

INSTITUTO POLITÉCNICO DE VIANA DO  
CASTELO

ENGENHARIA INFORMÁTICA

TECNOLOGIA MULTIMÉDIA

---

## Trabalho Phaser - Relatório Final

---

*Autor*

BENJAMIM MARTINS OLIVEIRA

Nº 19160

7 de Junho de 2018



Instituto Politécnico de Viana do Castelo  
Escola Superior  
de Tecnologia  
e Gestão

## Conteúdo

<b>1</b>	<b>Objetivo/Descrição do Jogo</b>	<b>2</b>
<b>2</b>	<b>Funcionalidades Implementadas</b>	<b>2</b>
<b>3</b>	<b>Aspetos Relevantes da implementação</b>	<b>2</b>
3.1	Uso de <i>tilemaps</i> . . . . .	2
3.2	Movimentação dos inimigos . . . . .	4
3.3	Transição entre níveis . . . . .	4
3.4	Coletar moedas/pontos de vida . . . . .	4
<b>4</b>	<b>Manual do Jogo</b>	<b>5</b>
4.1	Iniciar o Jogo . . . . .	5
4.2	Jogar . . . . .	5
4.3	Reiniciar o jogo a partir do ecrã de " <i>Game Over</i> " . . . . .	6
4.4	Reiniciar o jogo após se chegar ao fim do mesmo . . . . .	7
<b>5</b>	<b>Assets usados</b>	<b>7</b>
5.1	Texturas mapa . . . . .	7
5.2	Texturas moeda . . . . .	7
5.3	Inimigos . . . . .	8
5.4	Herói . . . . .	8
5.5	Vidas extra . . . . .	9

## 1 Objetivo/Descrição do Jogo

O jogo é um "*platformer*", com 2 níveis, o objetivo principal será chegar ao fim do ultimo nível, podendo também ser visto como objetivo atingir o máximo de pontuação possível. O primeiro nível possui apenas moedas e inimigos, já o segundo nível possui moedas, inimigos e esferas de vida (cada uma dá 4 pontos de vida ao jogador. É importante referir que os inimigos do segundo nível tiram o dobro da vida do que os do primeiro. O jogador começa com 30 pontos de vida (HP, ou *health points*). A única forma de se perderem estes pontos de vida é colidindo com os inimigos, enquanto o sprite do "herói" estiver em contacto com o sprite inimigo pontos de vida serão descontados ao total de vida.

## 2 Funcionalidades Implementadas

Lista de funcionalidades implementadas:

- Pontuação/moedas coletáveis;
- HP/Pontos de vida coletáveis;
- Níveis com diferente dificuldade;
- Inimigos diferentes;
- Diferentes ecrãs (Menu, Game Over e Fim de Jogo);
- Personagens animadas (tanto herói como inimigos);
- Ecrã de *Loading*;
- Uso de tilemaps;

## 3 Aspetos Relevantes da implementação

### 3.1 Uso de *tilemaps*

Para poder usar um *tilemap* para o jogo primeiro foi necessário criar um, para tal foi usado o programa **Tiled**.

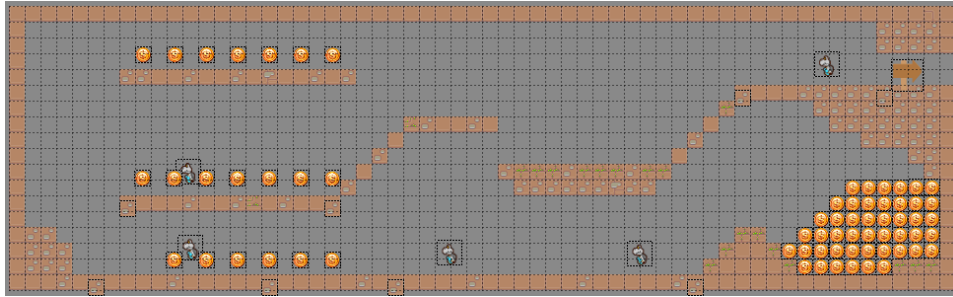


Figura 1: Mapa do primeiro nível.

Para importar o mapa para o jogo usamos a seguinte linha de código (no ficheiro *preload.js*)

```
this.game.load.tilemap('map', 'assets/feat.json', null, Phaser.Tilemap.TILED_JSON);
```

Figura 2: Importar mapa.

Depois de termos feito *preload* do mapa basta desenhar o mesmo no ecrã, para tal usa-se o seguinte:

```
map = game.add.tilemap('map');
map.addTilesetImage('ground_1x1');
layer = map.createLayer('Tile Layer 1');
layer.resizeWorld();
```

Figura 3: Desenhar mapa para o ecrã.

Para desenhar objetos foram usadas as seguintes linhas de código:

```
coins = game.add.group();
coins.enableBody = true;
map.createFromObjects('Object Layer 1', 182, 'coin', 0, true, false, coins);
coins.callAll('animations.add', 'animations', 'spin', [0, 1, 2, 3, 4, 5], 10, true);
coins.callAll('animations.play', 'animations', 'spin');
```

Figura 4: Criação de um objeto(moeda).

