



# **Faculdade de Engenharia da Universidade do Porto**

## **Laboratório de Programação Orientada por Objetos**

### **Snake**

Mestrado Integrado em Engenharia Informática e Computação

- Diogo Miguel Raposo Soares, 201203859, ei12102@fe.up.pt
- João Miguel Falcão Morgado, 201203920, ei12055@fe.up.pt

9 de junho de 2014

# Índice

Índice .....	1
1. Introdução .....	2
1.1. Objetivo do Relatório .....	2
1.2. Objetivo do Programa .....	2
1.3. Estrutura do Relatório .....	2
2. Manual de Utilização.....	3
2.1. Resumo das Funcionalidades Suportadas .....	3
2.2. Arranque do Programa .....	3
2.3. Modo de utilização .....	4
3. Conceção e Implementação .....	8
3.1. Estrutura da Packages .....	8
3.2. Estrutura de Classes .....	9
3.3. Padrões de Desenho .....	12
3.4. Mecanismos Importantes .....	12
4. Conclusão .....	13
Referências.....	14

# 1. Introdução

## 1.1. Objetivo do Relatório

Com a realização deste relatório pretendemos explicar o funcionamento do jogo que desenvolvemos bem como métodos usados no seu desenvolvimento e diagramas que facilitam a sua análise.

## 1.2. Objetivo do Programa

O objetivo do jogo “Snake” é aguentar uma cobra, cujo tamanho vai aumentando, o máximo de tempo possível, dentro de um tabuleiro, sem colidir contra nenhuma parede (se estas existirem), contra o seu próprio corpo ou contra bombas.

## 1.3. Estrutura do Relatório

Este relatório é constituído pelo Manual de Utilização, onde são explicados os aspetos relacionados com o decorrer do jogo e os seus modos. De seguida vem a Conceção e Implementação, onde se encontram os diagramas UML que resumem a estrutura e o modo de funcionamento do jogo.

## 2. Manual de Utilização

### 2.1. Resumo das Funcionalidades Suportadas

Antes de entrar no jogo o programa permite ao utilizador definir o tamanho do ecrã entre três possibilidades: 800x500, 1024x640 e *Full Screen*, bem como as teclas que levam ao movimento da cobra e pausa. Também é possível consultar uma lista com os dez jogadores que obtiveram maior pontuação ao jogar.

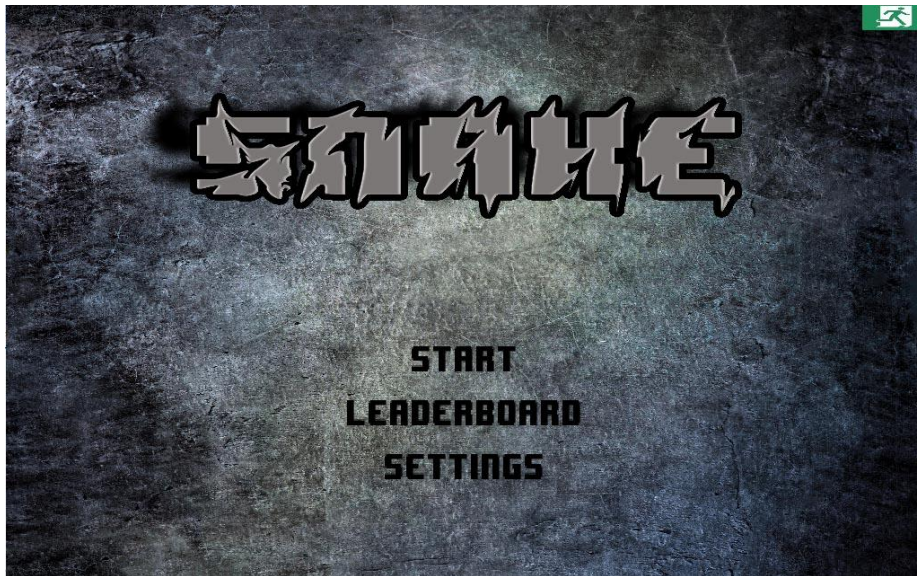
Ao entrar no jogo, o utilizador escolhe o tamanho do tabuleiro, a velocidade inicial da cobra e se quer jogar com ou sem paredes. Em vez de criar um novo jogo também é possível carregar um que tenha sido previamente gravado.

