

# Faculdade de Engenharia de Universidade do Porto



## FEUP

*RAIDEN*

João Pedro Milano Silva Cardoso  
200900579 - ee09063@fe.up.pt

Relatório realizado no âmbito do Mestrado Integrado de Informática e Computação  
Laboratório de Programação Orientada por Objectos

9 de Junho de 2014

# Índice de Conteúdos

Índice de Ilustrações.....	3
Introdução.....	4
Controlos.....	5
Regras do Jogo .....	5
Arranque do programa.....	5
GUI – Graphical User Interface.....	6
Menu Principal .....	6
Menu de Opções .....	7
Ecrã de Jogo.....	8
Menu de Pausa.....	9
Menu de Ajuda.....	10
Pontuações.....	11
Game Over .....	12
Design Patterns .....	13
State .....	13
Game Loop .....	13
Update Method.....	13
Template Method .....	13
Model-View-Controller .....	14
Diagramas UML.....	14
Package Diagram .....	14
Controller Diagram.....	15
Test Diagram .....	15
GUI Diagram .....	16
Máquina de Estados da Aplicação.....	17
Logic Diagram .....	18
Conclusão .....	19

## Índice de Ilustrações

Ilustração 1 - Menu Principal .....	6
Ilustração 2 - Menu de Opções .....	7
Ilustração 3 - Ecrã de Jogo.....	8
Ilustração 4 - Menu de Pausa .....	9
Ilustração 5 - Menu de Ajuda .....	10
Ilustração 6 - Pontuações .....	11
Ilustração 7 - Ecrã de Game Over.....	12
Ilustração 8 - Diagrama Package .....	14
Ilustração 9 - Diagrama Controller .....	15
Ilustração 10 - Diagrama de Testes .....	15
Ilustração 11 - Diagrama GUI .....	16
Ilustração 12 - Máquina de Estados da Aplicação.....	17
Ilustração 13 - Diagrama Lógica .....	18

## Introdução

O relatório apresentado foi realizado no contexto do 2º trabalho prático da Unidade Curricular de Laboratório de Programação Orientada por Objectos.

Neste trabalho, pretende-se a criação de um jogo estilo Raiden, um top-down shooter vertical, o qual tem interacção com teclado e rato.

O objectivo do trabalho é lidar com animação, através da aplicação e jogo principal e de ficheiros, através da pontuação.

No presente relatório encontram-se as informações pertinentes à aplicação – os controlos do jogo, as regras do jogo, explicação dos vários menus e ecrãs com os quais é possível interagir, os design patterns usados, os vários diagramas de classes.

## Controlos

Para movimentar a nave são usadas as teclas WASD, para movimentar para cima, esquerda, baixo e direita, respectivamente.

Para disparar é usado o botão esquerdo do rato e para mudar o modo da nave é usado o botão direito.

Para parar o jogo é usada a tecla P.

A interacção com os menus é feita usando o rato, o qual pode interagir com os botões, os quais estão delimitados por um rectângulo.

## Regras do Jogo

O jogador começa com 100 pontos de vida e escudos.

À medida que o jogo corre são criados inimigos no topo do ecrã, os quais se movem em direcção ao fundo do ecrã segundo um de dois padrões de movimento, em movimento vertical ou em “zig-zag”.

O jogador, enquanto azul, consegue destruir inimigos vermelhos, os quais são capazes de danificar o jogador. Enquanto azul, o jogador não é afectado por balas azuis.

O jogador, enquanto vermelho, consegue destruir inimigos azuis, os quais são capazes de danificar o jogador. Enquanto vermelho, o jogador não é afectado por balas vermelhas.

Se um jogador, independentemente da forma, colidir com um inimigo, perde vida.

Sempre que um inimigo é destruído, a pontuação aumenta em 100 pontos.

O jogo não tem condições de vitória.

O objectivo é sobreviver e alcançar a pontuação mais alta possível.

## Arranque do programa

Não é necessária nenhuma instalação ou biblioteca adicional. Todos os recursos estão incluídos no projecto. A class Raiden tem o main() da aplicação. Esta também pode ser iniciada através do ficheiro Raiden.jar.

## GUI – Graphical User Interface

### Menu Principal



Ilustração 1 - Menu Principal

Ao iniciar a aplicação, o menu principal é o primeiro ecrã a aparecer.

Enquanto neste menu, o jogador pode:

- Iniciar o jogo, ao clicar em “PLAY”.
- Aceder ao Menu de Opções ao clicar em “OPTIONS”.
- Aceder ao Menu de Ajuda ao clicar em “HELP”.
- Sair da aplicação ao clicar em “EXIT”.

