

UNIVERSIDADE FEDERAL DE VIÇOSA – CAMPUS FLORESTAL

MATHEUS FREITAS MARTINS – 3031

**RELATÓRIO REFERENTE A ATIVIDADE AVALIATIVA 04 -
Trabalho Prático 4 – Jogo de Memória Genius**

**Florestal
2019**

MATHEUS FREITAS MARTINS – 3031

**RELATÓRIO REFERENTE A ATIVIDADE AVALIATIVA 04 -
Trabalho Prático 4 – Jogo de Memória Genius**

Trabalho prático da disciplina
PROGRAMAÇÃO ORIENTADA A
OBJETOS – CCF 313, apresentado ao
curso de Bacharelado em Ciência da
Computação da Universidade Federal de
Viçosa – Campus Florestal.

Orientador: Fabrício Aguiar Silva

Florestal

2019

SUMÁRIO

1- Introdução	3
2 - Metodologia	3
3 - Decisões de projeto	4
4 - Funcionamento do software	5
5 - Conclusão	15

1- Introdução

O objetivo deste trabalho é o desenvolvimento de um jogo de memória que foi muito popular na década de 80, chamado de Genius. O objetivo do jogador é memorizar e repetir a sequência de cores que são compostas por 4 cores (vermelho, azul, amarelo, verde).

O sistema deve implementar, as seguintes funcionalidades: botão para iniciar/resetar o jogo, exibir a sequência de cores, informar acerto ou erro da sequência, sendo que o jogador deve escolher uma cor dentro de 5 segundos a cada escolha.



Fig. 1.0: Arquitetura do jogo Genius.

2 - Metodologia

Para este respectivo trabalho foi seguido o padrão de software (MVC) Model-View-Controller, que isola as regras de negócios da lógica de apresentação, a interface com o usuário. Possibilitando a divisão do projeto em camadas muito bem definidas para suas classes.

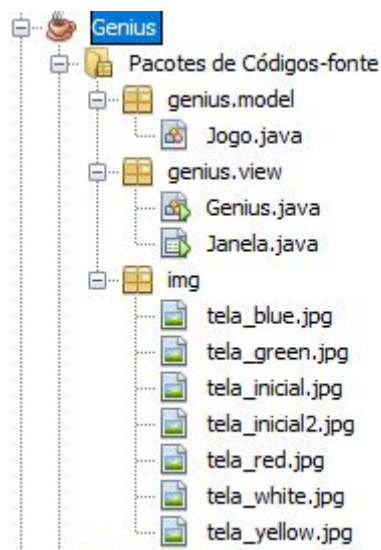


Fig. 2.0: Exibição da estrutura do projeto “Genius”, com as classes organizadas.

genius.model: É o coração da execução, responsável por tudo que a aplicação vai fazer a partir dos comandos da camada de controle em um ou mais elementos de dados, respondendo a perguntas sobre o sua condição e a instruções para mudá-las.

genius.view: Não se dedica em saber como o conhecimento foi retirado ou de onde ela foi obtida, apenas trata da exibição na interface do usuário.

img: imagens em “.jpg” das telas do jogo.

3 - Decisões de projeto

Foram utilizados estruturas de vetores que contém o armazenamento do ArrayList de sequência de cores gerada de forma aleatória e o Vetor em que é armazenado as escolhas do jogador. Foi necessário a implementação de 2 Threads para cuidar do tempo de espera limite de 5 segundos e o tempo de transição de cada sequência de cor gerada. O tratamento de erros e exceções foi feito por

condições de if-else, tendo em vista um funcionamento robusto que permite executar até quando o usuário desejar.

4 - Funcionamento do software



Fig. 4.0: Menu principal da aplicação do jogo Genius.



Fig. 4.1: Sequência de cores a ser escolhida. **Padrão: 8**, Mínimo: 1, Máximo: Sem limites.



Fig. 4.2: Cor vermelha (aleatória) sendo exibida logo após iniciar o jogo pelo botão “Jogar”.



