

**Trabalho n.º 1:** FEUP Manic Miner - DJCO

*Relatório de implementação*

Desenvolvimento de Jogos de Computador

4.º ano do Mestrado Integrado em Engenharia Informática e Computação

Elementos do Grupo:

Avner Pereira – 201208310 – [ext12289@fe.up.pt](mailto:ext12289@fe.up.pt)

Carlos Frias – 040509116 – [ei04116@fe.up.pt](mailto:ei04116@fe.up.pt)

Francisco Cabelo – 201202630 – [ext12104@fe.up.pt](mailto:ext12104@fe.up.pt)

Porto, 12 de abril de 2013

## Introdução

Com a elaboração deste trabalho pretendeu-se recriar um nível do famoso jogo de plataformas Manic Miner, introduzindo algumas inovações e relacionando o jogo com a Faculdade de Engenharia da Universidade do Porto.

O objetivo do jogo é o de percorrer as plataformas e recolher as “chaves” dentro de um limite de tempo e sem colidir contra os inimigos e, após estarem todas elas recolhidas, regressar e entrar no “goal” do nível. Para as “chaves” escolhemos as quatro letras da palavra FEUP.

Em adição às características originais do jogo acrescentamos, essencialmente, mais duas inovações, que são: A rotação da tela de jogo em noventa graus sempre que o jogador recolhe uma das chaves. Isto dificulta um pouco o jogo e obriga a um planeamento da sequência de recolha das chaves; A segunda inovação é o fato de ser permitido matar os inimigos lançando-lhes uma bola que o jogador pode recolher. Esta funcionalidade baixa o grau de dificuldade do jogo.

O jogador tem inicialmente três vidas que terá de utilizar para passar o nível e um score que vai aumentando com o recolher das “chaves” e com o tempo e número de vidas que não desperdiçado. Essa pontuação diminui com o número de vidas desperdiçadas e com o número de inimigos mortos.

O jogo foi desenvolvido em ambiente Windows com o motor de jogo Unity 4, utilizando c# como linguagem de script e a framework Orthello 2D que previdência ferramentas para desenvolvimento de jogos em duas dimensões.