No decorrer do jogo há 5 diferentes itens para a cobra comer: comida normal (representada por uma caveira, que faz o tamanho da cobra aumentar), um raio verde (faz a velocidade da cobra aumentar 1.5x), um raio vermelho (faz a velocidade da cobra diminuir 1.5x), uma tesoura (reduz o tamanho da cobra para metade) e uma bomba (mata a cobra e termina o jogo).

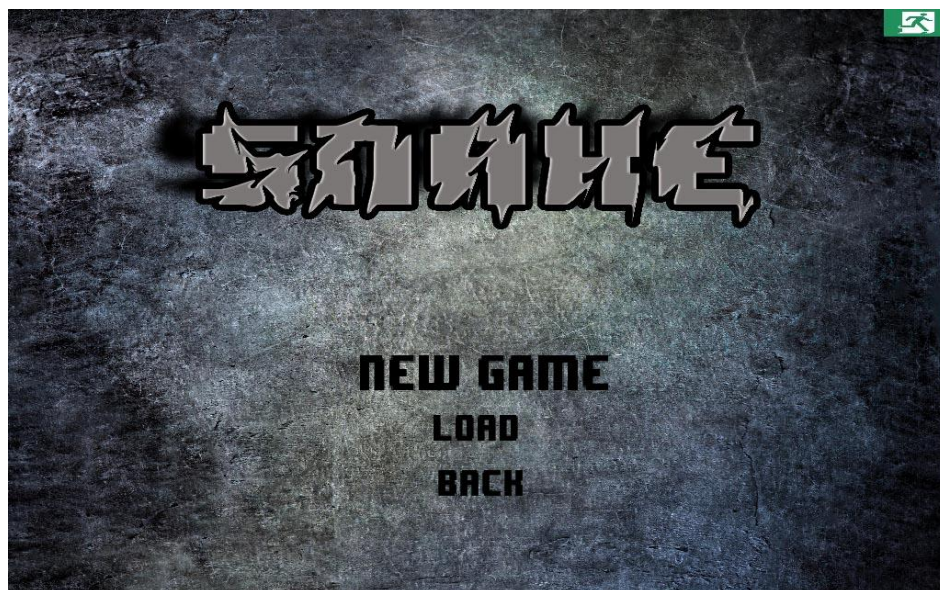
### 2.2. Arranque do Programa

Antes de iniciar o jogo o utilizador deve clicar em *Settings*, para escolher um tamanho de ecrã e as teclas a usar no jogo. Quando clicar em *Ok* volta ao Menu Inicial. Para arrancar o jogo clica-se em *Play* e depois *New Game* para iniciar um novo jogo ou *Load* para carregar um jogo gravado. Ao fazer *New Game* é apresentada uma nova janela onde o utilizador escolhe a dimensão do tabuleiro, a velocidade inicial da cobra e o modo de jogo (com ou sem paredes).

### 2.3. Modo de utilização



Menu Inicial: permite ao utilizador aceder ao menu onde o utilizador pode começar um novo jogo ou carregar um antigo, assim como à leaderboard e às definições do jogo.

