Instituto Superior de Engenharia de Coimbra

Licenciatura em Engenharia Informática
Programação Avançada

Leandro Adão Fidalgo | a2017017144

Laboratório P3
Trabalho Prático 1
Meta 1

Coimbra, 23 de maio de 2021

Índice

1. Introdução

O presente relatório descreve o projeto desenvolvido pelo aluno: Leandro Fidalgo, no âmbito da disciplina de Programação Avançada da Licenciatura em Engenharia Informática do Instituto Superior de Engenharia de Coimbra.

A primeira meta do trabalho prático consistia na implementação de toda a lógica do jogo em modo texto, suporte para dois jogadores humanos, humano vs. computador ou computador vs. Computador e ainda gravação ou carregamento do jogo (um jogo carregado de ficheiro deve permitir a sua continuação).

Este trabalho consiste no jogo do 4 em linha com 2 mini jogos associados, um mini jogo dos cálculos e outro das palavras.

O objetivo do presente trabalho é consolidar todos os conhecimentos adquiridos nas aulas teóricas e práticas ao longo de todo o semestre.

2. Decisões tomadas na implementação

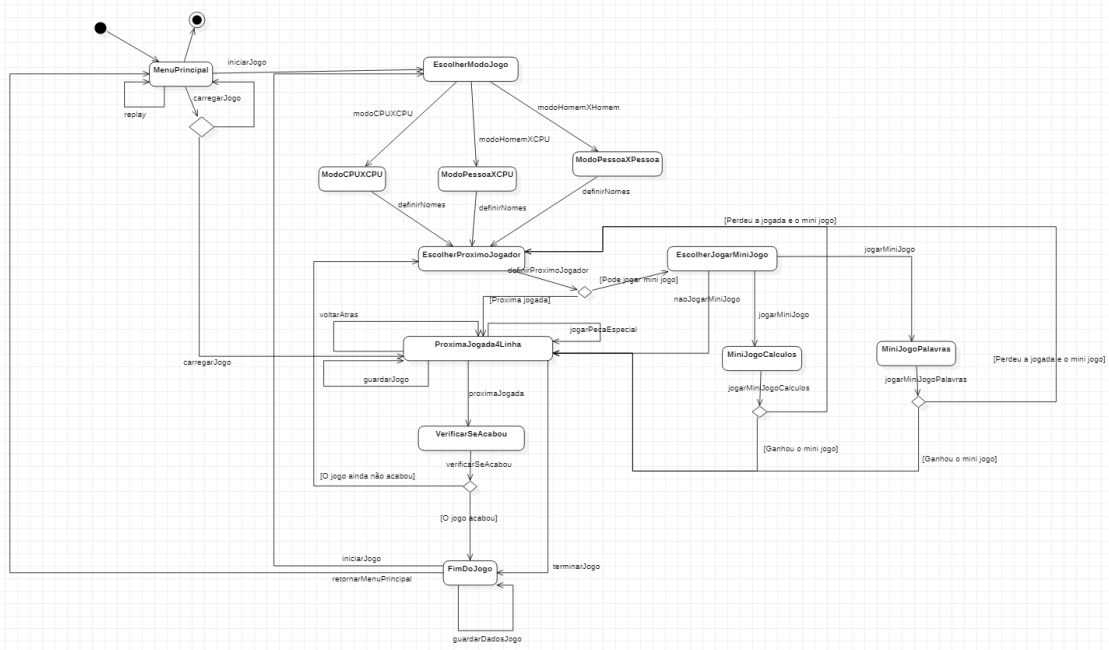
Durante a implementação do jogo foram surgindo algumas tomadas de decisão nomeadamente na forma de como é guardado o tabuleiro, foi optado por ser guardado num array bidimensional de inteiros, os Jogadores foram separados em Pessoa e CPU e foi ainda criada uma classe para os Minijogos para guardar informações relevantes aos mesmos.

Para voltar á jogada anterior durante o jogo foi utilizado uma Stack onde são guardados os tabuleiros ao longo do jogo e sempre que o jogador decide voltar á jogada anterior , dependendo do número de vezes que ele decide recuar é feito um pop á Stack.

