Departamento Engenharia Informática e de Sistemas

Instituto Superior de Engenharia de Coimbra

Licenciatura em Engenharia Informática – Curso Europeu

Programação Avançada 2020/2021

Programação Avançada

Jogo Quatro-em-Linha

Carolina Figueiredo - 2018017653

Turma P1

Índice

1. Descrição de decisões

Parte Lógica

Na package lógica temos as packages dados, estados e memento, estas representam a parte lógica do programa. As classes estão dividas pelas packages respetivas, ou seja, estão separadas conforme o tipo. Sendo que na dados temos a informação base para o funcionamento do jogo, na estados estão presentes as classes relativas à Máquina de Estados e na memento estão as classes que correspondem ao Memento.

A utilização do Memento foi essencial para guardar a informação para voltar atrás no jogo, já que foi a maneira que achei mais acessível para implementar, utilizando o método que foi dado nas aulas.

A utilização da herança nos Minijogos foi a maneira mais eficaz que encontrei para utilizar dois tipos de jogos diferentes. Para alem destes, ainda temos o Dicionário que é o utilizado para o 2º Minijogo, o Jogador que guarda a informação sobre os jogadores e é utilizado pela classe Jogo. Estas são as classes que fazem parte de dados.

A Máquina de Estados separa os vários momentos do jogo no qual o utilizador tem de tomar decisões. Utiliza informação da classe Situação e dos estados. É onde se procede à ligação entre a parte lógica e parte UI.

Parte UI Texto

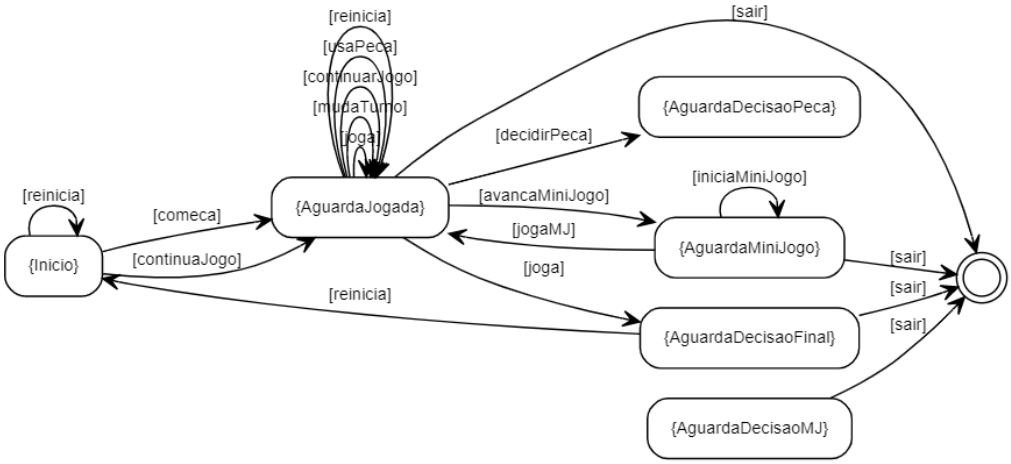
O início do programa dá-se na classe Main que assegura o início da UI de texto. A UI de texto é responsável pelo contacto com o utilizador e assegura a UI de todos os estados, onde são feitas questões, ou onde é passada a informação da parte lógica. O utilizador interage com UI e esta comunica à Máquina de Estados e vice-versa.

Parte Gráfica

A execução deste programa começa na classe JogoGalojfx.java, onde é criada uma Máquina de Estados e associada ao Jogo Observável, a qual tem acesso aos dados da Máquina de Estados. Esta classe fica responsável pela vista, a partir do PropertyChangeSupport da sinais as vistas para serem atualizadas.

Para alem desta iniciação temos também o PaneOrganizer que é responsável pela ligação entre o JogoObservavel e a classe PrincipalPane. Desta segue-se o JogoPane responsável pela vista de todos os estados.

2. Máquina de Estados



3. Descrição de Classes

jogo

Main:

Assegura o início da UI de texto e dá inicio à aplicação

jogo.logica.dados

Jogo:

Contém a lógica do jogo principal.

Jogador:

Contém informação dos utilizadores par utilização no jogo.

MiniJogo:

Classe Base para os Minijogos.

MiniJogo1:

Contém a lógica do minijogo 1 que consiste em cálculos.

MiniJogo2:

Contém a lógica do minijogo 2 que consiste em escrita de palavras.

