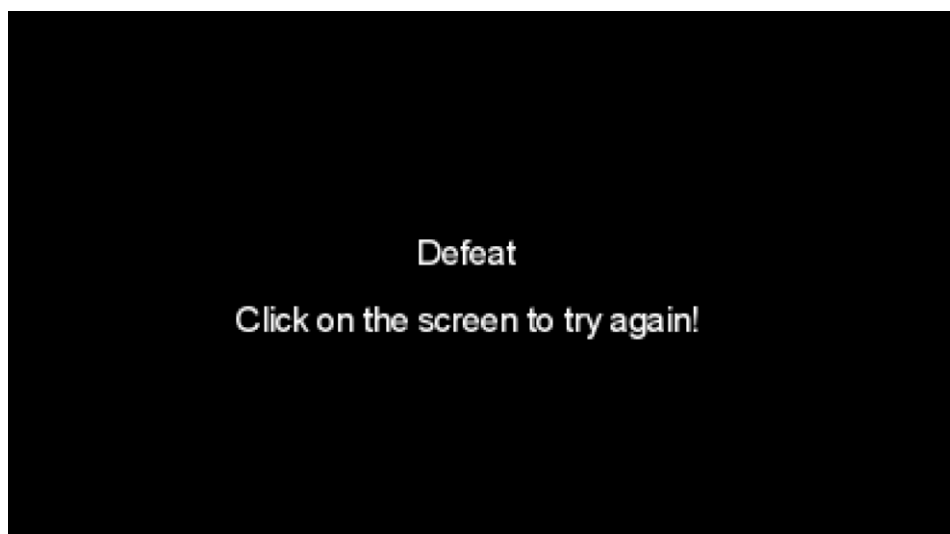


Figure 16: Ecrã Fim do jogo



## 4.7 O jogo

Esta parte irá focar no desenvolvimento do jogo. Irá ser apresentado os diferentes diagramas de classes, para a elaboração do jogo, assim como, a apresentação dos inimigos e da personagem principal.

## 4.8 Diagrama de Classes - Jogo

No que diz respeito ao jogo, foi necessário construir um conjunto de diagrama de classes, para os diferentes elementos. Nas imagens a seguir, podem ser encontrados os diferentes diagramas construídos.

O diagram abaixo, corresponde ao desenho da interface do jogador.

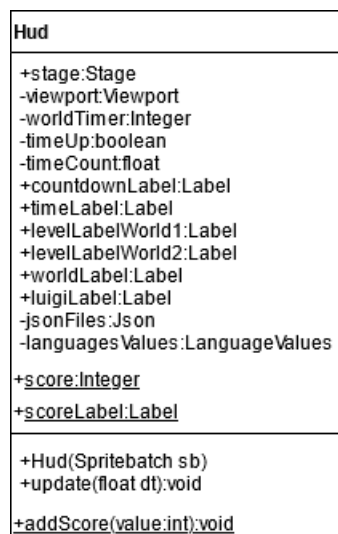


Figure 17: Diagrama de classes para a interface do jogador

O diagrama a seguir corresponde às diferentes interações que se pode produzir com as diferentes personagens. Quer seja fazer aparecer mais, ou menos inimigos, quer seja, efetuar colisão entre o jogador e os inimigos.