Fig. 4.3: Exibição da cor azul selecionada de forma aleatória pelo jogo ou escolhida pelo jogador.

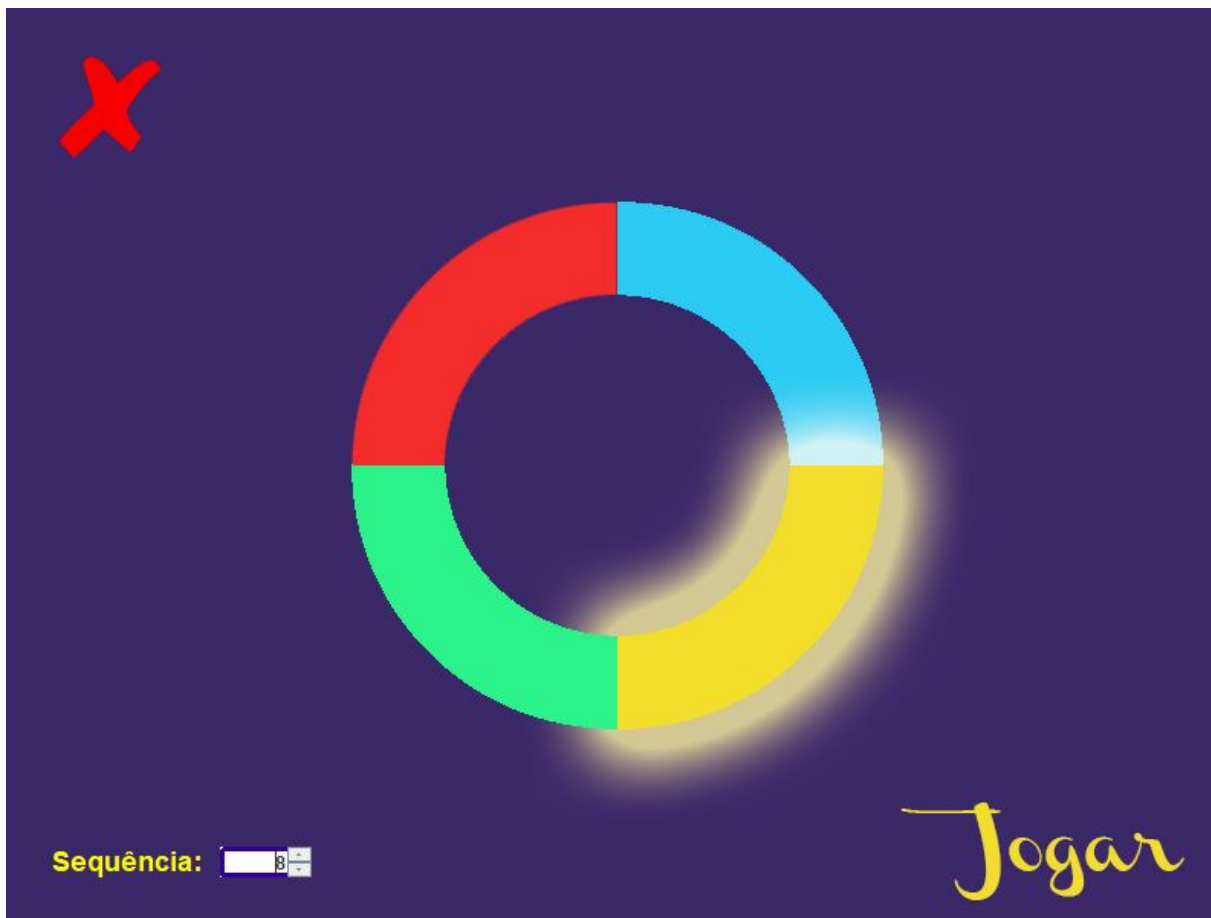


Fig. 4.4: Exibição da cor amarela selecionada de forma aleatória pelo jogo ou escolhida pelo jogador.

