

# ISEL

*Instituto Superior Engenharia de Lisboa*

## 3<sup>a</sup> Serie

Engenheiro: Carlos Guedes

Engenharia Informática e de Computadores

Semestre de Inverno 2011/2012

João Silvestre - 32766

12/02/2012



## Índice

---

Introdução .....	4
Estrutura do Código .....	5
As Classes .....	5
Board .....	5
Entity .....	5
GameObject .....	6
Model .....	6
ModelProvider .....	7
Effects .....	7
Modelo de Classes .....	<b>Erro! Marcador não definido.</b>
Aspecto do Jogo .....	9
Conclusão .....	10

## Introdução

---

Este trabalho consiste na criação de um jogo usando a tecnologia OpenGL para o desenho do ambiente 3D e da linguagem C++ para o desenvolvimento do mesmo. O jogo escolhido para ser implementado é o Pacman, este consiste numa personagem que tem que comer todas as pastilhas no campo enquanto foge de vários fantasmas que têm como objectivo apanhá-lo.

Junto com o projecto encontra-se documentação, gerada automaticamente por uma ferramenta, que poderá servir de complemento a este relatório.

## Estrutura do Código

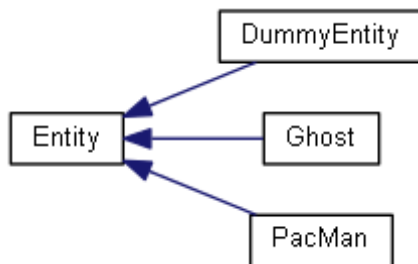
Nesta secção irá ser explicado o objectivo de cada “conjunto” de classes presentes no projecto e um diagrama, onde se justificar, indicando quais as classes que constituem o grupo.

### As Classes

#### Board

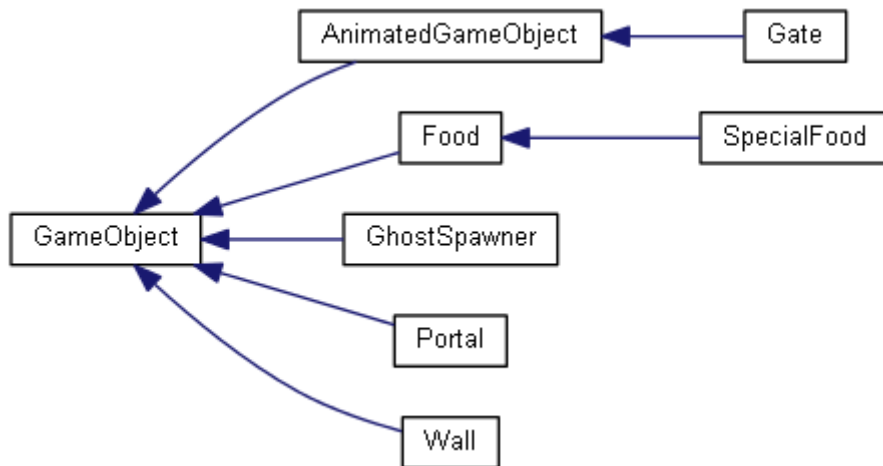
A classe board descreve o jogo a acontecer no momento, contendo o mapa a ser mostrado ao jogador assim como todas as entidades, pacman e fantasmas. Esta oferece métodos para se saber quais as entidades presentes no jogo e o objecto que se encontra em determinada posição. É também esta responsável por processar a lógica de colisão entre entidades, mandar actualizar e desenhar todos os objectos e entidades, e também é esta que controla a abertura e fecho dos portões.

