

Mestrado Integrado em Engenharia Informática e Computação  
2ºAno 2ºSemestre



*Monopoly*



Mário André Macedo Ferreira - [ei12105@fe.up.pt](mailto:ei12105@fe.up.pt)  
201208066

Professor Dr. Ademar Aguiar  
Professor Dr. Jorge Barbosa

Rua Dr. Roberto Frias, s/n, 4200-465 Porto, Portugal  
09 de Junho de 2014

# Índice

Índice .....	ii
1. Introdução .....	iii
2. Manual de Utilização.....	iv
2.1. Funcionalidades Suportadas.....	iv
2.2. Modo de Utilização .....	iv
3. Concepção e Implementação.....	vii
3.1. Estrutura de <i>Packages</i> .....	vii
3.1. Estrutura de Classes.....	vii
4. Conclusão .....	viii
5. Webgrafia .....	ix

# 1. Introdução

Este trabalho foi realizado no contexto da unidade curricular Laboratório de Programação Orientada a Objectos, LPOO, do 2º semestre do 2º ano do ano lectivo 2013/2014 do curso de Mestrado Integrado em Engenharia Informática e Computação (MIEIC).

O trabalho surge na continuação da matéria leccionada ao longo do semestre sobre programação orientada a objectos, animações, desenvolvimento de interface gráfica. O início do projecto foi marcado com um desenho inicial do diagrama de classes, utilizando a ferramenta *Enterprise Architect*, que foi sendo actualizado e alterado conforme as necessidades no decorrer da codificação. Para além disto foi ainda gerado documentação *Javadoc* de quase todo código, focando as partes mais importantes do mesmo.

No sentido de dar a entender a forma como o projecto foi desenhado, implementado, funcionalidades que possui, este relatório apresenta uma estrutura simples e que passa inicialmente por um pequeno manual de utilização da aplicação onde será realçada a interface com o utilizador com a ajuda de alguns *mockups*, sendo que parte deles são provenientes da primeira entrega deste mesmo projecto e outros sofreram pequenas alterações. Ainda no manual de utilização será feita uma descrição das funcionalidades do jogo que foram implementadas. Numa fase seguinte será abordada a estrutura de implementação do código onde se dará uma breve explicação sobre o papel das diversas classes e a ligação entre os *packages* envolvidos.

De um modo geral, penso ter implementado quase todas as funcionalidades pedidas, apesar de ter consciência que o projecto não está 100% funcional.

## 2. Manual de Utilização

### 2.1. Funcionalidades Suportadas

No que toca aos modo de jogo disponível, ao contrário do que foi pedido, “implementação do jogo em rede”, a aplicação apenas suporta o modo multiplayer de 2 até 4 jogadores num modo local, isto é, no mesmo computador. É de salientar que foi realizada a tentativa de realização do jogo em modo de rede, contudo sem sucesso.

Para além disto o jogo apresenta as funcionalidades de guardar o estado de jogo actual e de abrir um jogo previamente guardado ao invés de criar um novo jogo.

O jogo apresenta ainda as funcionalidades características de um jogo tradicional de *Monopoly*, nomeadamente, comprar propriedades, hipotecar e anular a hipoteca de uma propriedade, adicionar e remover apartamentos e hotéis.

### 2.2. Modo de Utilização

Inicialmente quando o programa inicializa, o jogador é confrontado com a hipótese de iniciar um novo jogo ou carregar um jogo previamente guardado como se vê na imagem seguinte:

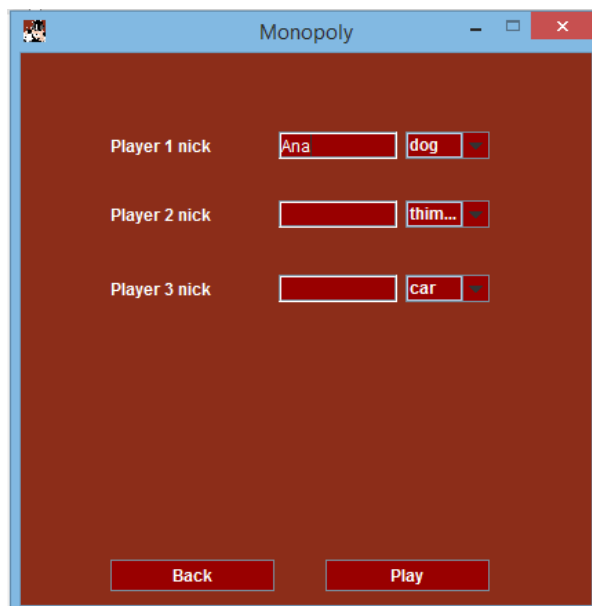


De seguida se o utilizador pressionar “*Load Game*” é-lhe apresentado uma janela para escolher um ficheiro “.dat” e caso este seja válido é apresentado a janela de jogo com os dados previamente guardados. No caso de o utilizador pressionar “*Exit*”

a aplicação é terminada. Por ultimo, se for pressionado abre a segunda janela de seleção de número de jogadores:



Nesta fase se for premido o botão “Back” regressa-se para a janela anterior e se for premido o botão “Play” é aberto uma janela onde os jogadores colocam os seus nomes e escolhem as peças que os representam:



Uma vez mais se for premido o botão “Back” regressa-se para a janela anterior e se for pressionado o botão “Play” inicia-se o jogo, abrindo uma janela com o tabuleiro e todos os controlos para o desenrolar do jogo:



Neste momento os jogadores já se encontram em jogo e os botões “Roll Dice”, “Buy”, “Mortgage” e “Buildings” apresentam os resultados das operações no painel de ação (zona entre a informação dos jogadores e os botões), como se vê no seguinte exemplo:



## 3. Conceção e Implementação

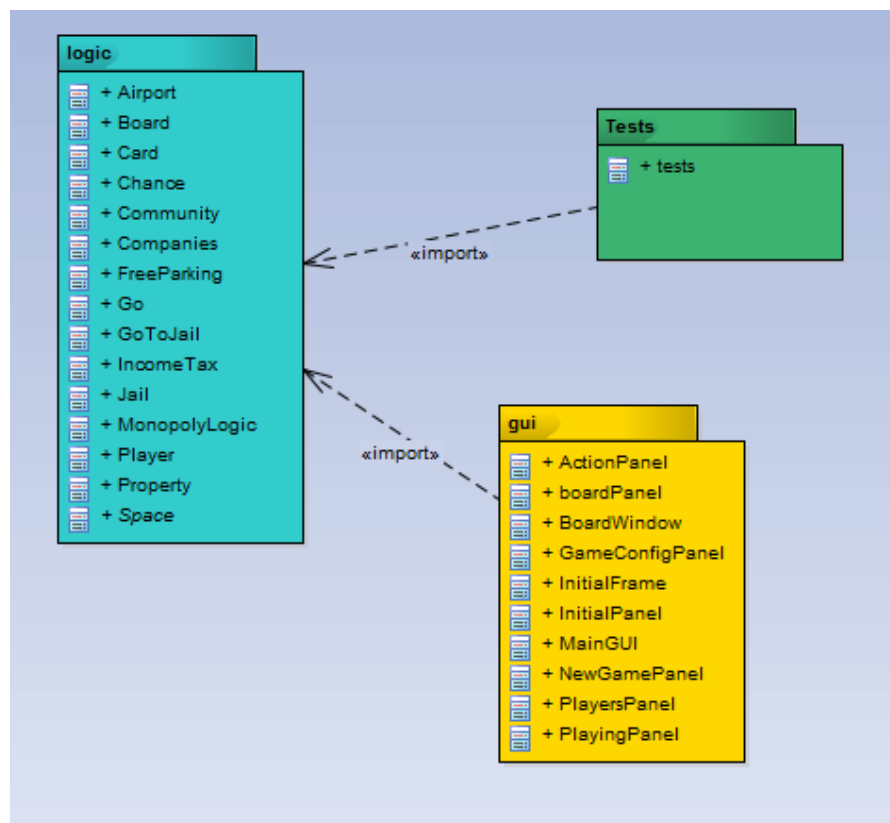
### 3.1. Estrutura de *Packages*

O projecto encontra-se dividido em 4 *packages*, nomeadamente, *logic*, *Tests* e *gui*, sendo que no *package logic* encontra-se desenvolvida toda a implementação da lógica do jogo e é o local onde são armazenados todos os conteúdos e onde se estabelecem todas as relações conceptuais do jogo.

No *package Tests* encontram-se alguns testes desenvolvidos em *JUnit* que ajudaram a testar o correcto funcionamento da lógica do jogo.

No *package gui* foi desenvolvido toda a interação com o utilizador, utilizando a biblioteca *Swing*.

Por fim, estes três *packages* associam-se entre si da seguinte forma:



### 3.1. Estrutura de Classes

A estrutura de classes encontra-se detalhada no ficheiro “Monpoly.eap” que se encontra junto deste relatório e do código do projecto.

## 4. Conclusão

Em síntese, ao longo de todo o trabalho foram encontradas várias dificuldades, nomeadamente, a elaboração do jogo em modo de rede que apesar de ter sido tentada implementar não foi conseguida com sucesso. Outras das dificuldades foram seguir a estrutura inicialmente planeada como se comprova pela diferença do UML apresentado na primeira entrega e do UML final, desenvolvimento de animações e por último dificuldades com os métodos “*repaint()*” e “*paintComponent()*”.

No entanto não deixo de realçar que grande parte destas dificuldades e o facto de o projecto não estar 100% funcional deveram-se ao facto de toda a carga de trabalhos e avaliações distribuídas do final do semestre não permitir concentração total em cada um dos projectos.

Para além deste ponto, outro dos motivos pelo qual estas dificuldades foram sentidas foi o facto de trabalhar sozinho.

No que toca a possíveis melhorias, e no seguimento dos pontos anteriores, estas passariam por adicionar o suporte de rede e animações ao jogo e também uma melhoria na interface com o utilizador.

Não obstante, penso que realizei um bom trabalho e que cumpri com grande parte a que me comprometi e a que o projecto obrigava.



## 5. Webgrafia

- Oracle,  
<http://docs.oracle.com/javase/7/docs/>
- Apontamentos das Aulas Teóricas  
<https://moodle.up.pt/course/view.php?id=1502>