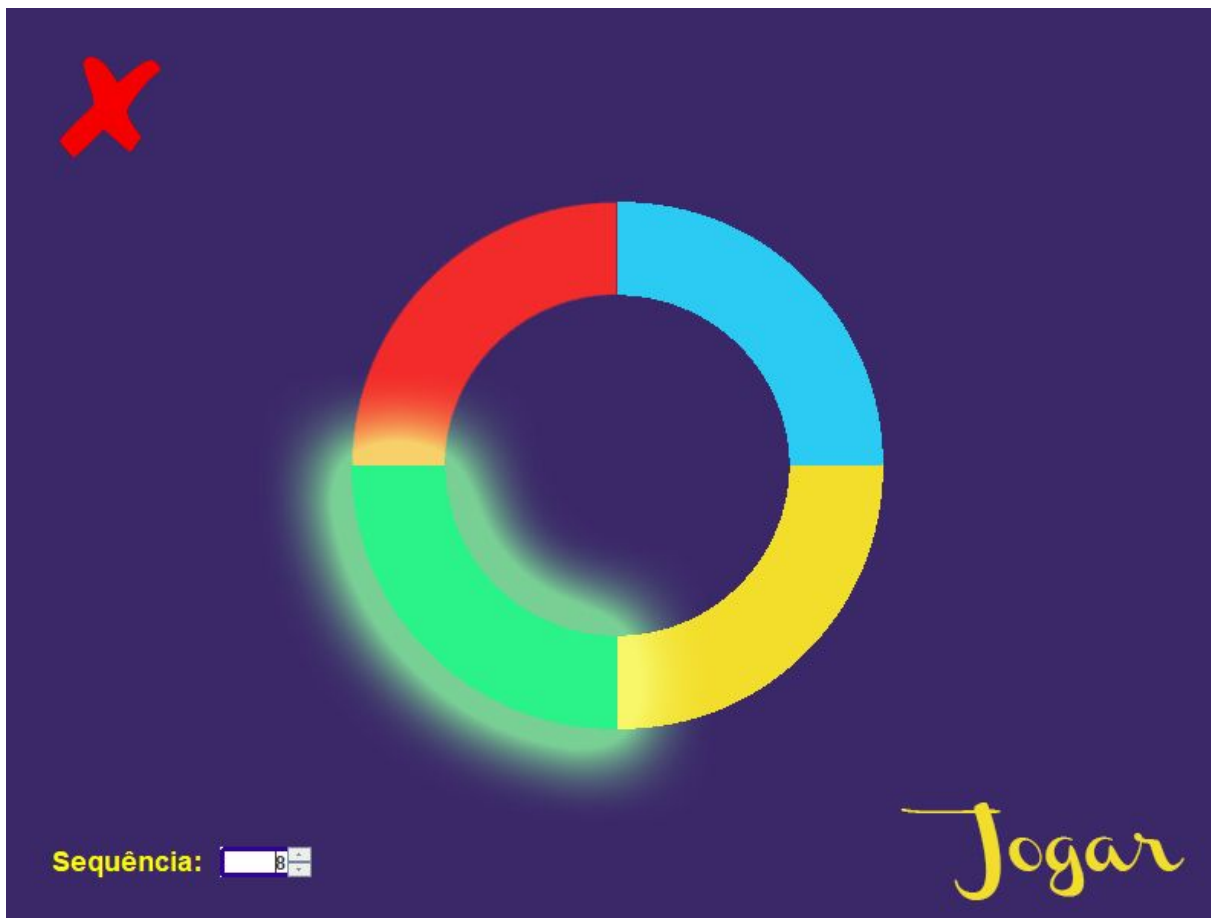


Fig. 4.5: Exibição da cor verde selecionada de forma aleatória pelo jogo ou escolhida pelo jogador.