### 3.2 Movimentação dos inimigos

Para definir a movimentação dos inimigos foram implementadas "barreiras" no mapa para delimitar a área de atividade de cada inimigo (sendo que estas barreiras não são visíveis ao jogador). Para definir o movimento dos inimigos usa-se o seguinte código:

```
game.physics.arcade.collide(grounds, inimigos, colisaoinimigosLayer, null, this);  
:  
function colisaoinimigosLayer(player, inimigo){  
    if(inimigo.animations.name == "runLeft"){  
        inimigo.body.velocity.x = 50;  
        inimigo.animations.play('runRight');  
    }else{  
        inimigo.body.velocity.x = -50;  
        inimigo.animations.play('runLeft');  
    }  
}
```

Figura 5: Mover inimigo.

### 3.3 Transição entre níveis

Para definir a passagem de nível cria-se uma espécie de portal, um objeto que não é visível. Esse objeto só se encontra num sitio em cada mapa e quando o jogador colide com ele o jogo muda o de nível automaticamente.

```
game.physics.arcade.overlap(fim, sprite, overlapFim, null, this);  
:  
function overlapFim(player, fim){  
    this.game.state.start("nivel_dois");  
}
```

Figura 6: Mudar de nível.

### 3.4 Coletar moedas/pontos de vida

O código usado para coletar moedas e pontos de vida é extremamente parecido, assim sendo será mostrado apenas o código usado para coletar moedas.

```

game.physics.arcade.overlap(sprite, coins, collectCoin, null, this);

function collectCoin(player, coin) {
    coin.kill();
    pontuacao += 1;
    textoPontuacao.text = 'Moedas: ' + pontuacao;
}

```

Figura 7: Coletar moeda.

## 4 Manual do Jogo

### 4.1 Iniciar o Jogo

Para iniciar o jogo deve-se clicar no botão verde **GO**;

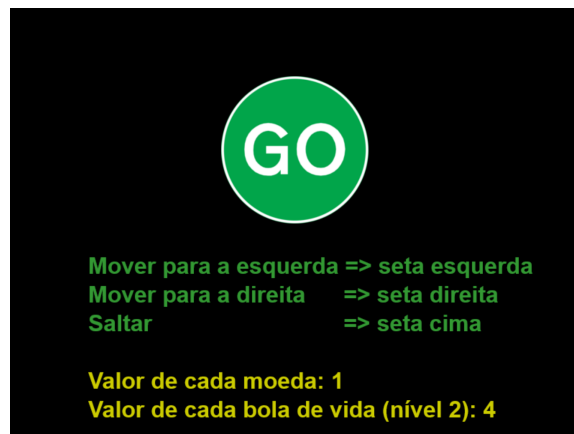


Figura 8: Iniciar jogo.

### 4.2 Jogar

Controlos de jogo:

- Saltar: Seta para cima;
- Andar para a esquerda: Seta para a esquerda;

- Andar para a direita: Seta para a direita;

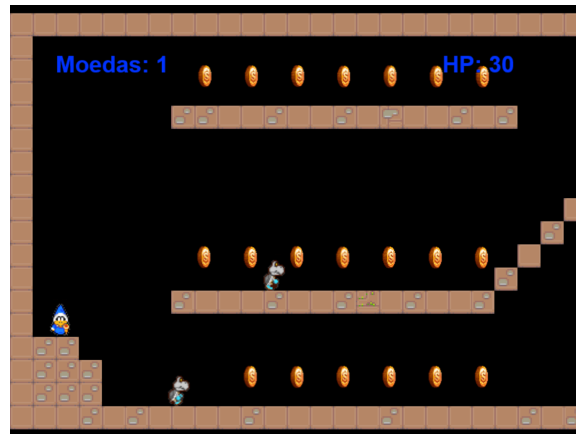


Figura 9: Jogo.

#### 4.3 Reiniciar o jogo a partir do ecrã de "*Game Over*"

Para reiniciar o jogo após perder toda a vida deve-se carregar no botão **GAME OVER**;



Figura 10: Game Over.

#### 4.4 Reiniciar o jogo após se chegar ao fim do mesmo

Para reiniciar o jogo após terminar o mesmo deve-se carregar no botão verde;

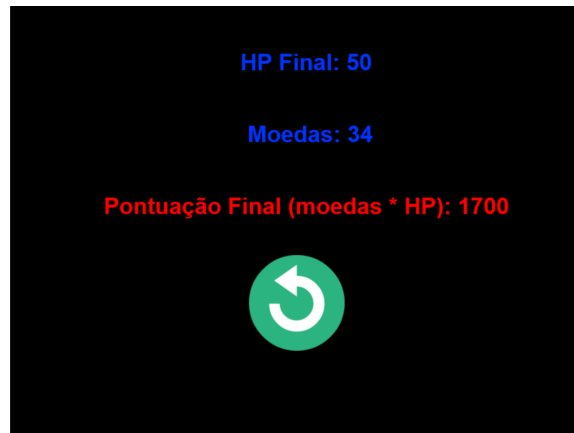


Figura 11: Fim do jogo.

## 5 Assets usados

### 5.1 Texturas mapa

Origem: *phaser.io*;



Figura 12: Texturas mapa.

### 5.2 Texturas moeda

Origem: *phaser.io*;





Figura 13: Texturas moeda.

### 5.3 Inimigos

Origem: <https://retrospriteresources.deviantart.com/art/Mario-Enemies-3-RPG-Maker-MV-635631029>



Figura 14: Inimigos.

### 5.4 Herói

Origem: <https://retrospriteresources.deviantart.com/art/Mario-Enemies-3-RPG-Maker-MV-635631029>



Figura 15: Herói.

## 5.5 Vidas extra

Origem: <http://ryanbown.blogspot.com/2010/11/>;



Figura 16: Vidas extra.