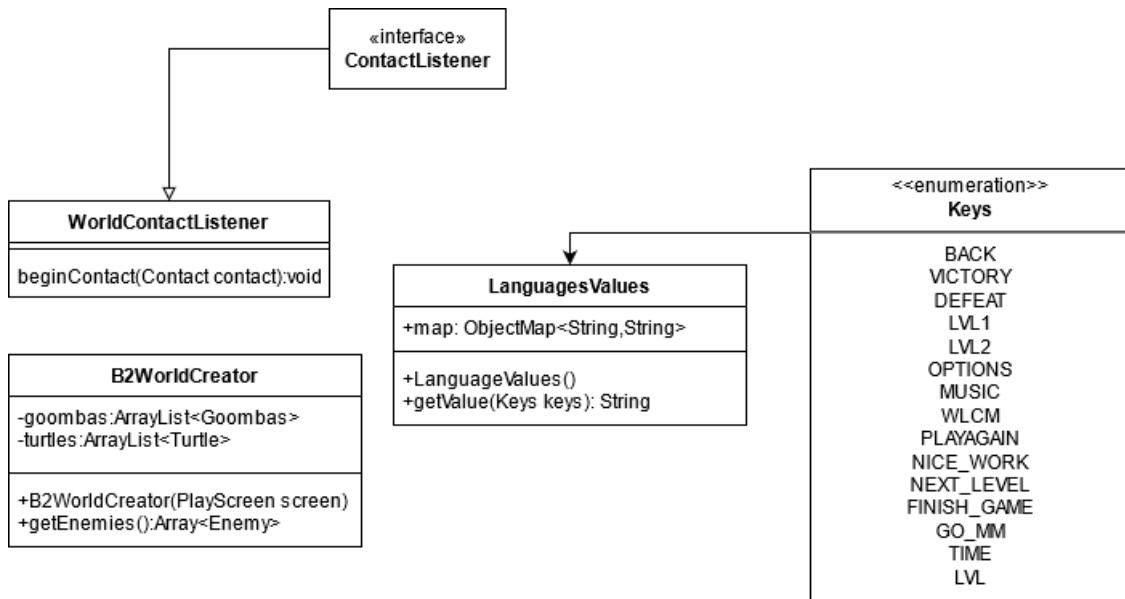


Figure 18: Diagrama de classes do package Utils

O diagrama a seguir, funciona como se fosse uma base de dados, onde permite adicionar as dimensões do mapa, adaptando a cada tela, assim como, as variáveis que permitem distinguir entre o jogador e os inimigos.

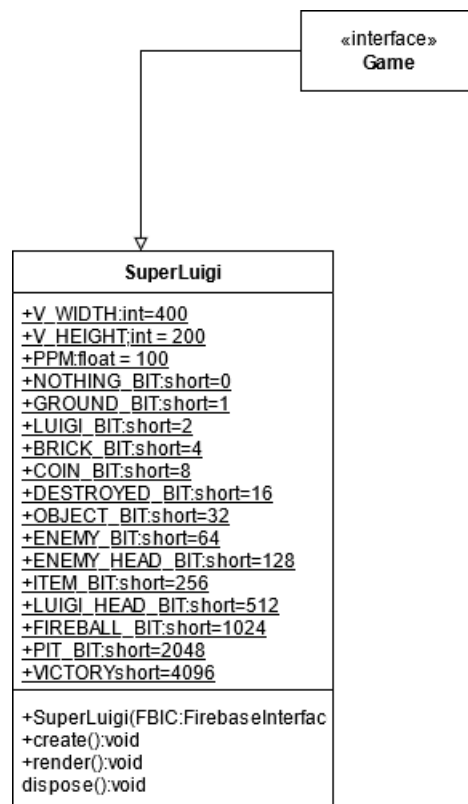


Figure 19: Diagrama de classes SuperLuigi

De seguida, é apresentado, como está organizado o package que corresponde, às sprites, das diferentes personagens, assim como as funcionalidades da moeda e do bloco que se pode partir.

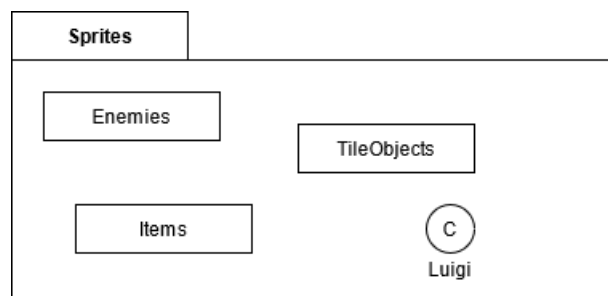


Figure 20: Organização do package Sprites

Na imagem abaixo, podemos verificar, com mais pormenor, as diferentes classes, construídas para os inimigos.

