Universidade do Porto

Faculdade de Engenharia

FEUP

Bomberman

Relatório Final

Laboratório de Programação Orientada por Objetos

2º ano do Mestrado Integrado em Engenharia Informática e Computação

Elementos do Grupo:

Rodolfo Rodrigues – 200200428– ei12151@fe.up.pt

Vítor Mineiro – 200105060 – ei12153@fe.up.pt

8 de Junho de 2014

1. Introdução

Objetivo do relatório

O presente trabalho, desenvolvido no âmbito da unidade curricular Laboratório de Programação Orientada a Objetos, do 2º ano do curso Mestrado Integrado em Engenharia Informática, da Faculdade de Engenharia da Universidade do Porto, propõe conceber um jogo, em linguagem Java, baseado no jogo **Bomberman** que inicialmente desenvolvido pela *Hudson Soft*.

A elaboração deste trabalho tem como principal desenvolver uma aplicação em Java e aplicando os conhecimento e técnicas abordadas ao longo do presente semestre, com especial foco nas técnicas de desenho e linguagem orientada por objetos.

Objetivo do jogo

O objetivo principal do jogo consiste destruir todos os inimigos, de forma a ativar a saída. Para destruir os inimigos possui bombas para depositar e com a explosão da bomba atingir os inimigos. No entanto, caso o jogador seja atingido pela explosão ou choque com os inimigos perde.

Ao longo do jogo o jogador poderá apanhar vários itens que lhe permitirão aumentar o número de bombas máximo que pode depositar em simultâneo, aumentar a potencia da bomba entre outras. Para apanhar os itens o jogador terá de destruir blocos de parede de tijolo.

Estrutura do relatório

No ponto 2 deste relatório temos um um pequeno manual da aplicação onde podemos ver um resumo das funcionalidades do programa, modos de funcionamento e interligação entre os diversos menus.

O ponto 3 engloba e explica a parte mais técnica da aplicação, como estrutura de packages e estrutura de classes utilizada na concepção do aplicação e que é possível visualizar no diagrama UML que se encontra no ficheiro T5_ei12151_ei2153_Bomberman_FINAL.eap que segue em anexo junto com o relatório. No ponto 3 são indicados os padrões de desenho utilizados e é feita uma breve explicação sobre os motivos pelo os quais foram utilizados.

2. Manual de utilização

Resumo das funcionalidade suportadas

Relativamente às funcionalidades suportadas da aplicação procuramos cobrir o maior número de funcionalidades propostas no enunciado do trabalho, no entanto, não nos foi possível cobrir a funcionalidade de funcionamento em rede, pelo que a aplicação implementa as seguintes funcionalidades:

1. Interface gráfica para o utilizador com Swing
2. Manipulação de ficheiros
3. Animação de objecto gráficos

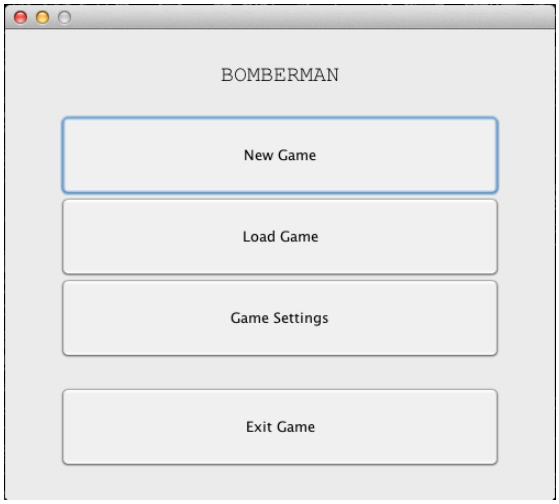
Arranque do programa

Junto com o relatório segue o ficheiro executável bomberman.jar onde se poderá testar o funcionamento do programa sendo apenas necessário uma máquina virtual de Java para correr o programa.

No entanto, para correr a aplicação através dos ficheiros de código fonte basta correr a aplicação, no IDE, através da classe MainWindow que se encontra dentro do package gui.

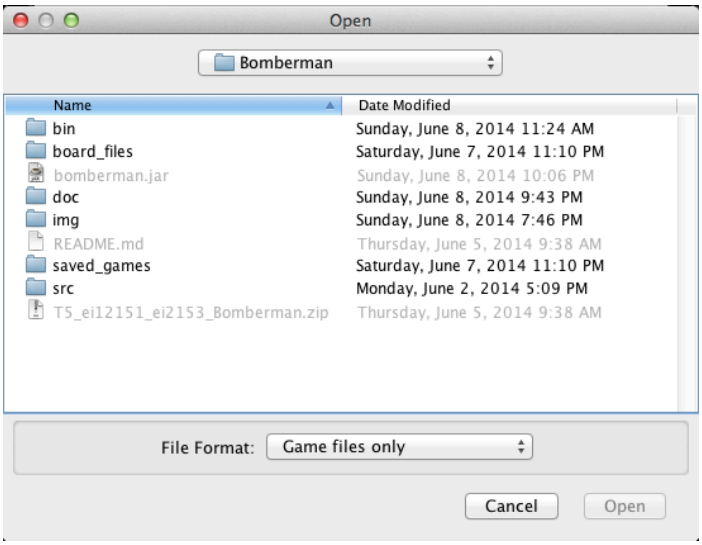
Modo de utilização (imagens dos ecrãs e explicação da informação e ações disponíveis em cada ecrã)