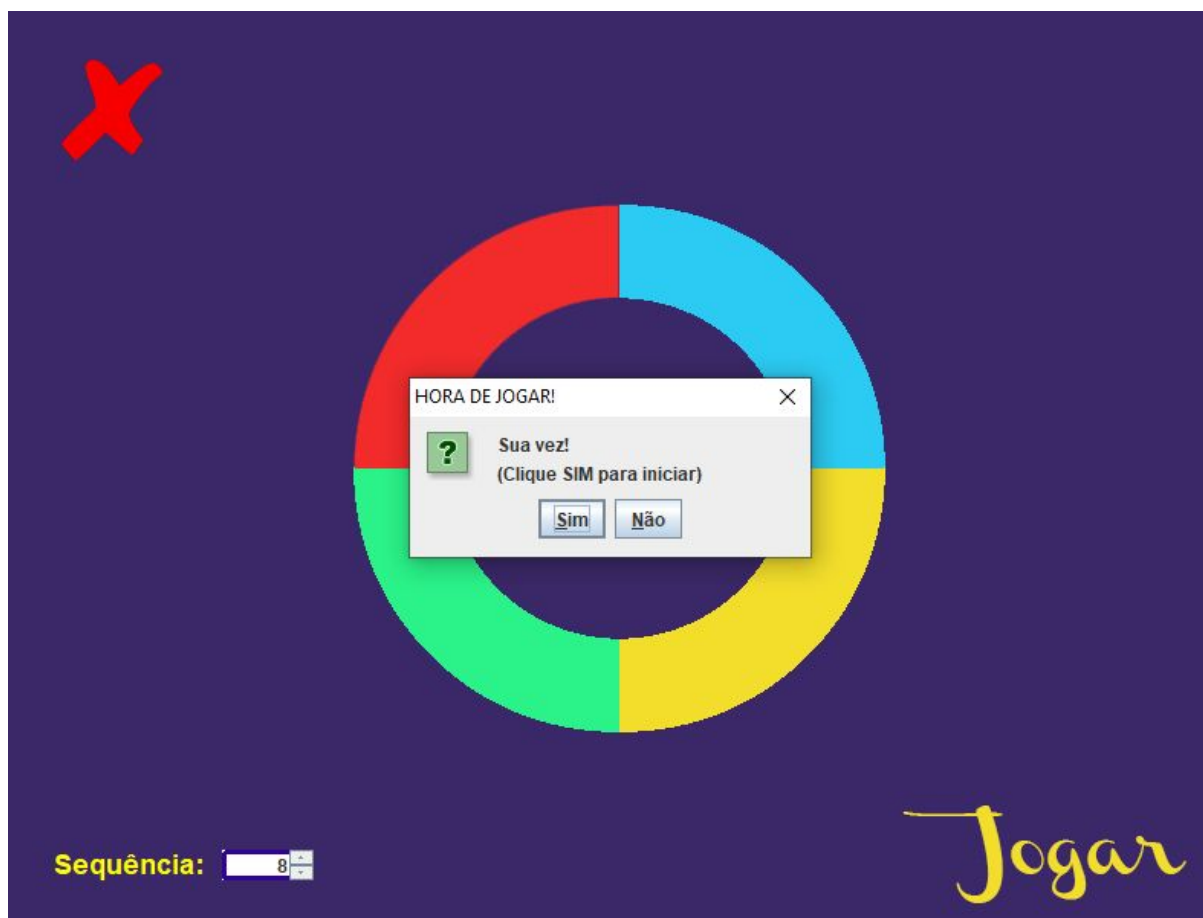


Fig. 4.6: Ao terminar a sequência de exibição das cores para o jogador, a aplicação pergunta ao jogador se ele deseja iniciar, caso queira, o tempo de aguardo de 5 segundos irá começar a ser contado. Se selecionar “Não” o jogador deve clicar em **Jogar** para obter uma nova sequência.