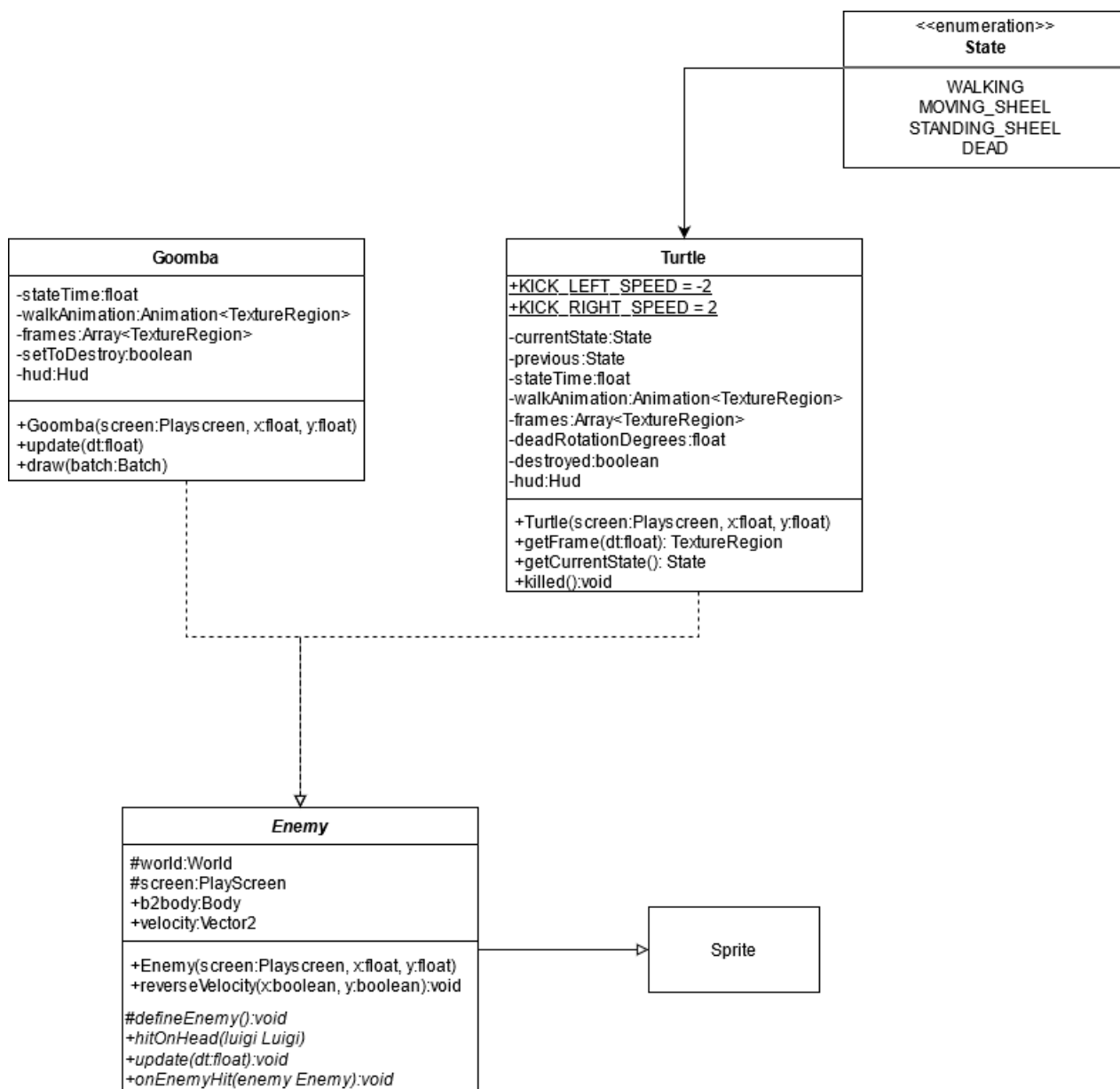


Figure 21: Diagrama de classes do package Enemies

Na imagem abaixo, podemos verificar, com mais pormenor, as diferentes classes, construídas para os diferentes objectos, que foram passados no tiled, durante a construção do mapa.