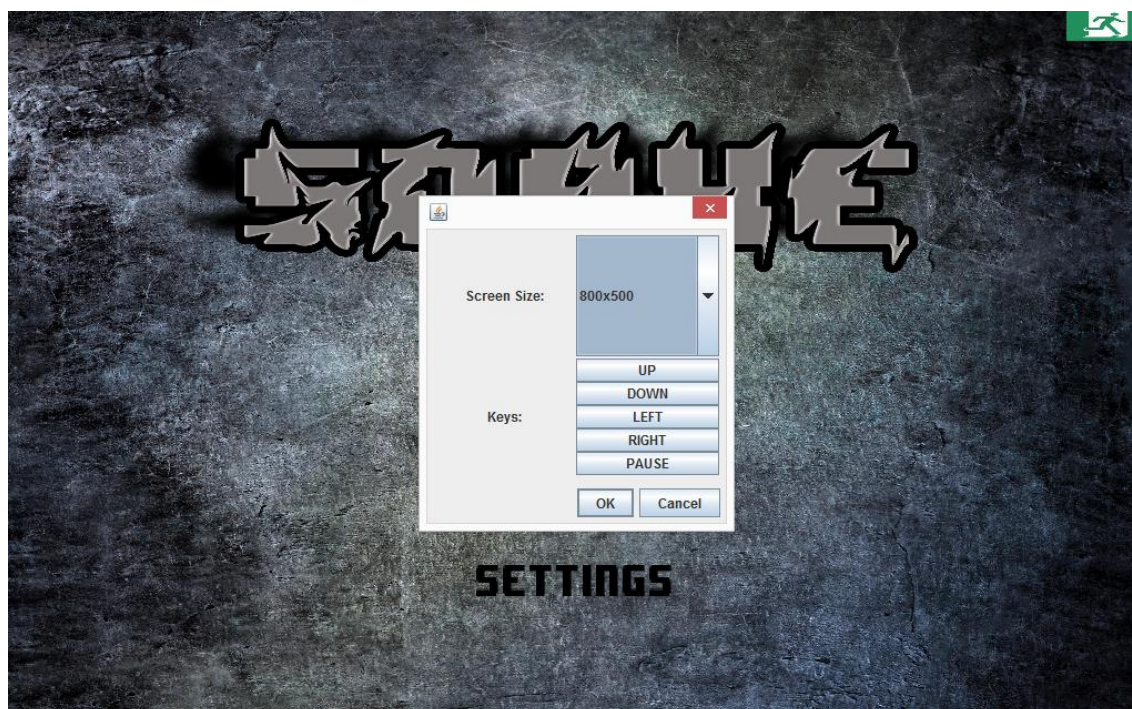


Neste menu o utilizador escolhe se pretende iniciar um novo jogo, carregar um jogo antigo ou voltar ao menu inicial.



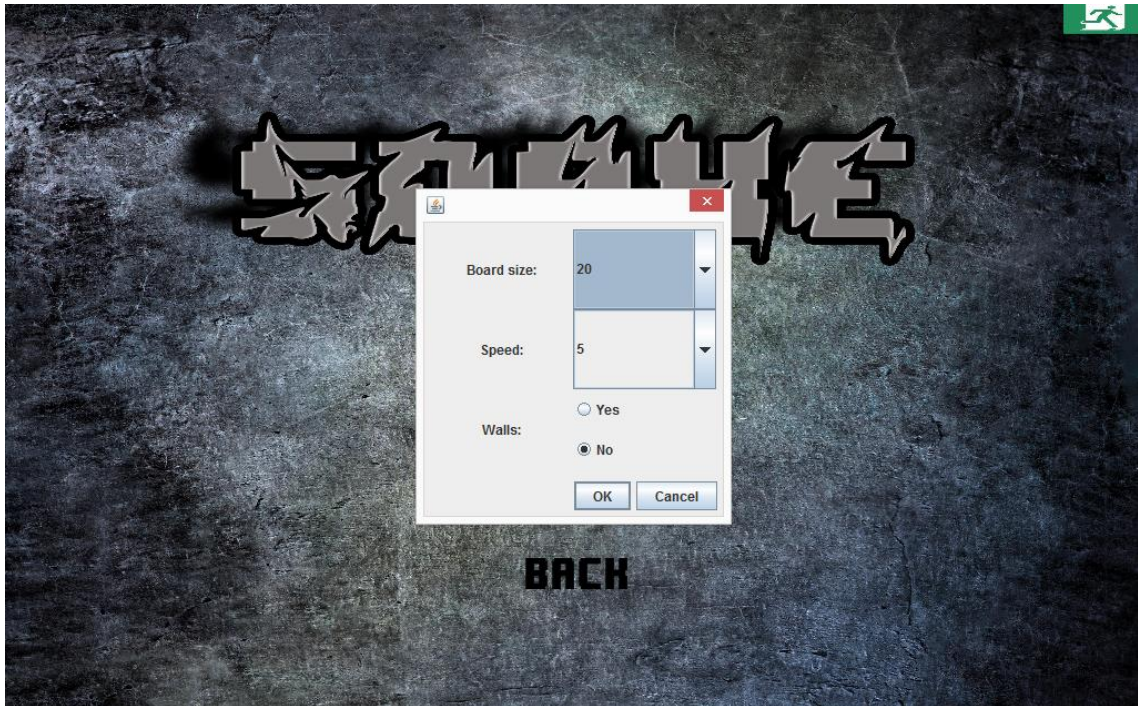


Leaderboard: Os dez jogadores que obtiveram maior pontuação.

