

Trabalho Prático II - POO

Luciano Belo - 3897, Guilherme Aguiar - 3496

May 11, 2021

Contents

1	Definições Gerais	2
1.1	Modelo	2
1.2	Controller	2
1.3	View	2
2	Single Player	3
3	Player vs Computador	3

1 Definições Gerais

O jogo da memória pode ser jogado por um único jogador ou um jogador contra a máquina. O jogo começa colocando todas as cartas viradas para baixo. A jogada é quando um jogador vira duas cartas e coloca-as para cima. Se o jogador virar duas cartas que não correspondem, ambos as cartas devem ser virados para baixo novamente no mesmo local. Se o jogador vira um par de cartas que coincidem em uma jogada, o jogador ganha o par de cartas, aumentando seu *score*. O objetivo do jogo é virar o maior número de pares de cartas possível. O vencedor é o jogador que reúne o maior número de pares, ou seja, com maior *score*. No modo de único jogador, o objetivo é identificar todos os pares.

O trabalho em questão foi desenvolvido com a API Swing com o tema "Computação", sendo assim as imagens dos cards correspondem à tecnologias, Frameworks, Linguagens de programação e áreas da ciência da computação.

1.1 Modelo

Na seção modelo são implementadas as seguintes classes:

1. Card → Classe para representar as cartas bem como responder aos cliques do mouse;
2. User → Classe que implementa funcionalidades para o jogador. Essa classe é usada em uma relação de herança com o objeto máquina.
3. Machine → Classe que implementa as funcionalidades para a máquina. É classe filha da classe User.

Cada card do jogo possui a interface *MouseListener*, que tem como função saber sobre as atividades e eventos do mouse como Mouse Clicks, Mouse Pressed, Mouse Released e Mouse Exited. É a partir da captura dos eventos em que é possível virar os cards e assim fazer o jogo acontecer.

1.2 Controller

Na seção de controller são implementados dois controllers básicos:

1. CardController → Implementa os controles do jogador perante as cartas. Verificando a validade dos pares e aferição de vitória, esse controle é presente tanto no modo multiplayer quanto no single player.
2. GameController → Implementa o controle que integra a máquina com o jogo. Nesse controle são implementadas funções que auxiliam a máquina a escolher as cartas que devem ser viradas. O controle em questão é presente apenas no modo multiplayer.

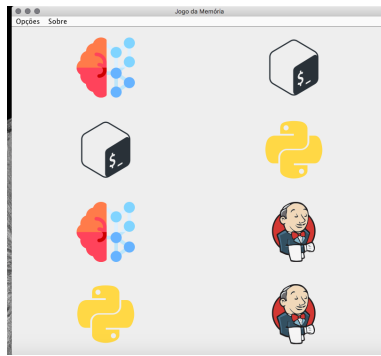
1.3 View

Foram implementadas 3 telas principais. A primeira dela é a Options, que integra os dois tipos de jogos. Cada um dos modos não tem correlação com o outro. Ambas as telas de jogo são aptas para receber comandos via tela por meio de cliques no mouse, portanto usam os parâmetros de Listen do java. Em todas as telas são implementados os menus superiores para as seguintes opções:

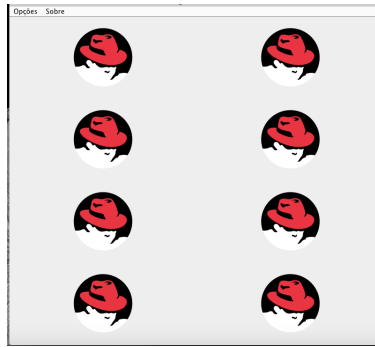
- Opções
 1. Novo jogo → Carrega um novo jogo zerando todas as configurações e placares computados.
 2. Sair → Simplesmente encerra a aplicação.
- Sobre → Apresenta algumas informações sobre o jogo da memória.

Nas telas dos jogos são implementados um JFrame que abriga um grid de 4 linhas e 2 colunas de cartas, no qual cada carta ocupa uma posição de coordenada definida para que não haja sobreposição.

Os valores de score não são exibidos para o usuário. O estado do jogo é informado ao fim da partida para o usuário indicando qual é o vencedor e no caso de empate também há tal informação. Essas exibições são feitas via pop-up. As telas do jogo podem ser vistas na figura abaixo. Da esquerda para direita, a foto 1 exhibe um tabuleiro com jogo em progresso, 2 exhibe um tabuleiro sem escolhas e 3 exhibe o menu de opção de jogo. Para trocar a opção de jogo é necessário que o usuário saia do programa e inicie o mesmo novamente.



(a) label 1



(b) label 2



(c) label 2

Figure 1: 2 Figures side by side

2 Single Player

No modo de jogo Single Player o jogador consegue jogar até que acabe as cartas disponíveis para escolha, ou seja, já tenha "adivinhado" todas as combinações. Sendo assim, basta que o usuário clique na carta desejada, para que ela vire. Após ter escolhido, caso o par de cards sejam iguais, eles irão permanecer virados, caso contrário as duas cartas escolhidas irão virar para o verso novamente.

Assim que o jogo é finalizado, uma mensagem aparece na tela informando que o jogador venceu a partida. Para jogar novamente, basta ir nas opções do menu e escolher o item "Novo Jogo", caso queira fechar a aplicação, é disponível também nas opções o item "Sair".

3 Player vs Computador

No modo de jogo Player vs Computador, são divididos turnos, sendo que cada turno termina após a escolha de dois cards. A dinâmica do jogo é a mesma do Single Player a diferença é que no primeiro turno começa pelo Player, depois o computador e assim por diante.

Contúdo, uma vez que o computador tem uma memória de maior confiança que o jogador caso ele tenha sempre o start terá uma chance de vitória expressiva, fazendo com que o ritmo do jogo não seja favorável, uma vez que existem apenas 4 pares a serem descobertos. Por essa razão o computador não avança ao ganhar um par, apenas o jogador pode fazer isso. A dupla optou por essa implementação a fim de tornar a disputa um pouco mais justa.

O computador, Jarvis, é uma classe filha da classe usuário, portanto possui atributos semelhantes como o armazenamento de inteiros, por exemplo. Ademais, a implementação do Jarvis conta com duas listas de cartas. Uma lista de todas as cartas do jogo e uma lista apenas das cartas que já foram exibidas. A primeira lista é usada para sortear uma nova abertura e a segunda é usada para identificar possíveis compatibilidades.

O computador nunca abrirá uma carta fechada imediatamente, sempre será escolhida uma carta fechada aleatoriamente. Uma vez escolhida uma carta aleatória, verifica se há uma correspondente em um vetor auxiliar que armazena todas as cartas já exibidas. Portanto ao passo que o jogo segue o computador constrói uma boa tática para vencer.

Assim que o jogo é finalizado, uma mensagem aparece na tela informando quem é o vencedor da partida. As opções para jogar novamente e sair funcionam igualmente para este modo de jogo.

Para acessar o placar, é necessário que após o termino do jogo, clique em "opções" e em placar.

References

- [1] Meyer, Bertrand. Object-oriented software construction. Vol. 2. Englewood Cliffs: Prentice hall, 199.
- [2]Sabharwal, Chaman L. "Java, java, java." IEEE Potentials 17.3 (1998): 33-37.
- [3]Zukowski, John. The definitive guide to Java Swing. Apress, 2006.
- [4]Thakur, Ram Naresh, and U. S. Pandey. "The Role of Model-View Controller in Object Oriented Software Development." Nepal Journal of Multidisciplinary Research 2.2 (2019): 1-6