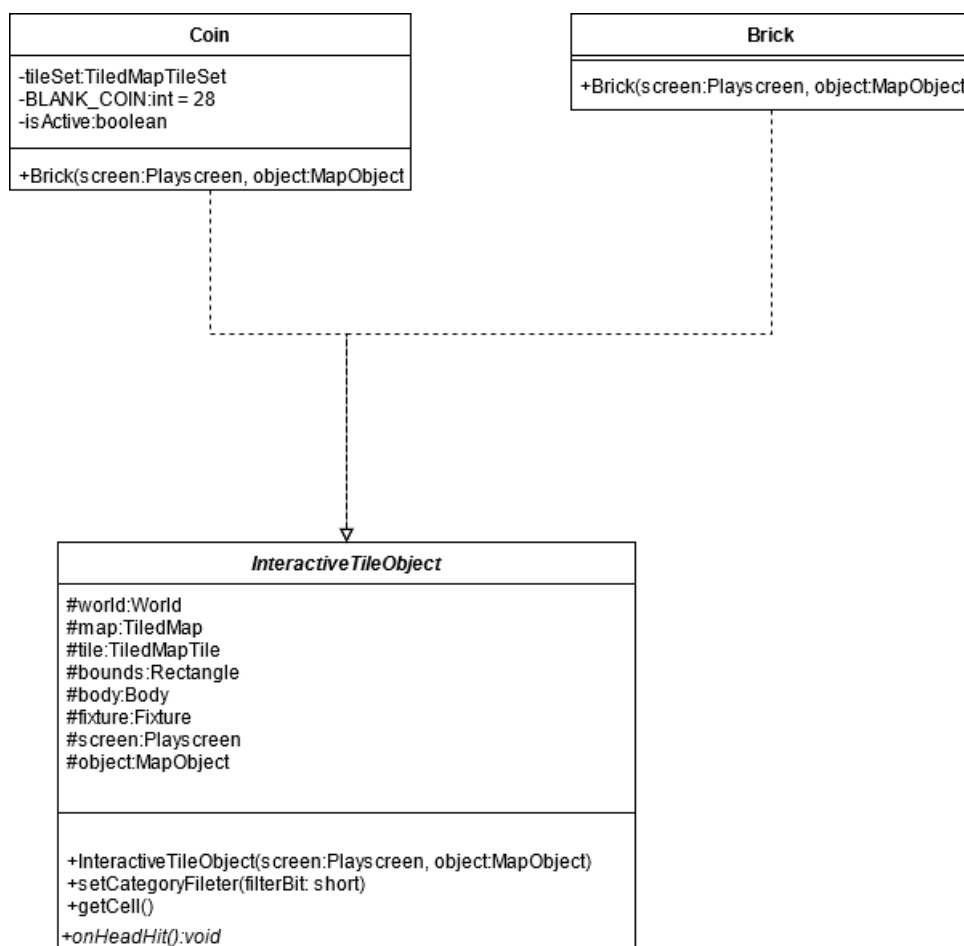


Figure 22: Diagrama de classes do package TiledObjects

Na imagem abaixo, podemos verificar, com mais pormenor, as diferentes classes, construídas para a mecânica do cogumelo.

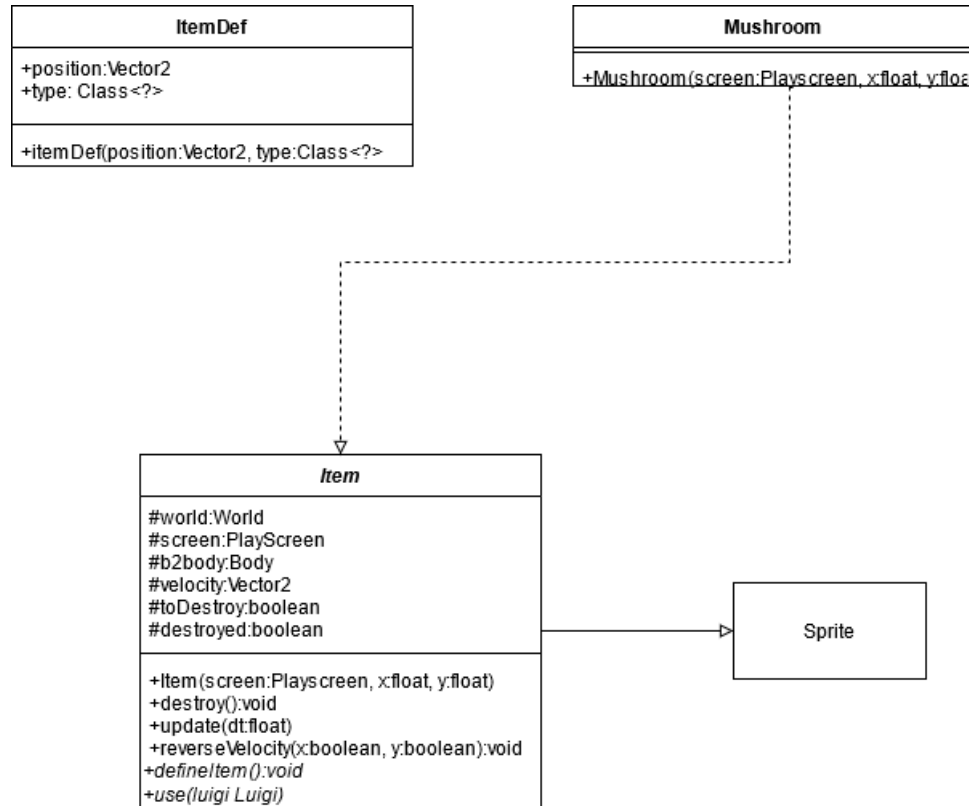


Figure 23: Diagrama de classes do package items

Por último, é apresentada a classe, onde é feita a lógica da personagem principal, mais em concreto, verificar se o jogador morreu, se comeu um cogumelo para crescer, se saltou e foi atingido por um inimigo.