Nesta trabalho procurou-se chegar a uma aplicação com um interface minimalista de forma a obtermos uma interface intuitiva e de fácil utilização para o utilizador. Desta forma o programa inicia com a seguinte janela.



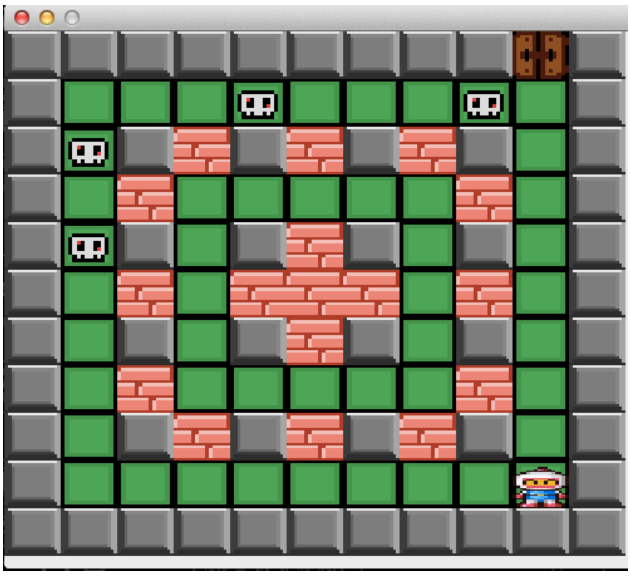
Como podemos ver, temos 4 botões, uma para iniciar um novo jogo (New Game), outro para carregar um jogo previamente gravado, outro para configurações de jogo e o ultimo para sair da aplicação.

Ao premir no botão para carregar o jogo abrirá uma janela de diálogo onde poderá navegar por um explorador de ficheiros para aceder à pasta onde se encontra o ficheiro guardado.



Como se pode ver na imagem acima, a aplicação aplica um filtro aos vários tipos de ficheiro permitido apenas selecionar ficheiros não suportados pela aplicação.

Após efectuado o carregamento do jogo, a janela principal do jogo passa a apresenta o labirinto do nível onde se encontra o jogador.



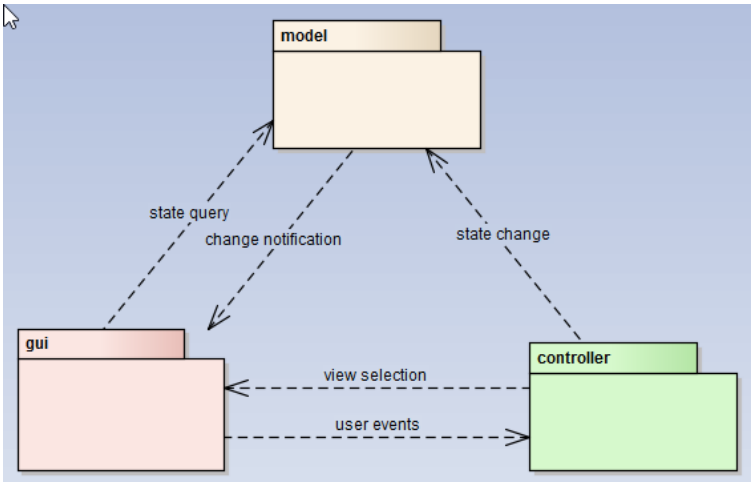
Durante o jogo o jogador poderá gravar, pausar o jogo ou mesmo carregar um jogo previamente gravado.. Para isso terá de pressionar na tecla Escape e ser-lhe-á apresentado um novo menu com as diversas opções.

Quando o jogo termina é apresentada uma nova janela com o resultado do jogo.

3. Concepção e implementação

Estrutura de packages

Dado que a aplicação a desenvolver é um jogo de computador identificamos o padrão de desenho Model-View-Controller. Nesse sentido procuramos dividir as diversas classes por packages de forma a agrupar as classes com o mesmo contexto.