## Menu de Opções

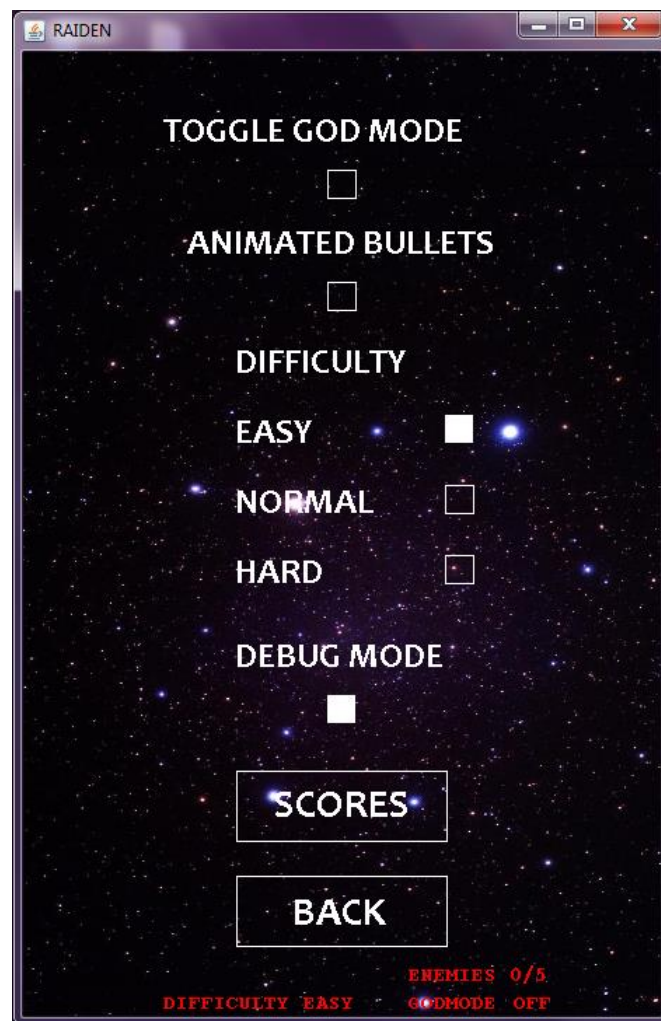


Ilustração 2 - Menu de Opções

O Menu de Opções pode ser acedido a partir do Menu Principal.

As opções disponíveis a jogador são:

- Toggle God Mode, que permite ao jogador ser invulnerável.
- Animated Bullets, que permite ao jogador escolher se as balas são animadas ou não.
- Difficulty, que permite ao jogador escolher entre três níveis de dificuldade, fácil, normal ou difícil.
- Debug Mode, que imprime no ecrã informações relativas ao jogo a vermelho, a dificuldade, o número de inimigos e o estado de Godmode.
- Score, que imprime no ecrã as 10 melhores pontuações.

## Ecrã de Jogo

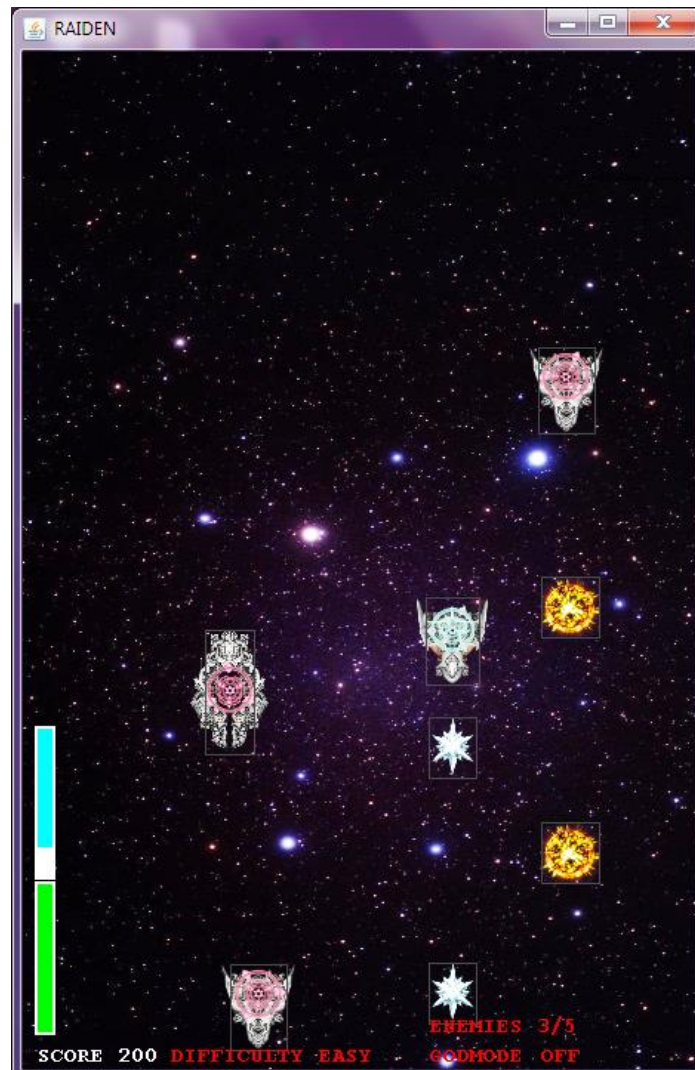


Ilustração 3 - Ecrã de Jogo

O ecrã de jogo pode ser acedido a partir do Menu Principal.

Neste ecrã decorre a acção principal do jogo.

Do lado esquerdo do ecrã estão representados a barra de vida, a verde, e a barra de escudos, a azul.