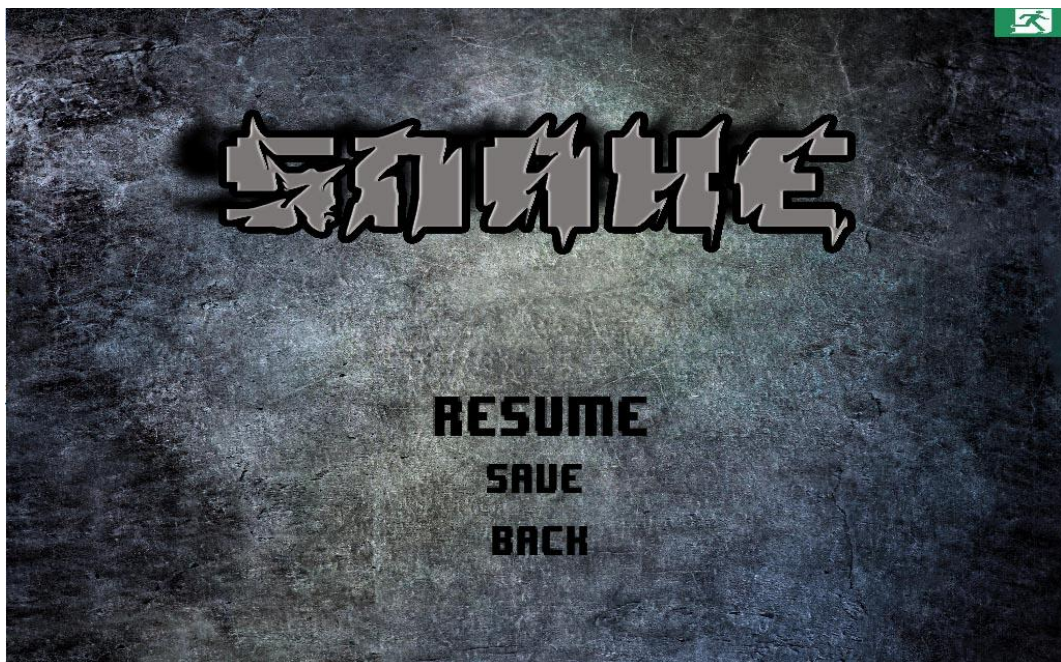


Definições do jogo: Permite ao utilizador definir o tamanho da janela e redefinir as teclas do jogo.





Novo jogo: Aparece quando se clica em *New Game*. Permite ao utilizador escolher o tamanho do tabuleiro, a velocidade inicial da cobra e se o tabuleiro tem, ou não, paredes.



Menu Pausa: Permite ao utilizador voltar ao jogo, guardar o jogo que está a decorrer e voltar ao Menu Inicial.