# Implementação

## Snapshots do jogo

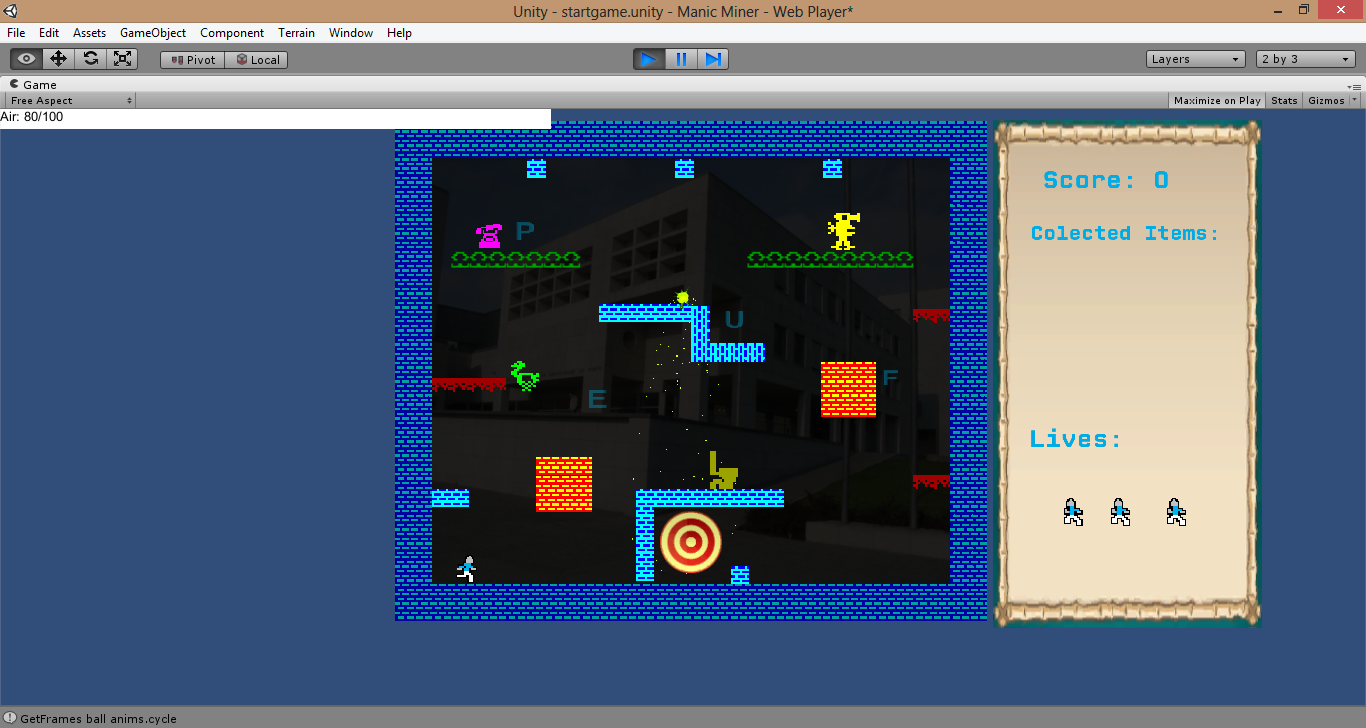


Figura 1: Snapshot do início do nível de jogo.