Para os Replays foi implementada uma classe onde são guardadas pequenas informações que irão ser guardadas para depois mostrar quando o Replay for efetuado. Os Replays são guardados num HashMap onde a key é sempre a data atual quando foi guardado o Replay e o value é o Replay, sendo este HashMap serealizado para um ficheiro chamado “Jogos” onde estarão sempre as informações relativas ao Replay.

Para guardar e carregar o jogo também são usados ficheiros, onde os DadosJogo são serealizados para dentro do mesmo.

3. Diagrama da máquina de estados



3. 1. Estado MenuPrincipal

Ao iniciar o jogo o primeiro estado a surgir será o MenuPrincipal, logo aí o utilizador poderá tomar 4 decisões, uma delas será terminar o jogo e o jogo termina por completo, a outra será fazer o replay, sendo que ao fazer o replay volta para o mesmo estado, poderá ainda carregarJogo que significa carregar um jogo de um ficheiro, porém ao carregar o jogo poderá ocorrer um erro ou o ficheiro não existir e irá voltar ao MenuPrincipal, se não ocorrer nenhum erro o jogo irá seguir para o estado ProximaJogada4Linha e irá continuar a jogada onde este tinha sido guardado. Por fim pode iniciarJogo e ir para o estado EscolherModoJogo.

3. 2. Estado EscolherModoJogo

No estado EscolherModoJogo poderá escolher 3 opções CPUXCPU, PessoaXCPU, PessoaXPessoa, todos eles estão separados em estados diferentes como é possível verificar no diagrama.

3. 3. Estados ModoCPUXCPU, ModoPessoaXCPU, ModoPessoaXPessoa

Nestes estados a única implementação é a definição do nome dos jogadores.

3. 4. Estado EscolherProximoJogador

Neste estado, primeiro é alterado o próximo jogador que irá jogar e de seguida é verificado se este irá ter oportunidade de jogar o mini jogo ou não. Se tiver a oportunidade de jogar o mini jogo irá ser encaminhado para o estado EscolherJogarMiniJogo, senão irá ser encaminhado para o estado ProximaJogada4Linha.

3. 5. Estado EscolherJogarMiniJogo

Neste estado o utilizador poderá escolher jogarMiniJogo ou então naoJogarMiniJogo, ao escolher jogarMiniJogo o programa irá verificar se é a primeira vez que está a jogar os mini jogos e caso seja irá escolher um aleatoriamente um dos mini jogos e de seguida irá ser guardado o mini jogo escolhido

para mais tarde o próximo mini jogo não ser o mesmo e ao escolher naoJogarMiniJogo será encaminhado para o estado ProximaJogada4Linha que irá ser abordado mais tarde.

3. 6. Estado MiniJogoCalculos

No estado MiniJogoCalculos o utilizador será abordado para ir introduzindo o valor dos cálculos que lhe serão apresentados e caso o tempo já tenha ultrapassado os 30 segundos e o jogador não tenha acertado 5 vezes o mini jogo dos cálculos acaba e o jogador perde a vez de jogar e perde também o mini jogo. Caso o jogador acerte 5 vezes em menos de 30 segundos, recebe uma peça especial que poderá utilizar mais tarde e ainda tem possibilidade de efetuar uma jogada no 4 em linha.

3. 7. Estado MiniJogoPalavras

No estado MiniJogoPalavras é idêntico ao mini jogo dos cálculos simplesmente aqui o jogador terá de introduzir escrever as palavras que lhe serão projetadas no ecrã escolhidas aleatoriamente o mais rápido possível, tendo este período de correspondente á metade do número de caracteres existentes nas palavras, se perder perde a vez de jogar e perde também o mini jogo. Se ganhar recebe uma peça especial que poderá utilizar mais tarde e ainda tem possibilidade de efetuar uma jogada no 4 em linha.

3. 8. Estado ProximaJogada4Linha

Neste estado o jogador Pessoa poderá efetuar uma jogada no tabuleiro, efetuar a jogada de uma peça especial (para isso o jogador tem de ter pelo menos 1 peça especial), guardar o estado atual do jogo num ficheiro, para mais tarde o carregar se assim entender, ainda pode retroceder a jogada para uma ou até no máximo 5 jogadas anteriores e por fim o jogador poderá ainda terminar o jogo se assim o entender.