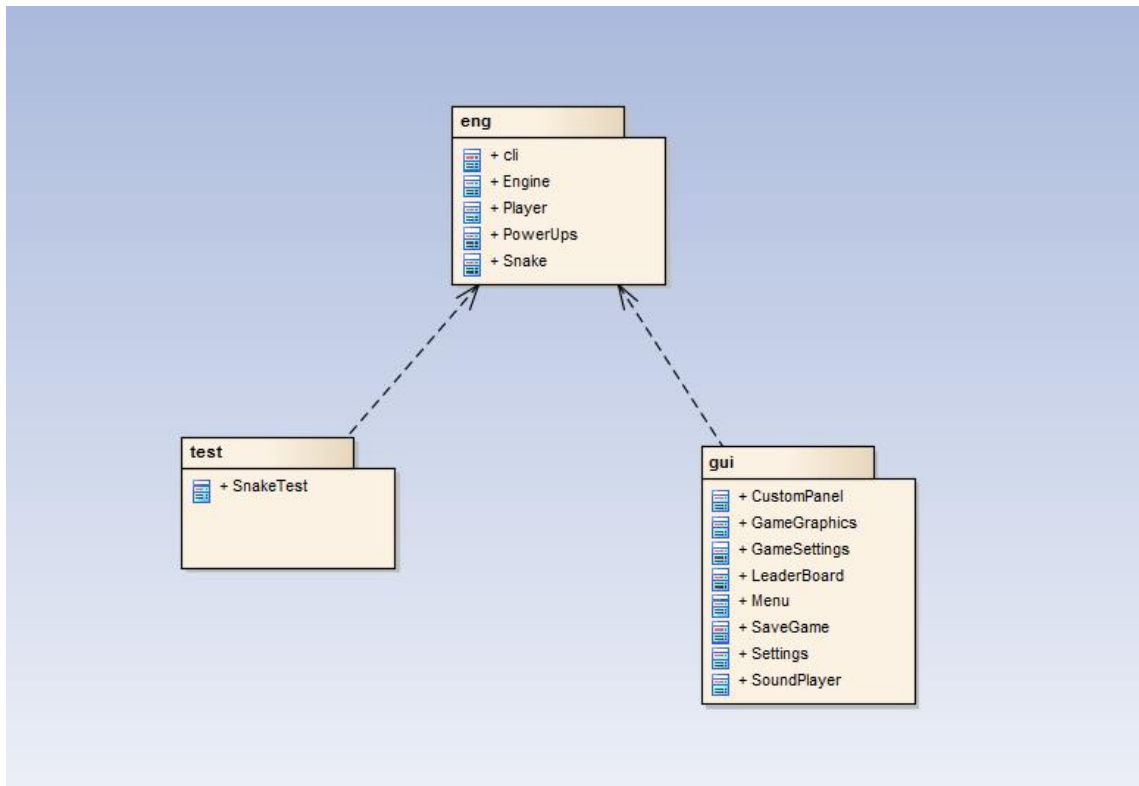


Game Over: o utilizador escreve o nome que vai aparecer na *leaderboard* associado à pontuação efetuada no jogo que terminou.