Por baixo é pintada a pontuação.



## Menu de Pausa

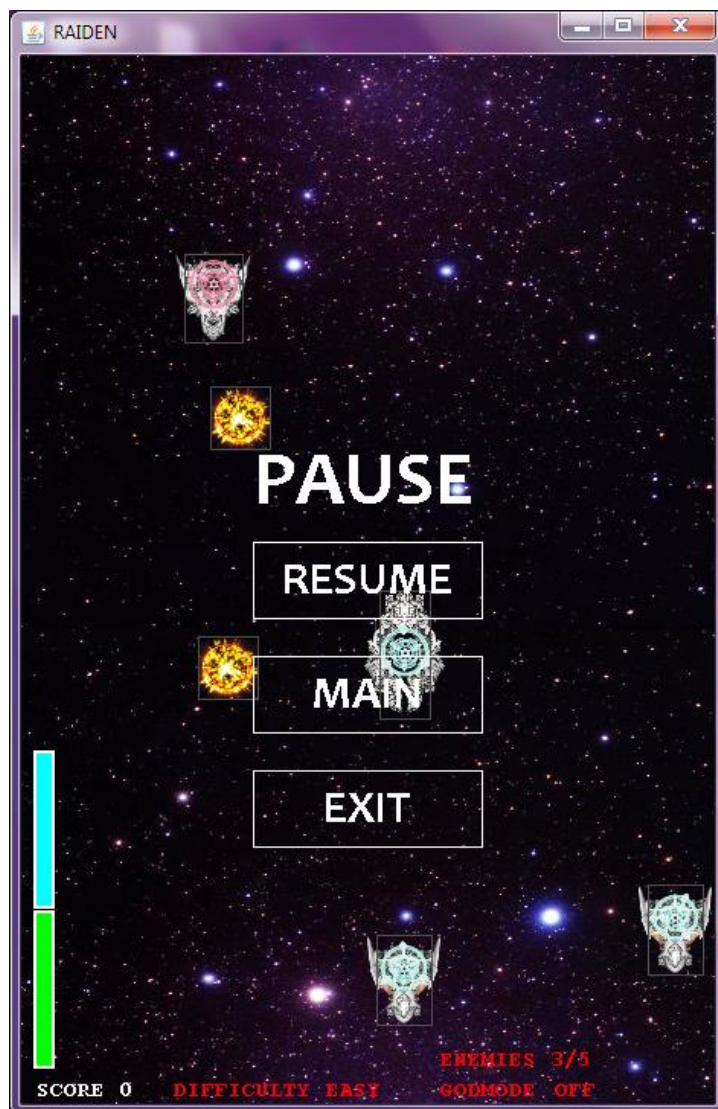


Ilustração 4 - Menu de Pausa

O menu de Pause pode ser acedido a partir do ecrã de jogo ao carregar na tecla P.

As opções disponíveis ao jogador são:

- Resume, que permite ao jogador continuar o jogo.
- Main, que permite ao jogador voltar ao Menu Principal.
- Exit, que permite ao jogador sair da aplicação.

## Menu de Ajuda

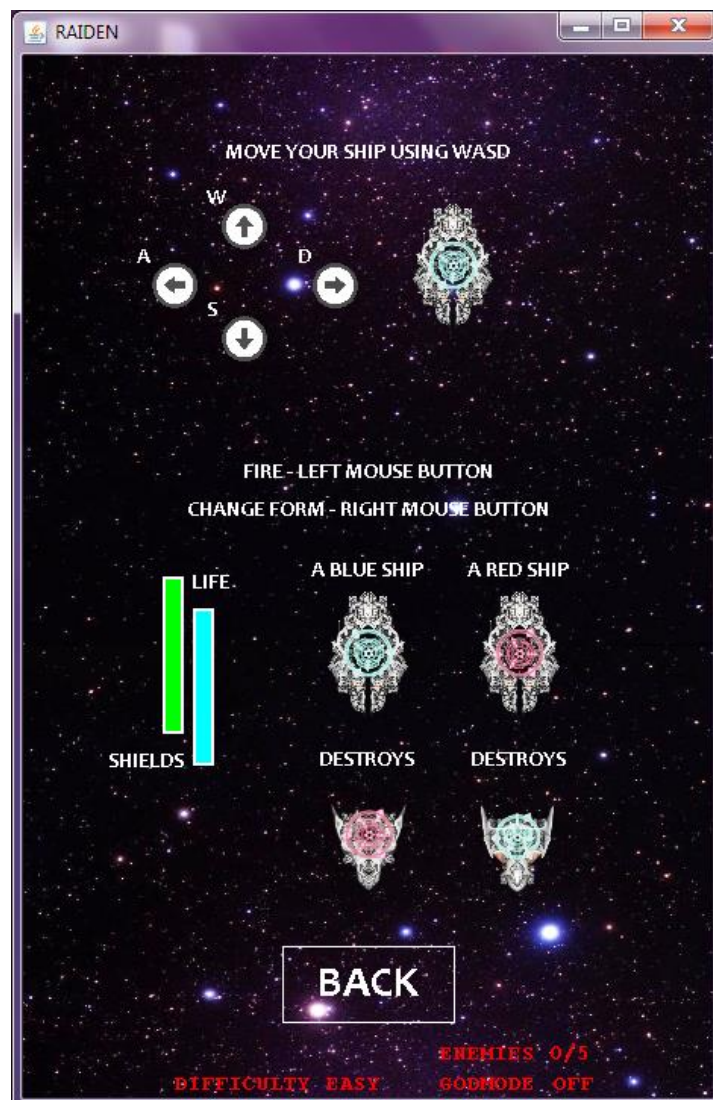


Ilustração 5 - Menu de Ajuda

O Menu de Ajuda pode ser acedido a partir do Menu Principal.