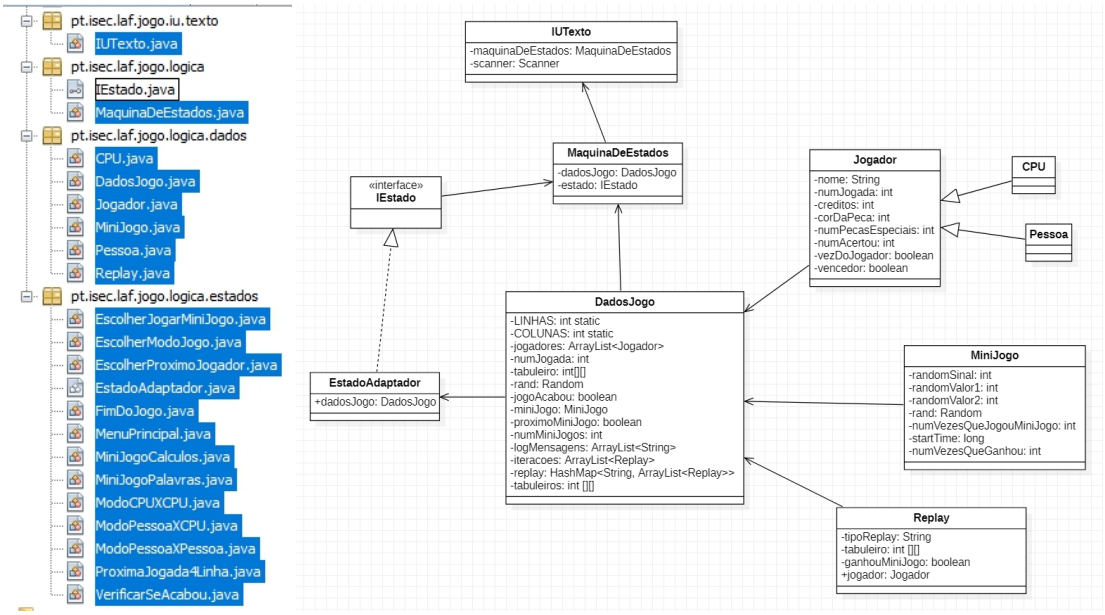
3. 9. Estado VerificarSeAcabou

Neste estado, o estado atual do tabuleiro é verificado e assim verifica se algum dos jogadores ganhou ou se o tabuleiro já se encontra totalmente preenchido.

3. 10. Estado FimDoJogo

Neste estado o jogador poderá guardar o jogo para mais tarde ver o seu replay, guardar o estado atual do jogo tal como no estado ProxiaJogada4Linha e por fim pode iniciar um novo jogo ou retornar ao estado MenuPrincipal, sendo todos estes passos questionados ao jogador por esta ordem.

4. Descrição das classes



4. 1. Classes relacionadas com a interface

Apenas existe uma classe relacionada com a interface sendo esta a Classe IUTexto e que tem toda a lógica relacionada com o que irá ser apresentado ao utilizador e relacionada com as introduções que o utilizador realiza. Esta classe contém a MaquinaDeEstados.

4. 2. Classes relacionadas com a lógica

A única classe relacionada com a lógica do jogo é a MaquinaDeEstados e ainda uma interface para auxiliar nos diversos estados com o nome IEstado. A MaquinaDeEstados tem toda a lógica referente á mudança dos estados e ainda incorpora os DadosJogo.

4. 3. Classes relacionadas com os dados

Nas classes relacionadas com os dados existem diversas. Porém a classe principal e que contém todas as informações é a classe DadosJogo, que contém um ArrayList de Jogador, que contém o MiniJogo e ainda o Replay. O CPU e a Pessoa são classes que fazem extend da classe Pessoa. Na classe DadosJogo é guardada toda a informação acerca do Jogo e esta classe é Serializable e todas as classes que estão nos DadosJogo são Serializable e mais tarde servirá para guardar os jogos.