### 3. Conceção e Implementação

#### 3.1. Estrutura da Packages

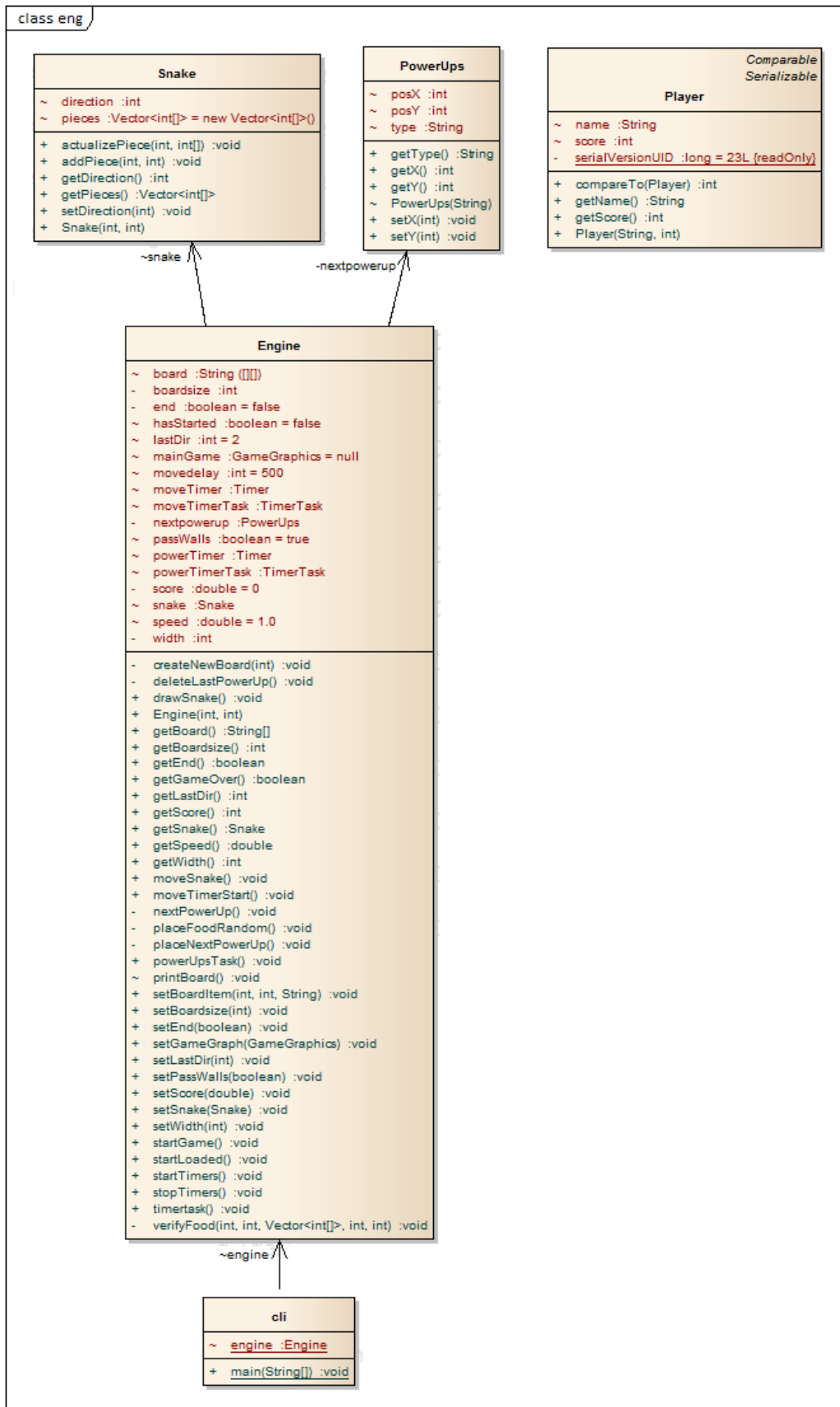


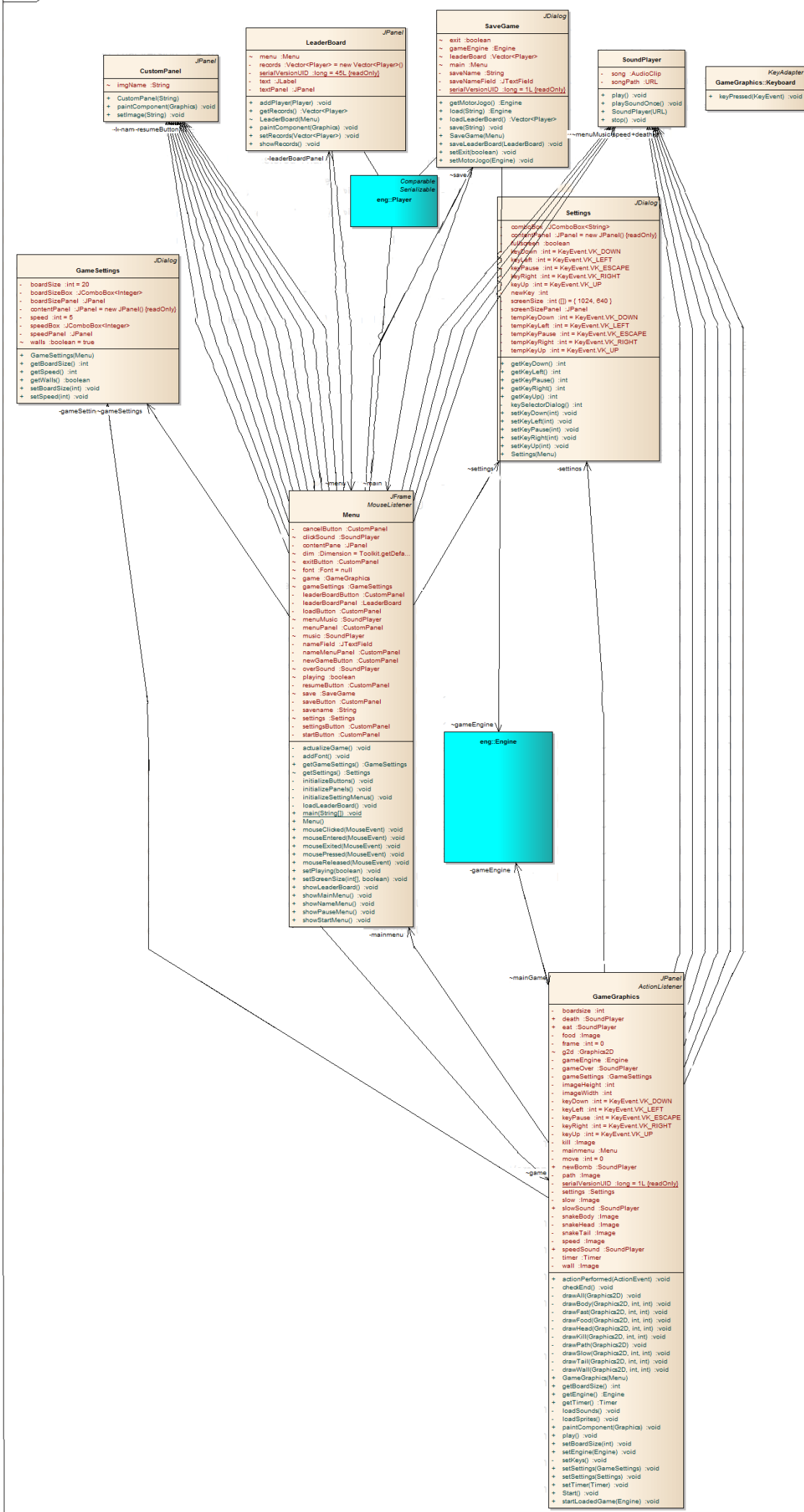
Eng – Trata de todos os aspetos relacionados com a mecânica do jogo.

