

***MONOPOLY***

*Relatório final*

Laboratório de Programação Orientado a Objetos

2º ano MIEIC

Elementos do Grupo:

João Sousa – ei12042

Pedro Faria – ei12097

Índice

1. Introdução

O programa desenvolvido no âmbito da unidade curricular de LPOO (Laboratório de programação orientada a objectos), tem como objetivo reproduzir o famoso jogo de tabuleiro “Monopoly”.

O jogo tem como intuito simular o conceito de “monopólio”, sendo que a condição de terminação se verifica quando um dos jogadores domina toda a estrutura do jogo (capital financeiro, casas, hotéis, entre outros), fazendo com que os restantes jogadores não possam pagar as suas dívidas.

1. Manual de utilização

2.1 - Funcionalidades Suportadas

Durante o desenvolvimento do projeto, procurou-se implementar o máximo de funcionalidades, de modo a obter uma réplica o mais próxima possível do jogo em questão. Deste modo, este inclui praticamente todas as opções possíveis apresentadas no jogo oficial:

1. Estrutura básica
   1. Escolha do número de jogadores (2-4);
   2. Guardar e carregar jogos;
2. “In-game”

2.1- Lançamento virtual de dados através de números aleatórios;

2.2- Compra de propriedades;

2.3- Adição e remoção de edifícios;

2.4- Hipoteca de edifícios;

2.5- Verificação das propriedades associadas a cada jogador;

2.6- Término de turno;

* 1. – Formato de ficheiros

De forma a garantir uma experiência mais enriquecedora, foram adicionadas as opções “Save” e “Load”, que permitem guardar/carregar jogos. Como tal, foi utilizado o conceito de “Serialization” em Java, que consiste na “tradução” de objetos para um ficheiro *“.dat”* e que, posteriormente, poderão ser reutilizados.

2.3 - Modo de utilização

O programa apresenta uma interface simples e user-friendly, como se apresenta de seguida:

1. Conceção e implementação

3.1 - Estrutura de packages

|  |  |  |  |
| --- | --- | --- | --- |
| **Package** | GUI | Logic | Tests |
| **Função** | Contém as funções relativas à interface | Contém as funções relativas à lógica do jogo | Contém as funções de teste |

- ADICIONAR UML DE PACKAGES

3.2 Estrutura de classes

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Classe** |  |  |  |  |  |  |  |
| **Função** |  |  |  |  |  |  |  |

UML PARA CLASSES GOES HERE

* 1. Padrões de desenho

- ***Behavioural pattern***

- ***Memento***

*“Recognizable by behavioural methods which internally changes the state of the whole instance”*

O padrão de desenho “Memento” é utilizado na implementação da serialização, cujo intuito foi explicado acima (Fazer o “save” e “load” dos ficheiros). Este é utilizado na classe Game e em todas as classes que importam *“java.io.Serializable*”.

- ***Structural pattern***

- ***Decorator***

*“Recognizable by creational methods taking an instance of same abstract/interface type which adds additional behaviour”*

O padrão “Decorator” é utilizado em todas as subclasses de “*java.io.InputStream”, “OutputStream”,* nas quais se incluem “*java.io.FileInputStream*” e “*java.io.ObjectInputStream*”. As últimas duas classes referidas estão presentes na classe “MenuPanel.java”, sendo utilizadas na serialização dos objetos e na sua escrita para o ficheiro *“.dat”*.

* 1. Dificuldades encontradas

- interligação entre GUI e a lógica do jogo

- fazer com que o UML não sofresse muitas alterações quando comparado com o UML inicial

- implementação do desenvolvimento em rede, que foi mais tarde “abandonado”, optando assim por fazer animações no projeto

- dificuldades em apresentar toda a informações dos jogadores, tanto a nível de criatividade como a nível de implementação

1. Conclusão

4.1 Cumprimento dos objetivos / Possíveis melhorias

Indo de encontro, com as dificuldades encontradas, alguma da informação não se encontra visível na GUI da forma que seria mais intuitiva, para além da existência de pequenos erros que limitam certas funcionalidades do jogo. Sendo este, sem duvida um dos aspetos mais importantes a melhoras.

Um dos objetivos internos do projeto foi a implementação do jogo em rede. Visto esse objetivo não foi cumprido, optamos por adicionar animações ao jogo tornando-o mias divertido para o utilizador.

A nível da lógica do jogo, todas as regras fundamentais do *Monopoly* se encontram implementadas com sucesso, o que pode ser visível através dos testes realizados. Desta forma, a nível da lógica, a única coisa que achamos que poderia ser melhorada seria a otimização de algum código.

* 1. Contribuição

De uma forma geral, ambos os elementos do grupo contribuiram para a realização deste trabalho. Contudo, procurou-se distribuir de forma uniforme o volume de trabalho pelos elementos do grupo