#### Entity



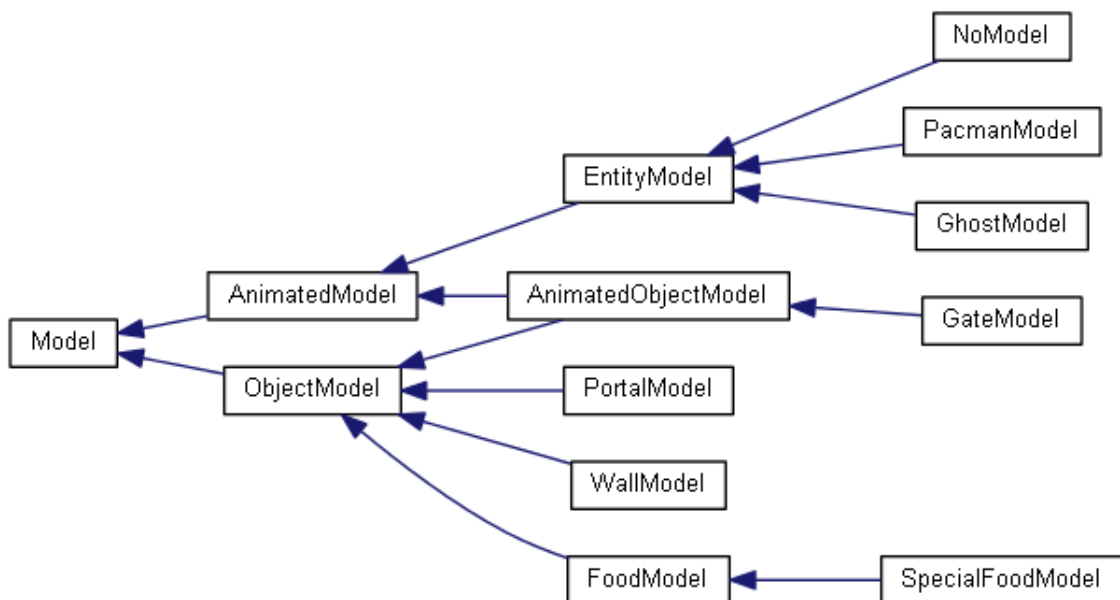
As entidades descrevem componentes móveis do jogo, ou seja, que não têm uma posição física no *board*. Estas componentes descrevem as “personagens” do jogo, os fantasmas e o pacman. A classe base das entidades oferece métodos para facilitar a movimentação em cada uma das 4 direcções, sendo que sempre que acontece um movimento esta irá notificar o objecto, que se encontra na sua nova posição, que está em colisão com a mesma. A entidade PacMan é controlada pelo utilizador usando as teclas direccionais do teclado, sendo que a entidade Ghost é controlada por uma de duas AI, dependendo da situação actual do jogo. Se os portões se encontrarem abertos a AI activa será uma de fuga da “jaula”, assim que os portões fecham entra em acção a AI que irá por os fantasmas a perseguir o Pacman.

## GameObject



Os objectos descrevem os componentes imóveis do jogo, este são descritos pela posição em que se encontram no *board* para que possam ser desenhados no sítio certo. Estes também implementam um método sobre a acção a tomar no caso de ocorrer uma colisão, por exemplo, no caso da comida é actualizado o estado da entidade Pacman em como comeu mais uma pastilha.

## Model



Tanto as *entity* como os *gameobject* têm que ser desenhados no ecrã, este aspecto foi feito os possíveis para separar ao máximo da lógica do jogo. Permitindo uma rápida troca do aspecto de todo o jogo sem que isto vá ter influências no seu comportamento. Os modelos encontram-se ainda divididos em 3 diferentes tipos:

- Os modelos de entidades, que são sempre animados sendo esta animação executado quando ocorre um movimento.
- O modelo de objectos, estes não são animados e apenas se sabem desenhar de forma estática.
- O modelo de objectos animados, que permitem que o objecto tenha algum tipo de animação, quando esta animação ocorre não se encontra especificado. Neste momento existe apenas um caso, o portão, que tem a animação quando se abre e fecha.

### ModelProvider

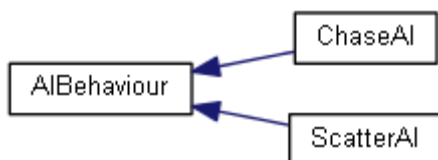
Esta classe é responsável por devolver o modelo certo para determinado tipo de objecto ou entidade, embora de momento esta decisão seja feita estaticamente, esta classe abre portas a carregar dinamicamente de bibliotecas os modelos, entre outras coisas, dando uma maior liberdade na escolha do aspecto do jogo.

### Effects



Os efeitos descrevem o que acontece quando o pacman come determinada comida, embora de momento apenas se encontre desenvolvido um tipo de comportamento seria aqui que se deveria criar o comportamento da comida “especial”, alterando a AI dos fantasmas para uma de fuga e também a sua cor, se pretendermos replicar por completo o jogo original.