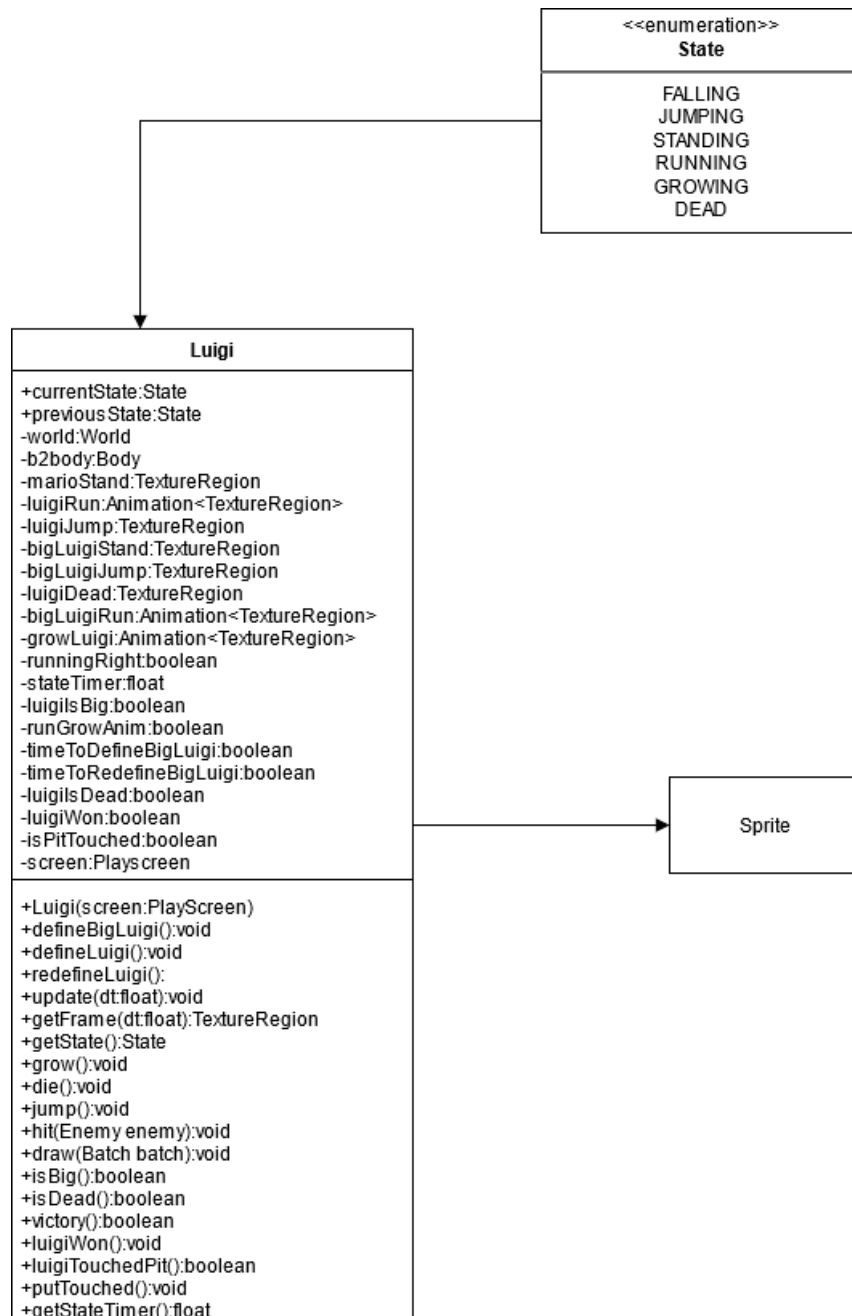


Figure 24: Diagrama de classes da classe Luigi

Para finalizar a parte que corresponde aos diagramas, é apresentada a imagem, como o projeto está organizado, nos diferentes packages.