Este ecrã explica os comandos e interacção entre naves.

O botão Back permite voltar ao Menu Principal.

## Pontuações



Ilustração 6 - Pontuações

O ecrã de pontuações pode ser acedido a partir do Menu de Opções.

As dez melhores pontuações são impressas no ecrã.

O botão Back permite voltar ao Menu de Opções.



## Game Over

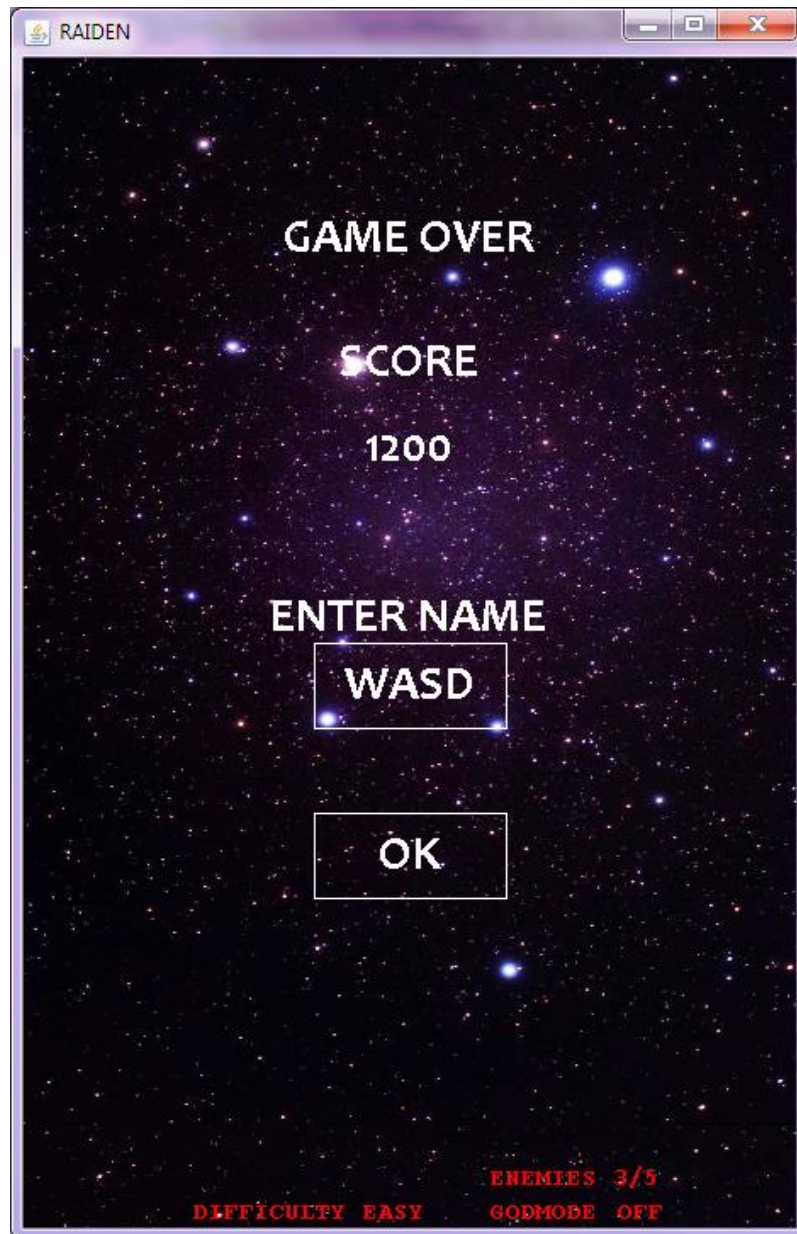


Ilustração 7 - Ecrã de Game Over

Quando ocorre game over o ecrã de game over aparece.

Aqui o jogador pode introduzir o seu nome, o qual será adicionado ao ficheiro de pontuações, score.txt. O nome pode ter um máximo de 5 caracteres.

# Design Patterns

## State

Para o programa saber que menu ou ecrã tem de pintar e para o controlador saber quais são as acções permitidas, são utilizados estados para navegar mais facilmente dentro da aplicação. A vida da própria aplicação pode então ser representada por um máquina de estados.