No entanto ao passarmos o padrão de desenho de alto nível para a separação em concreto das classes, encontramos a primeira dificuldade. Esta dificuldade consistia na separação da lógica de jogo do modelo de jogo. Contudo chegamos à seguinte tabela de packages:

Package	**Classes**	**Responsabilidade**
model	GameModel Position	Lógica Geral de Jogo Análise do estado do jogo
model.monster	Monster MonsterState MonsterAlive MonsterDead	Lógica de jogo: - movimentação dos monstros - controlo dos estados dos montros
model.board	Board BoardFactory Item BoardExit UndestructibleWall Path ItemState ItemHidden ItemDetonating ItemActive ItemExploding ItemInactive	Construção do tabuleiro Lógica de jogo: - movimentos possíveis - lógica dos itens - transições de estados dos itens

Package	Classes	Responsabilidade
model.player	Player PlayerState PlayerAlive PlayerDead Bomb	Lógica de jogo: - movimentação do jogador - controlo dos estados do jogador
gui	MainWindow MainMenuPanel GameType GamePanel PausePainelDialog SettingsDialog	Visualização do estado jogo Interação com o utilizador
tests	BoardTests MonstersTests PlayerTests	Testes da aplicação - testes unitários aos vários métodos e lógicas de jogo

Para uma análise mais detalhada é possível consultar o ficheiro em T5_ei12151_ei2153_Bomberman_FINAL.eap que se encontra na pasta /doc/uml dentro do ficheiro zip enviado juntamente com este relatório.

Padrões de desenho utilizados

No desenvolvimento da aplicação foram utilizados diversos padrões de desenho que nos permitiram melhorar a interligação entre os diversos objetos do jogo, melhorar o desempenho da aplicação, ultrapassar algumas dificuldades, mas acima de tudo permitiu-nos obter uma aplicação modelar e com bastante potencial evolutivo sem que seja necessário alterar muito o código.

De uma forma resumida os padrões de desenho ultilizados foram:

1. Singleton
    - O padrão de desenho foi implementado na classe GameModel, porque na aplicação só é possível jogar um jogo de cada vez,

embora um jogo possa ter vários labirintos. Sendo assim estamos a evitar criar objeto desnecessário, poupando recursos ao computador.

2. Strategy
    - Este padrão foi implementado em praticamente todos os objetos do jogo. O uso deste padrão permitiu-nos manusear as alterações de estado dos objetos e a manusear os seus sprites.

3. Visitor
    - Padrão utlizado para validar movimentos possíveis dos monstros e jogadores, tratamento de colisões entre o jogador e os monstros. Nesse sentido, os jogador, os monstros e a bomba foram implementados como visitante e os itens como visitados.

4. Factory
    - Este padrão foi implementado na classe boardFactory. O objectivo principal da sua implementação foi obtermos a criação de novos tabuleiros de forma desligadas da lógica do jogo.

Mecanismos importantes

No que diz respeito a mecanismos importante da aplicação, comportamento do programa, relativamente ao relatório intermédio não houve alterações significativas pelo que junto do relatório segue também o ficheiro relativo ao relatório intermédio onde é possível analisar o diagrama de sequências dos objetos tipo do jogo e em cada uma das classes é possível visualizar o respectivo diagrama de estados.

Dificuldades encontradas e sua resolução (se aplicável)

De uma forma geral o grupo sentido algumas dificuldades na fase inicial do desenvolvimente da aplicação mais especificamente em selecionar quais os melhores padrões de desenho que se aplicavam à aplicação e na sua transposição para código. Uma vez ultrapassadas essas dificuldades o processo de desenvolvimento da aplicação decorrer de forma normal, com necessidade de efectuas algumas pesquisas e consultas de documentação.

4. Conclusões

Grau de cumprimento dos objetivos

De uma forma geral o grupo considera que os objetivos foram alcançados com especial destaque no nível de aplicações de padrões de desenho e modelação da aplicação.

Melhorias possíveis

Embora consideremos que os objectivos tenham sido alcançados, existe ainda algumas melhorias que podem ser efectuadas, nomeadamente na interface gráfica e eventualmente adicionar alguns elementos de jogo que foram removidos aquando a sua concepção.

Nível de contribuição dos elementos do grupo

De uma forma geral todos os elemento do grupos contribuíram para a realização deste trabalho. Contudo procurou-se distribuir de forma uniforme o volume de trabalho pelos os elementos do grupo.

5. Referências

Durante o desenvolvimento da aplicação recorremos às seguinte bibliografia:

Bruce Eckel; Thinking in Java. ISBN: 0-13-027363-5

Russ Miles and Kim Hamilton; Learning UML 2.0. ISBN: 978-0-596-00982-3

Para além da bibliografia supracitada foram efectuadas diversas pesquisas na internet. Dos sítios consultados destacamos os seguintes:

stackoverflow.com

http://docs.oracle.com/

Nota: A maior parte das pesquisas na internet efectuadas foi utilizado o motor de busca da Google.

www.google.com