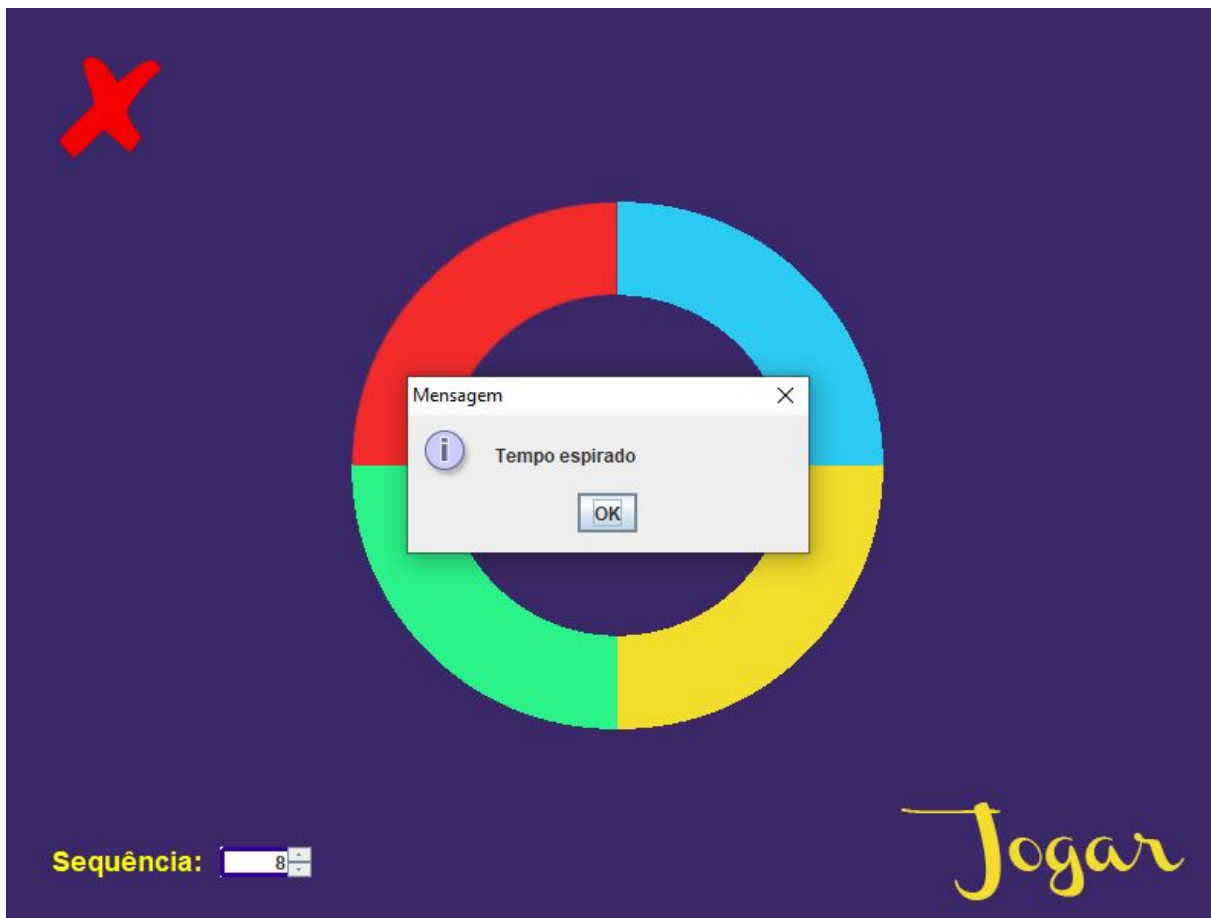


Fig. 4.7: Caso o jogador não selecione nenhuma cor ao iniciar o jogo ou após selecionar alguma cor dentro de **5 segundos**, uma mensagem de tempo expirado é exibida.

```
private final long MAX_TEMPO_ESPERA = 5000;
```

Fig. 4.8: Tempo definido por uma constante no código em 5000 milissegundos (5 segundos).