Os estados são:

- Menu, indica que está no Menu Principal
- Game, indica que está no ecrã de jogo
- Help, indica que está no ecrã de ajuda
- Pause, indica que está no menu de Pausa
- GameOver, indica que está no menu Game Over
- ShowScores, indica que está no ecrã de pontuações
- Options, indica que está no Menu de Opções

## Game Loop

A aplicação cria um thread que actualiza constantemente os elementos do jogo, sem ser interrompido pelos inputs. Este loop actualiza o jogo, e pede para pintar os elementos. O loop encontra-se na classe MainTh, que cria o thread principal.

## Update Method

Cada objecto tem um método update, que especifica como ele se comporta por ciclo, de modo a ter a menor quantidade de código possível no ciclo. Deste modo é também mais fácil alterar o comportamento de cada objecto. Os objectos que são actualizados são as balas, a nave do jogador e as naves inimigas.

## Template Method

De modo a poder pintar os vários ecrãs, o melhor modo de o fazer é fazendo override o método paintComponent, o que é possível devido ao Template Method.

## Model-View-Controller

Para uma melhor gestão do programa as classes estão separadas segundo a sua posição no modelo MVC. Estão então separadas em GUI, o qual pede informação a Logic, que gere a lógica do jogo e os seus elementos, e Controller, que actualiza elementos em Logic, de modo a que o ciclo detecte os comportamento a tomar.

## Diagramas UML

### Package Diagram

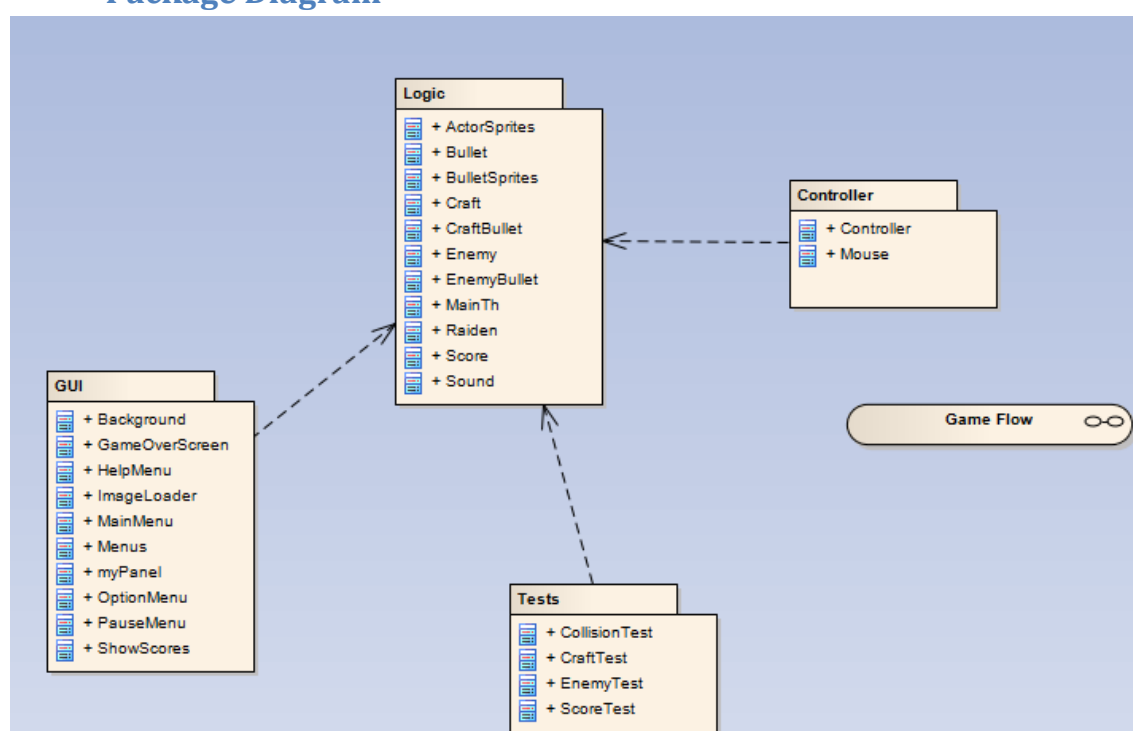


Ilustração 8 - Diagrama Package

Package	Responsabilidade
Logic	Lógica Interna da aplicação
GUI	Visualização da aplicação
Controller	Interacção com o utilizador
Tests	Testes