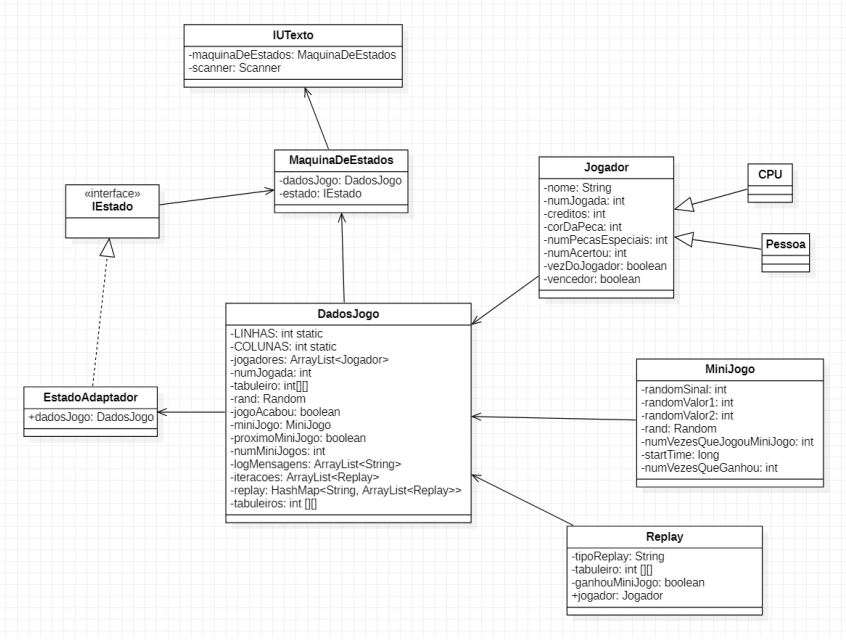
A classe MiniJogo serve para guardar informações acerca dos mini jogos, a classe Jogador serve para guardar os vários atributos dos jogadores e a classe Replay serve para guardar pequenas informações que serviram para fazer o replay de um jogo que estará guardado dentro de um ficheiro.

A classe DadosJogo ainda guarda os vários tabuleiros para quando é efetuada a operação de voltar á jogada anterior, contém o logMensagens que é onde são colocados os diversos logs e ainda contém um HashMap<String, ArrayList<Replay>> que irá ser usado para guardar os replays, sendo a String utilizada para guardar o nome do Replay e o ArrayList irá conter as várias informações que irão ser impressas ao utilizador quando fizer o replay.

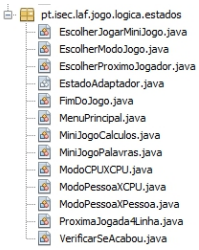
4. 4. Classes relacionadas com os estados

Nas classes relacionadas com os estados, a classe EstadoAdaptador implementa o IEstado e serve para fazer as diversas funcionalidades que irão ser implementadas nas classes dentro do mesmo package, sendo estas classes os estados que foram apresentados na máquina de estados.

5. Descrição do relacionamento entre as classes



No resumido diagrama de classes anterior é possível verificar a ligação entre classes. O EstadoAdaptador faz a implementação do diversos métodos que estão disponíveis na interface IEstado. Na MaquinaDeEstados é alterado o estado dependendo dos acontecimentos. Todos os estados ex tendem do EstadoAdaptador e implementam as funções que irão necessitar dependendo do que querem realizar. Os estados existentes estão demonstrados na figura abaixo exceto o EstadoAdaptador, estes estados não estão representados no diagrama pois tornava-se demasiado extenso e pouco percetível, porém todos eles aparecem no diagrama da máquina de estados.



6. Funcionalidades

Fases	Implementado	Parcialmente	Não implementado
Jogo do 4 em linha	**X**		
Preparado para jogador virtual ou humano	**X**		
Mini jogo cálculos	**X**		
Mini jogo palavras	**X**		
Peça especial	**X**		
Voltar á jogada(s) anterior(es)	**X**		
Replay de um jogo	**X**		
Guardar jogo	**X**		
Carregar jogo	**X**		
Lógica da máquina de estados	**X**		
Iniciar novo jogo	**X**		
Terminar jogo	**X**		
Registo de logs	**X**		

7. Conclusão

Com o desenvolvimento desta meta foi possível aprender muito sobre o conceito de máquina de estados entre outros conceitos abordados durante as aulas. Na realização desta meta não foi utilizado mementos nem comandos, logo não coloquei em prática esses conceitos.

Durante o desenvolvimento desta meta foram surgindo problemas e desafios que foram superados com a ajuda dos professores da disciplina, os apontamentos por eles disponibilizados e da Internet.