Test – Contém o ficheiro de testes.

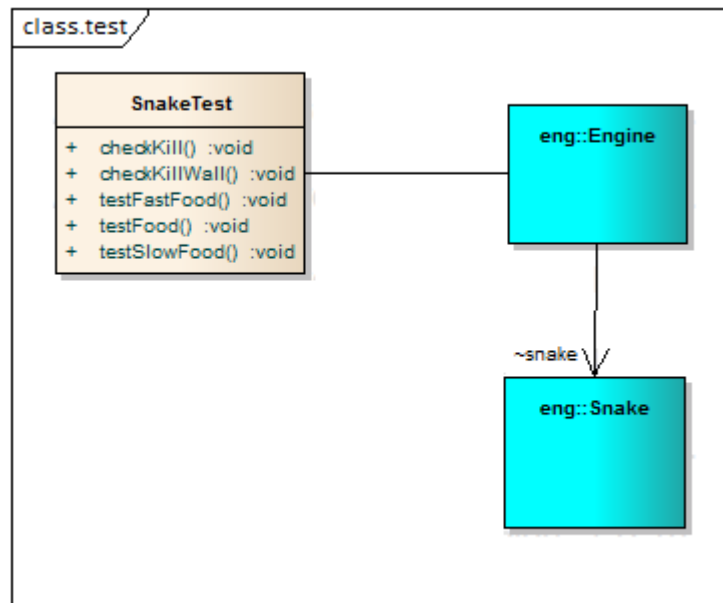
Gui – Trata dos aspetos relacionados com a interface gráfica.

## 3.2. Estrutura de Classes









Nas classes do package “eng”, a classe *Engine* é responsável pelo movimento da cobra no tabuleiro, bem como por fazer as verificações de fim de jogo, gestão de *powerups* e de comida normal. A classe *Snake* contém a direção para onde a cobra está a ir e o vetor que contém as suas peças. Na classe *Powerups* o parâmetro “type” é o tipo de *powerup* em questão (acelerar, desacelerar, matar). A classe *Player* tem como objetos os jogadores que vão ser inseridos na *leaderboard*.