## Controller Diagram

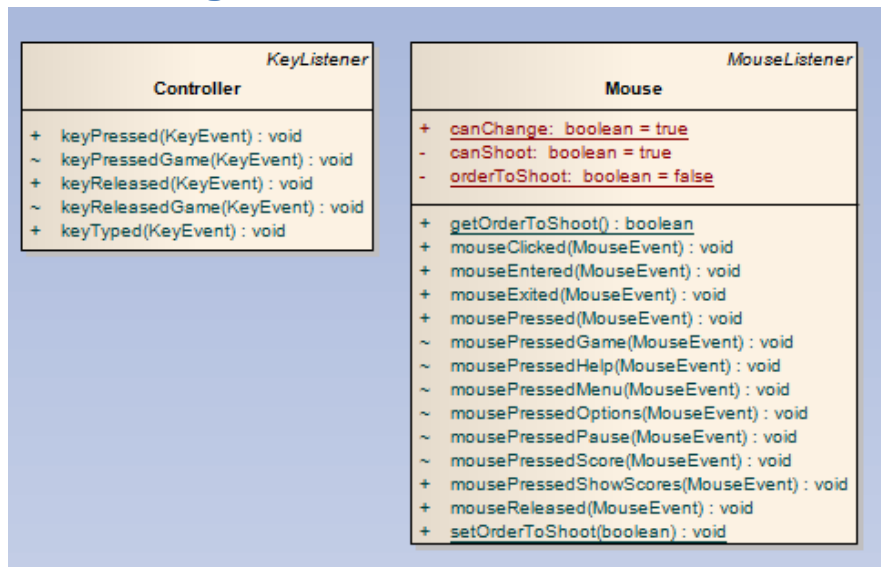


Ilustração 9 - Diagrama Controller

Classe	Responsabilidade
<b>Controller</b>	Interação do jogador com o teclado
<b>Mouse</b>	Interação do jogador com o rato

## Test Diagram

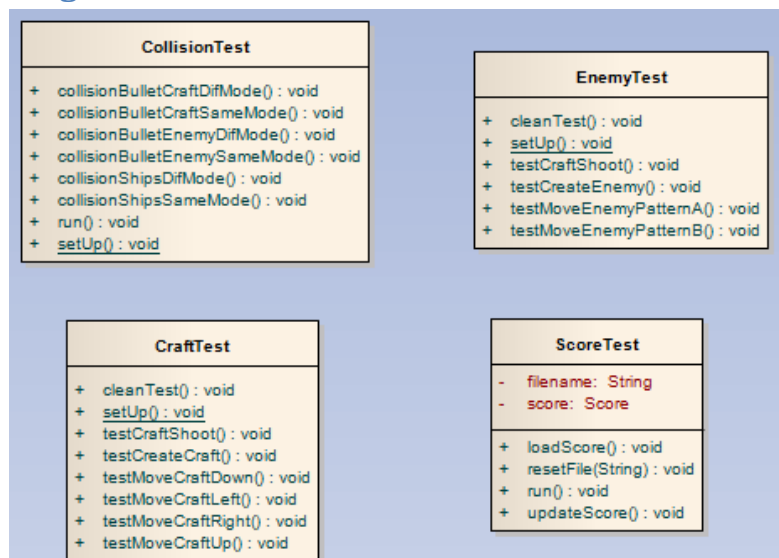


Ilustração 10 - Diagrama de Testes

Classe	Responsabilidade
<b>CollisionTest</b>	Testes de Colisão entre elementos
<b>ScoreTest</b>	Testes de leitura e actualização de pontuação
<b>CraftTest</b>	Testes de criação e movimento do jogador
<b>EnemyTest</b>	Testes de criação e movimento do inimigo

