

André Luis Storino Junior - 9293668
Fabio Fogarin Destro - 10284667
Gabriel Baltazar Domingos - 10284945
Paulo André Carneiro - 10295304
Vitor H. Gratieri Torres - 10284952

MentalRun - Trabalho final POO
Profº Dr. Marcio E. Delamaro
SCC0103 - Programação Orientada a Objetos

ICMC - UNIVERSIDADE DE SÃO PAULO

26/06/2018

Sumário

1	Introdução	3
2	Descrição do projeto	3
2.1	Jogo MentalRun	3
2.2	Como jogar:	3
2.2.1	Caça-palavras	4
2.2.2	Encontre o Par	4
2.2.3	Encontre o único	4
2.2.4	Olhos de Águia	4
2.2.5	Ordem Crescente	4
2.2.6	Qual tem mais?	4
2.2.7	Sequência numérica	4
2.2.8	Todos iguais	5
2.3	Desenvolvimento	5
2.3.1	Classe BaseJogos	5
2.3.2	Classe Cliente, Servidor e ServerThread	5
2.3.3	Random	5
2.3.4	ThreadPontuacao	5
2.3.5	Inicio	6
2.3.6	Restante das Classes	6
3	Divisão do trabalho	6

1 Introdução

O trabalho final da disciplina Programação Orientada à Objetos, consiste na utilização e aplicações dos conceitos aprendidos em aula, tendo-se a liberdade de implementação de qualquer tipo de projeto. Este relatório possui a explicação do trabalho, a forma como utilizá-lo e como foi feita a implementação.

2 Descrição do projeto

2.1 Jogo MentalRun

O jogo, em sua totalidade, é composto por 8 mini-jogos que, juntos, formam uma corrida mental. O usuário deve inserir o seu nome e selecionar se deseja jogar sozinho ou em dupla. Quando o primeiro mini-jogo começa, começa-se a contar o tempo que o usuário demora para resolver os desafios até que tenha passado por todos os 8 mini-jogos. O critério para vencer se dá pela comparação, entre os usuários, do menor tempo gasto por cada um.

Os mini-jogos são:

1. Caça-palavras
2. Encontre o Par
3. Encontre o único
4. Olhos de Águia
5. Ordem Crescente
6. Qual tem mais?
7. Sequência numérica
8. Todos iguais

2.2 Como jogar:

Cada mini-jogo, embora muito semelhantes, tem jogabilidades e regras diferentes. Esta parte do relatório é responsável por explicar como cada um funciona assim como a forma de jogá-los. Uma regra comum a todos é quando acontece uma penalidade, ou seja, ao errar em algum dos mini-jogos. A penalidade consiste numa pause de 3 segundos até que o jogo retorne e o usuário possa continuar sua corrida.

2.2.1 Caça-palavras

Intuitivamente, consiste num caça-palavras onde o usuário deve selecionar, individualmente, as letras correspondentes a palavra dada. A ordem das letras podem estar na horizontal como também na vertical, vale lembrar que as palavras podem estar ao contrário. Terminadas as rodadas, o mini-game é trocado.

2.2.2 Encontre o Par

Diversos ícones são dispostos no tabuleiro, cabe ao usuário analisá-los e clicar nos semelhantes, caso sejam iguais, novos ícones são carregados e aumenta-se a quantidade disponível, com o intuito de aumentar a dificuldade do jogo. Vale lembrar que devem possuir cor e formato iguais.

2.2.3 Encontre o único

Diversos ícones são dispostos no tabuleiro, cabe ao usuário buscar pelo único ícone que não possui repetição e clicá-lo. A cada rodada os ícones são dispostos em outra configuração e a quantidade também aumenta.

2.2.4 Olhos de Águia

Com o tabuleiro completo e disposição alinhada, o usuário deve ser capaz de identificar um ícone que, embora se pareça muito com os seus vizinhos, é diferente. A dificuldade começa após algumas rodadas, quando a fonte diminui a sua espessura.

2.2.5 Ordem Crescente

Com o tabuleiro completo de números e disposição alinhada, o usuário deve clicar nos números em ordem crescente, caso erre, uma penalidade é aplicada.

2.2.6 Qual tem mais?

Dois ícones são dispostos no tabuleiro de forma aleatória e com repetições, cabe ao usuário selecionar qual possui menos para convertê-los no que, supostamente, possui mais. A cada rodada a quantidade de ícones aumenta, com o intuito de aumentar a dificuldade.

2.2.7 Sequência numérica

Diversos números são dispostos aleatoriamente no tabuleiro, cabe ao usuário clicar, formando uma sequência, nos números que julgar participarem dela.

2.2.8 Todos iguais

Diversos ícones, diferentes ou não entre si, são dispostos aleatoriamente no tabuleiro. O usuário deve ser capaz de, ao clicar diversas vezes num ícone, deixá-lo com o formato desejado para, por fim, deixar todos os ícones da rodada iguais.

2.3 Desenvolvimento

O trabalho foi desenvolvido utilizando-se a linguagem de programação orientada a objetos Java juntamente com a IDE Eclipse. Para que o jogo funcionasse de forma correta, foram criadas 16 classes que compõe o projeto: BaseJogos, CacaPalavras, Cliente, EncontrePar, EncontreUnico, Inicio, OlhoDeAgua, OrdemCrescente, QualTemMais, Random, SequenciaNumerica, ServerThread, Servidor, ThreadPontuacao, TodosDiferentes, TodosIguais. Para o desenvolvimento, diversas bibliotecas foram utilizadas, e a interface gráfica foi feita utilizando-se o Swing, conforme aprendido em aula.

2.3.1 Classe BaseJogos

Principal classe do projeto, nela que está a base de todos os mini-jogos implementados, como o contador de tempo, o formato que o tabuleiro deve ter, os botões e a parte gráfica. Sua importância se dá pelo fato de ser uma classe abstrata a ser estendida pelas demais classes do projeto.

2.3.2 Classe Cliente, Servidor e ServerThread

Essas classes são responsáveis por fazer a interface online do jogo, que mantém pontuações e manipula a comunicação entre os jogadores. São responsáveis para interpretar comandos e tratar erros de conexão. A classe Cliente possui um *addShutdownHook* que é responsável por fazer a desconexão através da comunicação com o servidor para fechar os canais de comunicação entre os computadores.

2.3.3 Random

Classe responsável por retornar um número aleatório, extremamente utilizada na implementação da lógica dos mini-jogos. Possui uma semente que é modificada conforme a necessidade do uso da classe.

2.3.4 ThreadPontuacao

Classe responsável para ler e retornar as saídas do terminal.

2.3.5 Início

Classe responsável para o funcionamento do jogo, seja ele online ou não. Nela são chamados os mini-jogos e toda a interface que faz com que tudo esteja integrado, como a parte lógica dos jogos e a interface gráfica.

2.3.6 Restante das Classes

As demais classes do projetos são as implementações lógicas dos mini-jogos que o projeto possui. Utilizam a classe abstrata *BaseJogos* como base para o que foi implementado.

3 Divisão do trabalho

O projeto foi dividido entre os membros do grupo na seguinte proporção:

André Luis Storino Junior - 20%

Fabio Fogarin Destro - 20%

Gabriel Baltazar Domingos - 20%

Paulo André Carneiro - 20 %

Vitor H. Gratieri Torres - 20%