### AIBehaviour



Esta classe serve para descrever o comportamento de uma entidade tendo em conta o *board* e um objectivo, objectivo este que se encontra definido como uma entidade. No caso

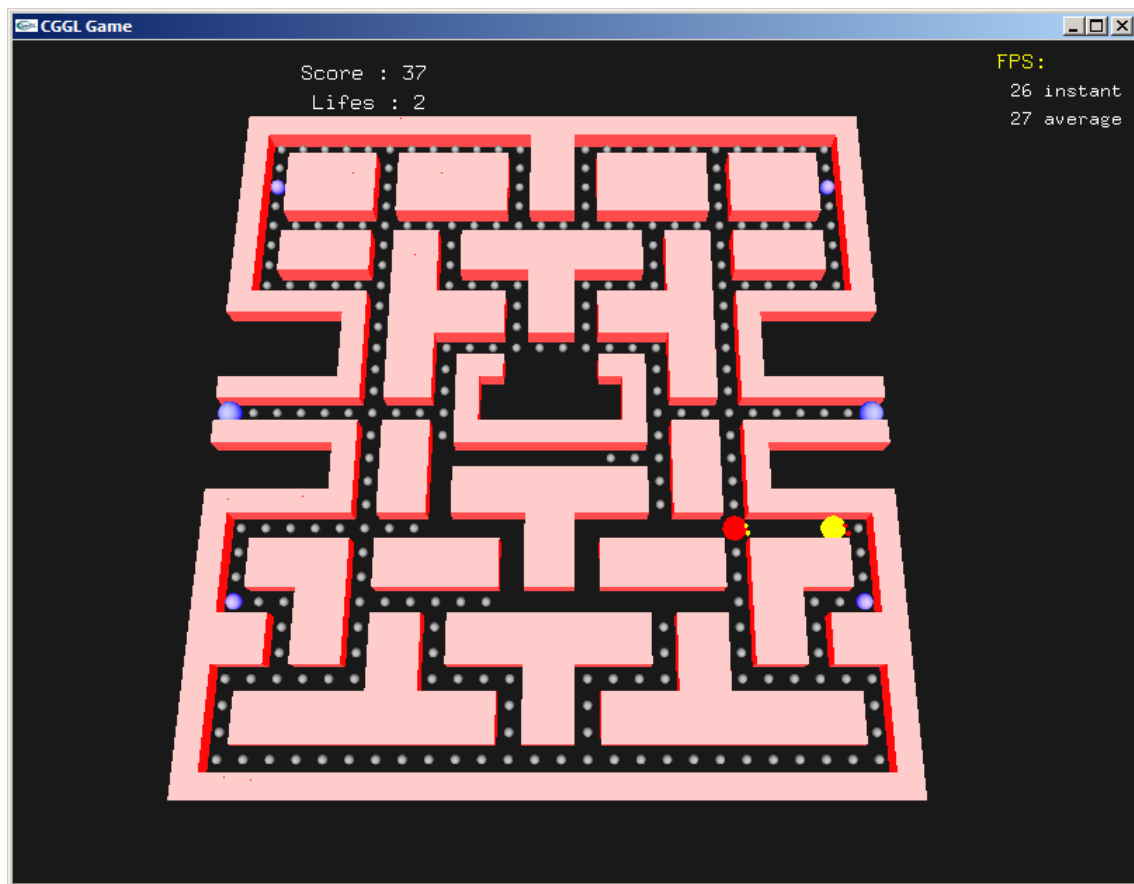
particular da *ScatterAI*, que tem como objectivo tirar o fantasma da “jaula” é usado uma entidade *dummy* que indica onde se encontra o *gate*.

### GameStaticSettings

Não é uma classe, é sim um *header file* com algumas definições de dados que embora estáticamente, em momento de compilação, permitem configurar alguns parâmetros do jogo, como o número de fantasmas em jogo e o detalhe de alguns dos modelos.



## Aspecto do Jogo



## Conclusão

---

Embora tenha sido implementada uma versão funcional do jogo é claro que muito mais poderia ter sido implementado/melhorado para uma experiência mais agradável na utilização do mesmo. Embora acredita-se ter-se feito um bom trabalho na implementação do jogo em si, embora existam ainda alguns bugs que até à data de entrega não foi possível corrigir, acredita-se que o aspecto visual acabou por ser um posto de parte. Quer-se salientar que isto se deve em parte à tentativa falhada de implementação de um outro jogo para ser usado como entrega neste trabalho, tendo este consumido boa parte do tempo disponível e não ter sido possível continuar com o mesmo.