## GUI Diagram

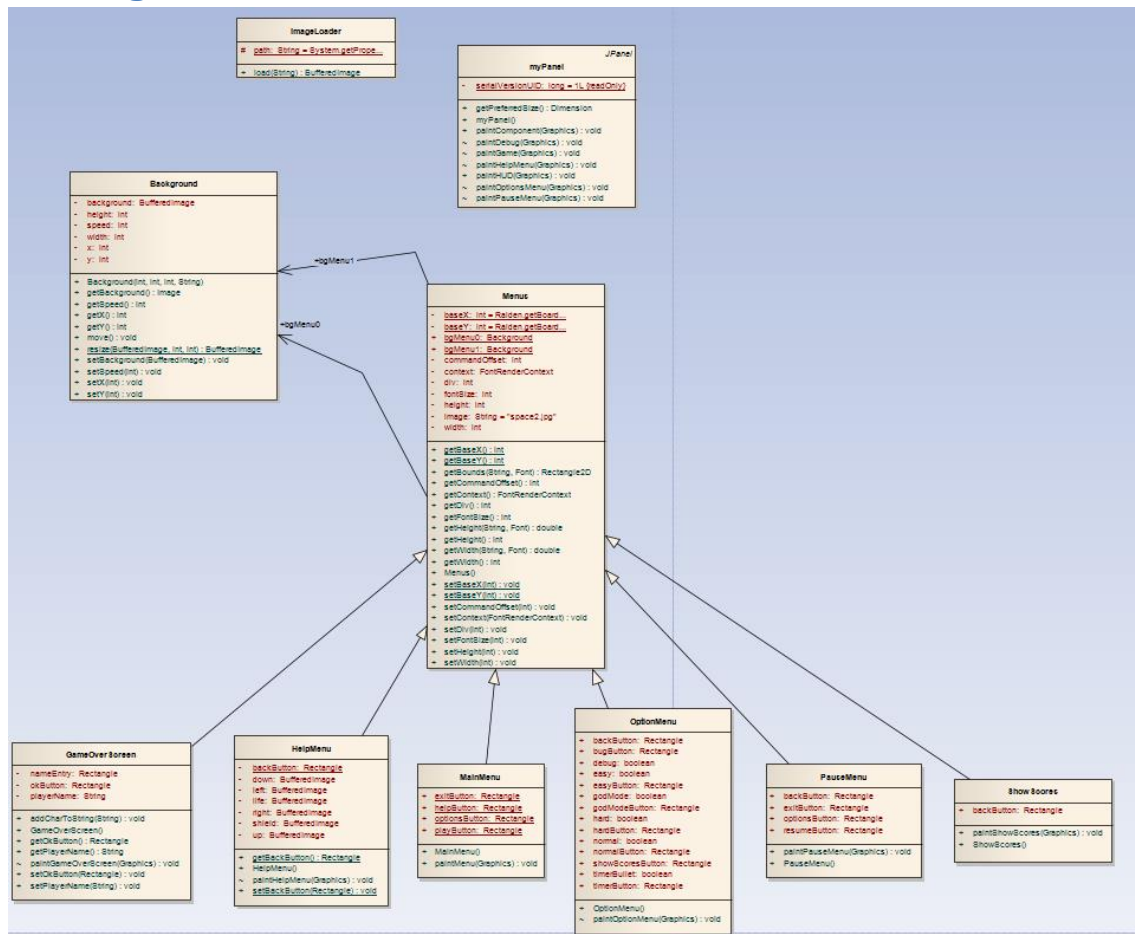


Ilustração 11 - Diagrama GUI

Classe	Responsabilidade
<b>ImageLoader</b>	Carregar Imagens
<b>myPanel</b>	Pintar os elementos
<b>Background</b>	Criação do background
<b>Menus</b>	Superclasse dos vários Menus
<b>GameOverScreen</b>	Introdução do nome do jogador e actualização da pontuação
<b>HelpMenu</b>	Mostrar ecrã de ajuda
<b>MainMenu</b>	Menu Principal
<b>OptionMenu</b>	Menu de Opções
<b>PauseMenu</b>	Menu de Pausa do Jogo
<b>ShowScores</b>	Mostra as dez melhores pontuações



## Máquina de Estados da Aplicação

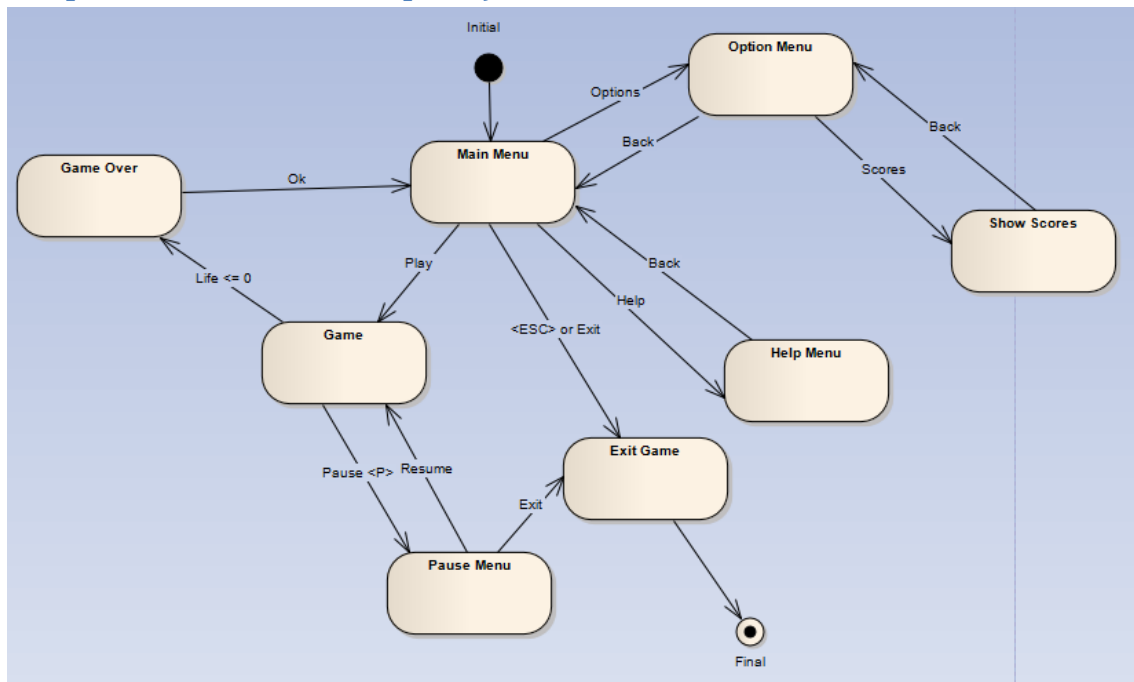


Ilustração 12 - Máquina de Estados da Aplicação

Estado	Significado
<b>Main Menu</b>	A aplicação encontra-se no Menu Principal
<b>Option Menu</b>	A aplicação encontra-se no Menu de Opções
<b>ShowScores</b>	A aplicação encontra-se a Mostar as Pontuações
<b>Game</b>	A aplicação encontra-se no Jogo
<b>Pause Menu</b>	A aplicação encontra-se no Menu de Pausa
<b>Game Over</b>	A aplicação encontra-se no Ecrã de Game Over
<b>Help Menu</b>	A aplicação encontra-se no Menu de Ajuda
<b>Exit Game</b>	A aplicação vai encerrar