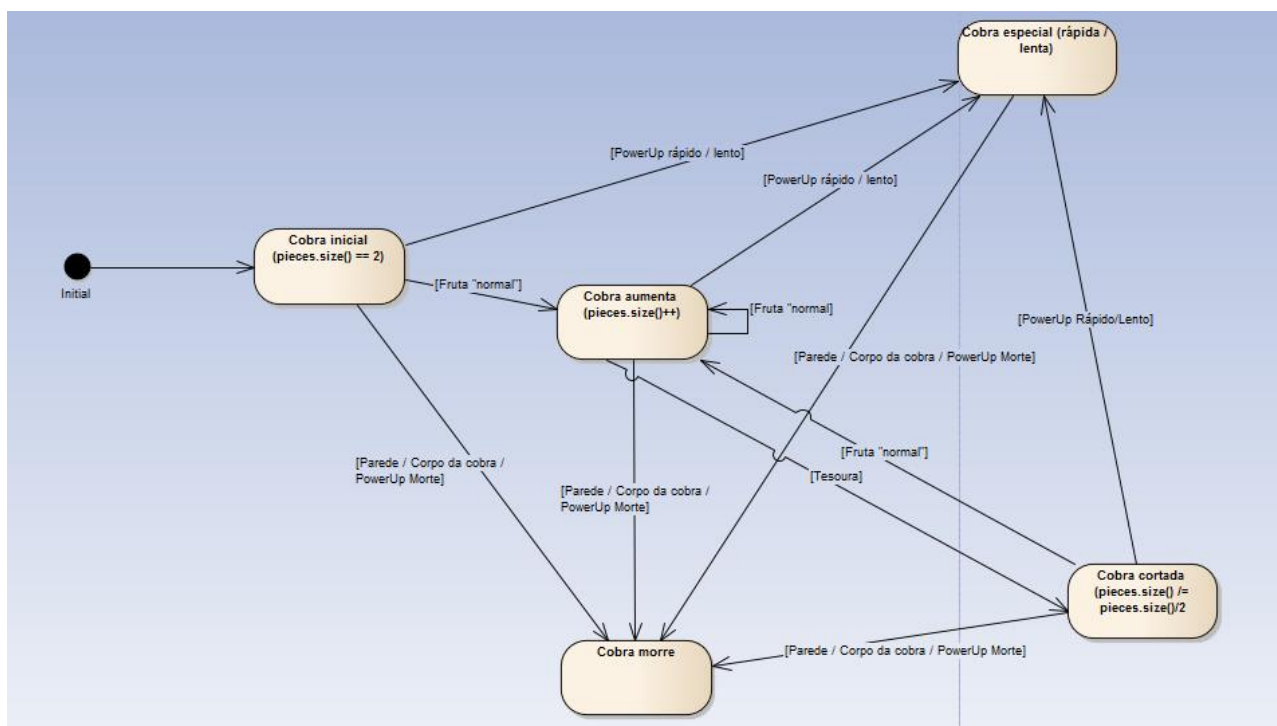
Nas classes do package “gui” a classe *Menu* é responsável pela criação de todos os painéis dos vários menus presentes no jogo, bem como os vários painéis que são utilizados como botões. A classe *Settings* é responsável por mostrar um menu que permite ao utilizador mudar a resolução do ecrã e redefinir as teclas utilizadas no jogo. A classe *Game Settings* é responsável por mostrar um menu que permite ao utilizador escolher o tamanho do tabuleiro de jogo, a velocidade da cobra e se existem paredes ou não. A classe *GameGraphics* é responsável por desenhar o tabuleiro de jogo e todos os seus componentes e receber os *inputs* do utilizador e passa-los ao *engine*. A classe *SaveGame* é responsável por todos os *saves* e *loads* que são feitos pelo jogo, tanto da sua *leaderboard* como dos jogos gravados.

### 3.3. Padrões de Desenho

No nosso jogo usamos o padrão de desenho *Model-View-Controller*. Como *Model* temos o tabuleiro, a cobra que nele se move e os diferentes tipos de comida para a cobra. A cobra é manipulada por um *Controller*, que é ativado pelo utilizador ao premir as teclas que fazem a combra movimentar-se no tabuleiro. Como *View*, temos o output do programa, que usa o conteúdo da matriz tabuleiro para, consoante as posições dos diferentes objetos, mostrar ao utilizador o estado do jogo.

### 3.4. Mecanismos Importantes

A cobra, elemento principal do jogo, tem o seguinte mecanismo:



## 4. Conclusão

Com o desenvolvimento deste trabalho foi-nos possível aprofundar os nossos conhecimentos de Programação Orientada por Objetos e do quão prático isso é.

Conseguimos cumprir com todos os objetivos a que nos propusemos no início e ainda acrescentámos novas funcionalidades no programa que achámos ter interesse incluir e que nos foram surgindo ao longo do desenvolvimento.

Realizámos ambas quantidades de trabalho aproximadamente iguais, tanto a nível pessoal como *online*.



## Referências

- <http://stackoverflow.com/> (várias páginas – consultado pela última vez a 7 de Junho de 2014).
- Slides da Unidade Curricular