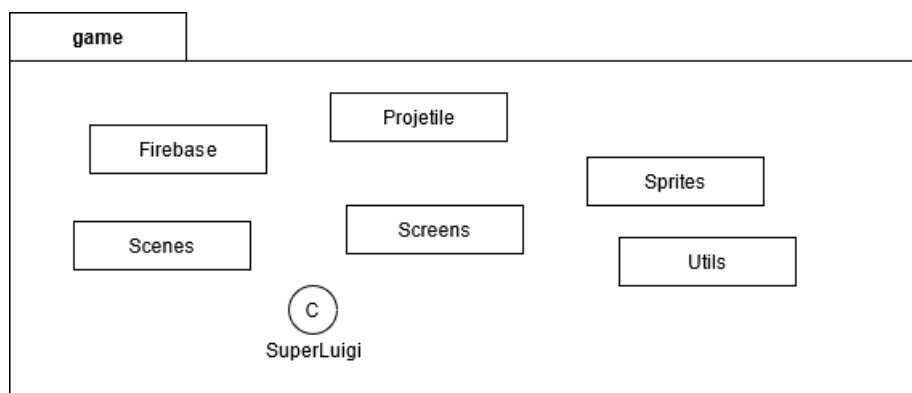


Figure 25: Organização do projeto



## 4.9 Inimigos/Inteligência Artificial

Acerca dos inimigos presentes no jogo, vão existir dois tipos. Um deles será uma espécie de cogumelo andante de nome Goomba. Como foi necessário inserir o tema do covid, recriou-se este inimigo, associando mutações ao inimigo, ficando com outra cor, e outro aspeto.

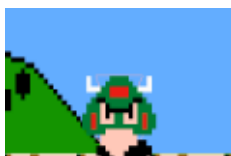


Figure 26: Inimigo gomba com mutações

O outro inimigo consiste numa tartaruga. Visto que, este também sofreu mutações, surge com uma cor vermelha, e não consegue morrer.



Figure 27: Inimigo tartura

No entanto, o facto de nunca morrer, o jogador pode usufruir da sua carapaça para matar inimigos, quando o jogador salta por cima dele.



Figure 28: Inimigo tartaruga, depois do jogador ter saltado por cima

#### 4.10 Personagem Principal

A personagem principal, como já foi referido, foi usada uma sprite, com um conjunto de outras sprites. Na table a seguir, pode-se ver as diferentes ações que o jogador pode efetuar.

Ação	Descrição
Esquerda	Move-se para a esquerda.
Direita	Move-se para a direita.
Saltar	O jogador salta, sendo assim que derrota os inimigos
Comer Cogumelo	Aumenta o tamanho da personagem.

A seguir, é apresentado a sprite, dentro do jogo, da personagem principal que o jogador pode interagir dentro do jogo.



Figure 29: Personagem principal

Na imagem a seguir, podemos ver a personagem, com outro tamanho, quando ingere um cogumelo.



Figure 30: Personagem principal, após ter comido um cogumelo

### 4.11 Interface do jogador

A interface do jogador, irá consistir em três parâmetros:

- No canto da esquerda, irá ter em conta, a pontuação do jogador, sempre que coleciona moedas, ou derrota os inimigos.
- No centro, existe o nome do nível.
- No canto da direita, existe o tempo para completar o nível. Caso o tempo acabe(ou seja, atinga o valor de 0), o jogador morre, reiniciando o nível.



Figure 31: Interface do jogador

### 4.12 Resultado do Jogo

Após os diagramas feitos, as mecânicas implementadas, os inimigos criados, e a interação com o jogador criada, é apresentado, na imagem a seguir o produto final do jogo, contendo todos os elementos referidos neste relatório.