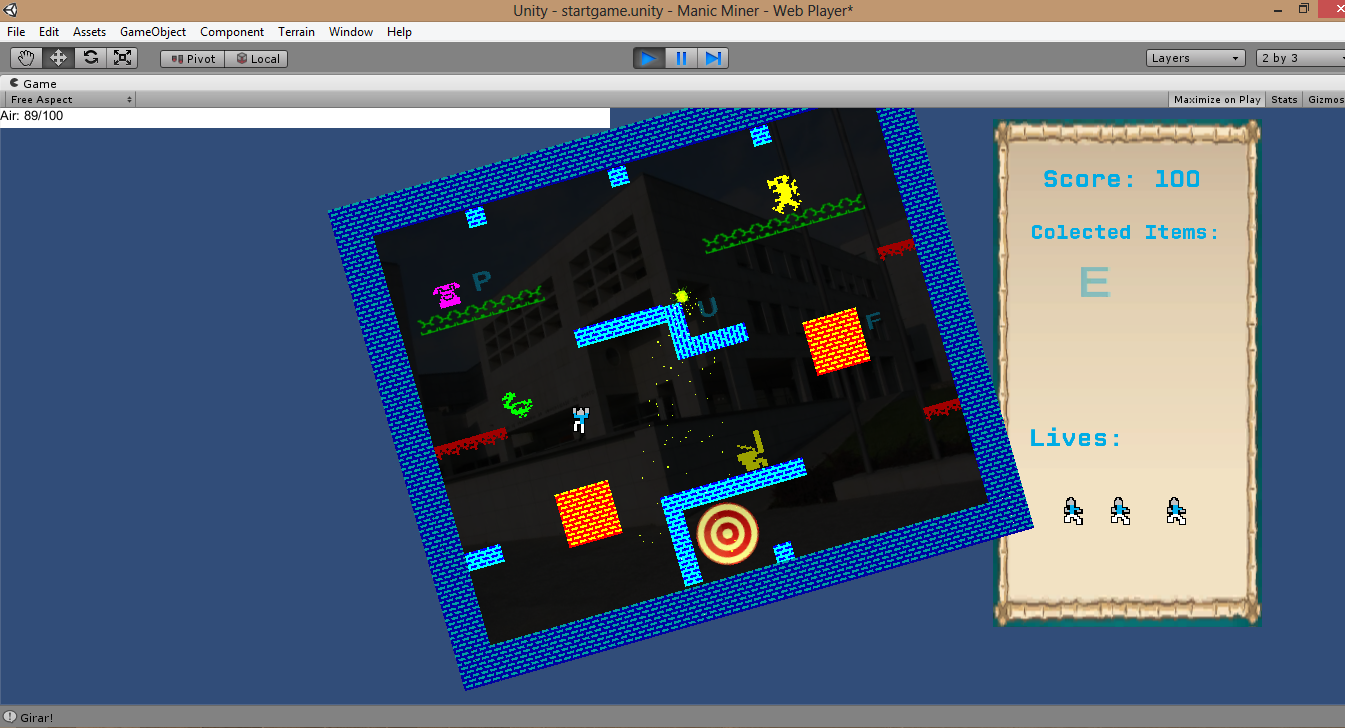


Figura 2: Snapshot da rotação da tela de jogo.

## Funcionalidades implementadas

O jogo possui as seguintes funcionalidades:

1. Movimentar o jogador para a esquerda e para a direita ao longo das plataformas do jogo utilizando as teclas direcionais;
2. Saltar obstáculos (salto normal e salto duplo), recolher e lançar uma bola contra os adversários e cenário, utilizando também as teclas direcionais, nomeadamente as teclas “cima” e “baixo;
3. Rotacionar num ângulo de noventa graus toda a tela, em relação ao seu centro, sempre que se recolha uma das “chaves”;
4. Matar os inimigos lançando-lhes uma bola;
5. Duas plataformas quadradas móveis, uma com movimentos horizontais e a outra com movimentos verticais;
6. Duas plataformas do tipo passadeira, tal como existe no jogo original, em que o jogador é arrastado numa direção a uma determinada velocidade;
7. Algumas plataformas instáveis, tal como as do jogo original, que caem passado alguns instantes após o jogador as ter pisado;
8. Inimigos móveis e estáticos os quais provocam a perda de uma vida ao jogador caso os toque;
9. Uma barra de tempo que vai diminuindo e que determina obviamente o tempo que o jogador tem para terminar o nível. Caso o tempo chegue ao fim o jogador perde;
10. Um display lateral onde é apresentado o score do jogador, quais os itens recolhidos até ao momento e o número de vidas disponíveis;
11. Um ecrã inicial do jogo, apresentando o nome do jogo e a opção de iniciá-lo e dois ecrãs para fim do jogo, um para o caso do jogador ganhar e outro para o caso do jogador perder.

## Arquitetura da aplicação

O projeto está organizado da seguinte forma: n pasta Assets contém os seguintes diretórios:

1. Animations: onde estão guardadas todas as animações criadas, nomeadamente, as animações dos inimigos, as animações das plataformas móveis e a animação das plataformas instáveis;
2. Audio: onde estão guardados os ficheiros de som utilizados no jogo;
3. Font: que armazena os ficheiros relativos aos tipos de letra utilizados;
4. Materials: onde estão guardados os materiais criados, neste caso apenas o material para a bola;
5. Orthello: armazena os ficheiros relativos à framework Orthello 2D;
6. Prefabs: armazena animações “pré-fabricadas”, no caso apenas as partículas que são lançadas pela bola;
7. Scenes: que guarda as cenas do jogo, nomeadamente, a cena inicial, o único nível do jogo e as duas cenas de finais;
8. Script: onde estão armazenados os srcipts c# do jogo;
9. SpriteAtlases: onde estão arquivadas as spritesAtlases para as texturas apresentadas no jogo;
10. Sprites: que guarda as imagens iniciais para as texturas de todos os objetos que irão surgir no jogo

# Conclusões

A realização deste trabalho permitiu desenvolver os nossos conhecimentos acerca do desenvolvimento de jogos de computador e sua mecânica, permitiu a familiarização com o motor de jogos Unity 4 e a utilização de linguagem de script em c#.

O desenvolvimento do produto decorreu de uma forma fluída, sem dificuldades de maior, com a plena colaboração de todos os elementos do grupo.

Desenvolvemos um produto estável e pensamos ter cumprido os objetivos propostos para o trabalho.