## Logic Diagram

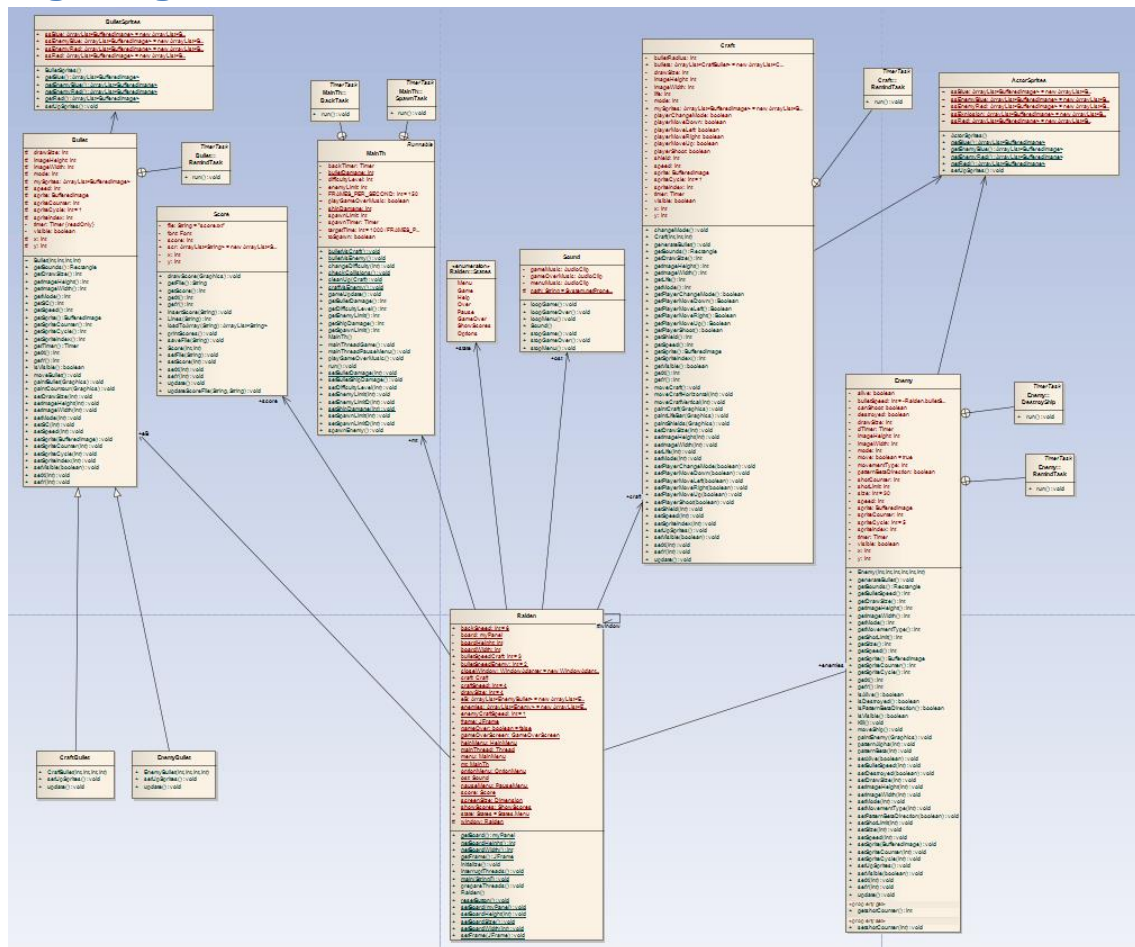


Ilustração 13 - Diagrama Lógica

Classe	Responsabilidade
<b>BulletSprites</b>	Data Class com os sprites para as balas
<b>ActorSprites</b>	Data Class com os sprites para as naves
<b>Raiden</b>	Class Principal com inicializações
<b>Bullet</b>	Superclasse das balas
<b>CraftBullet</b>	Criação/Update das balas do jogador
<b>EnemyBullet</b>	Criação/Update das balas dos inimi
<b>Craft</b>	Criação/Update da nave do jogador
<b>Enemy</b>	Criação/Update dos inimigos
<b>MainTh</b>	Criação do thread principal com os loop do jogo. Contém a verificação de colisões e as funções usadas para o update do jogo
<b>Score</b>	Leitura/Impressão/Gestão das pontuações
<b>Sound</b>	Carrega banda sonora

## Conclusão

Os objectivos – lidar com animação e ficheiros – foram cumpridos.

Algumas melhorias para o projecto poderiam ser:

- Organização dos objectos de jogo de modo a melhorar o cálculo de colisões, até mesmo diminuir o número de objectos a processar, através de organização tendo em conta a distância ao jogador ou por quadrantes.
- Inclusão de um boss final, que seria encontrado em certa altura do jogo, de modo a existir uma condição de vitória.
- Inclusão de mais padrões de movimento para os inimigos.
- Inclusão de power-ups, que poderiam restaurar vida ou escudos, ou alterar como a nave do jogador dispara.