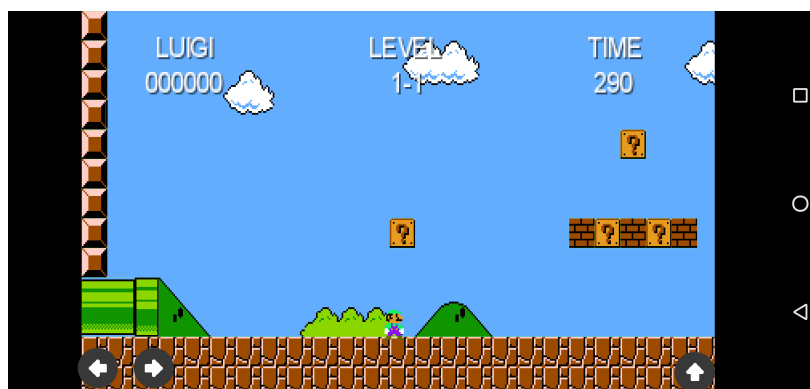


Figure 32: Ecrã Jogo

### 4.13 Extras adicionados

Para além do menu de opções feito como um extra, foi feita a opção de o jogo conter suporte para múltiplos idiomas. Para além do idioma atual que é o português, foi adicionado o idioma de Espanhol e Inglês.

Uma outra opção, foi ter feito o suporte para dispositivos computacionais. Não sendo parte da unidade curricular da disciplina, foi interessante adaptar este jogo, para os dispositivos computacionais. Na imagem abaixo, é possível de ver o jogo, a ser corrido, num ambiente, não android.

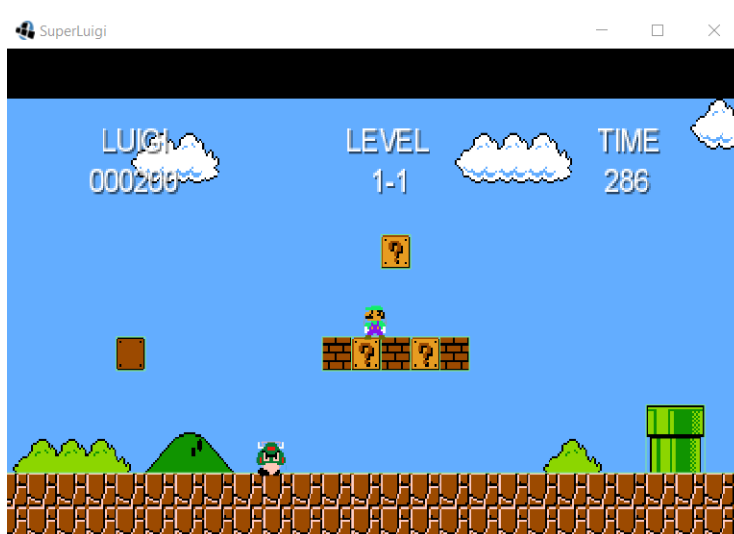


Figure 33: Jogo com suporte a computador

## 5 Conclusões

De modo a recapitular todos os aspetos sobre os quais o projeto realizado, serão referidos os diferentes momentos da implementação.

O correto funcionamento inicial deste projeto é de extrema importância, uma vez que, serviu como "passo inicial" para ter a versão final do jogo pronta, aumentando a complexidade deste jogo.

Após este processo inicial os métodos desenvolvidos focaram-se, em criar interação com o jogador, e recriar os diferentes inimigos, criando mais interação com o jogador.

Através dos diferentes conceitos, como por exemplo, "bodys", "OrthographicCamera", foi possível de recriar este jogo para os dispositivos móveis.

Apesar de grande parte do projeto ter sido realizado com sucesso, surgiram alguns problemas durante a realização do mesmo. Um deles consiste em certos bugs que não possíveis de arranjar. Quando o jogador salta para um bloco, e se manter premido a tecla de salto, a personagem tende a ficar presa, no ar, por baixo do bloco.

Um outro problema consiste em ter efetuado a ligação ao Firebase. O libgdx não suporta o firebase, pelo que, não foi possível efetuar o registo e o login do utilizador, apenas guardar o score de cada nível.

Em termos de implementação, ficou de criar as bolas de fogo. Devido à falta de tempo, não foi possível recriar essa mecânica. Tal facto se deveu em que esta mecânica apresentava certos erros durante a execução.

Apesar de alguns problemas gerados, e de alguns inconvenientes encontrados, foi possível realizar o projeto com sucesso, levando os conhecimentos adquiridos para eventuais projetos futuros.

## 6 Bibliografia

- Ferramenta Tiled: <https://www.mapeditor.org/>
- Libgdx: <https://libgdx.com/>
- Sprite: <https://www.sprisers-resource.com/>
- Firebase: <https://firebase.google.com/>
- Link do questionário: <https://docs.google.com/forms/d/1YKN-9qyqj38g36BXrkiDGiV6a-r07XDmRhq2gub0Uow/edit>