Dicionário:

Utilizado pelo minijogo 2 e contem as palavras que este utiliza.

jogo.logica.estados

EstadoAdapter:

Classe abstrata, serve de adaptador de estados e contem os elementos que os estados vão utilizar

Inicio:

Estado responsável por iniciar o jogo.

AguardaJogada:

Estado responsável pela ação do jogo principal.

AguardaDecisaoMJ:

Estado que aguarda pela decisão do Mini jogo.

AguardaMiniJogo:

Estado responsável pela ação do Minijogo.

AguardaDecisaoPeca:

Estado responsável pela utilização da peça.

AguardaDecisaoFinal:

Estado responsável por reiniciar ou sair do jogo.

Fim:

Estado responsável pela finalização do jogo.

jogo.logica.memento

CareTaker:

Tem a responsabilidade de gerir e guardar os mementos

Memento:

objeto que guarda o estado interno do originator.

jogo.logica

MaqEstados:

Classe que gere os estados e comunica com a UI.

jogo.ui.texto

UItexto:

Responsável pela interação com o utilizador.

jogo.utilsUI

Utils:

Contem o scanner para facilitar a interação.

jogo.ui.gui

PrincipalPane:

Responsável pela inicialização da vista.

TabuleiroPane:

Responsavel pela vista do Tabuleiro.

JogoPane:

Responsavel pela vista do vários estados.

PaneOrganizer:

Base da Parte gráfica, faz ponte entre JogoGalojfx e PrincipalPane.

ConstantesGUI:

Contem as constantes necessárias a GUI.

Jogo.ui.gui.estados

AguardaDecisaoFinalPane:

Parte gráfica AguardaDecisaoFinal.

AguardaJogadaPane:

Parte gráfica AguardaJogada.

InicioPane:

Parte gráfica AguardaMiniJogo.

AguardaMiniJogoPane:

Parte gráfica AguardaMiniJogo.

Jogo.ui.gui.resources

ImageLoader e Resources:

Responsavel pelo load das imagens.

4. Descrição de Relacionamento entre Classes

jogo

Main:

UItexto

JogoGalojfx:

Extends Application

jogo.logica.dados

Jogo:

Jogador

MiniJogo1:

Extends MiniJogo

MiniJogo2:

Extends MiniJogo

Utiliza Dicionário

jogo.logica.estados

EstadoAdapter:

Implements IEstado

Inclui Jogo

Inicio:

Extends EstadoAdapter

AguardaJogada:

Extends EstadoAdapter

AguardaDecisaoMJ:

Extends EstadoAdapter

AguardaMiniJogo:

Extends EstadoAdapter

Inclui MiniJogo

AguardaDecisaoPeca:

Extends EstadoAdapter

AguardaDecisaoFinal:

Extends EstadoAdapter

Fim:

Extends EstadoAdapter

jogo.logica.memento

CareTaker:

IMementoOriginator

IMementoOriginator:

Memento

jogo.logica

MaqEstados:

Implements Serializable e IMementoOriginator

IEstado

Jogo

MiniJogo

CareTaker

jogo.ui.texto

UItexto:

MaqEstados

jogo.ui.gui

JogoPane:

Extends BorderPane

PaneOrganizer:

Extends BorderPane

PrincipalPane:

Extends BorderPane

TabuleiroPane:

Extends BorderPane

jogo.ui.gui.estados

JogoPane:

Extends BorderPane

PaneOrganizer:

Extends BorderPane

PrincipalPane:

Extends BorderPane

TabuleiroPane:

Extends BorderPane

AguardaDecisaoFinalPane:

Extends BorderPane

AguardaJogadaPane:

Extends BorderPane

InicioPane:

Extends BorderPane

AguardaMiniJogoPane:

Extends BorderPane

5. Funcionalidades e Regras do Jogo

Funcionalidades implementadas:

Implementação do jogo em modo texto;

Suporte para dois jogadores humanos, humano vs. computador ou computador vs. computador;

Opção de voltar atrás;

Funcionamento do jogo;

Ambos os minijogos;

Utilização da peça;

Logs do jogo em histórico;

Logs de 5 jogos em histórico;

Opurtunidade de reiniciar o jogo;

Gravação/carregamento do jogo;

UIGráfica;

Funcionalidades parcialmente implementadas:

Funcionalidades não implementadas:

Replay jogos anteriores.