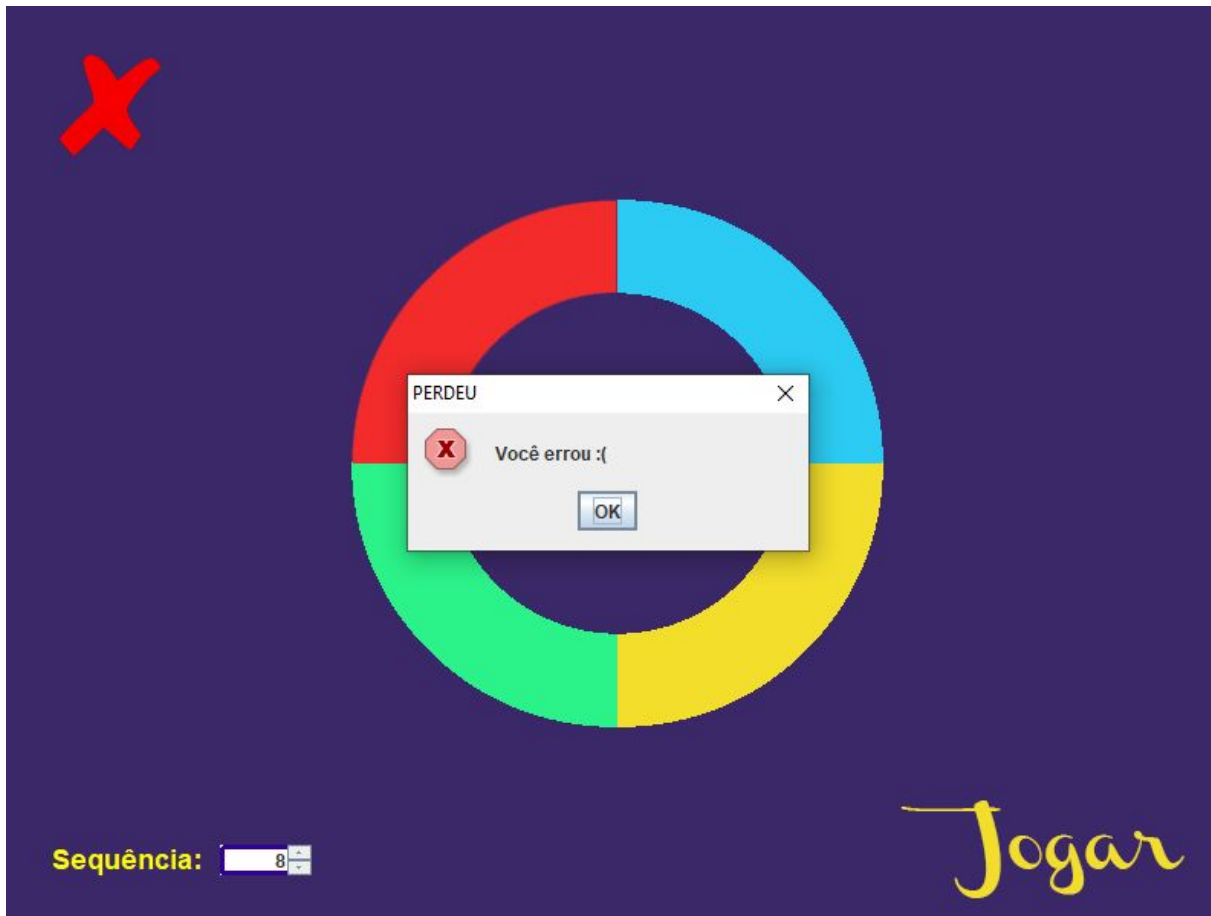


Fig. 4.9: Caso o jogador erre a sequência em alguma determinada posição, uma mensagem de erro é exibida instantaneamente indicando que errou.

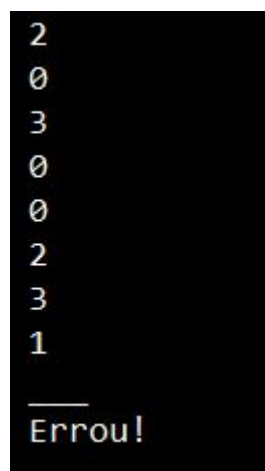


Fig. 4.10: Exibição do erro pelo console.



Fig. 4.11: Caso o jogador acerte todas sequências de cores exibidas, o jogador é parabenizado ao concluir.

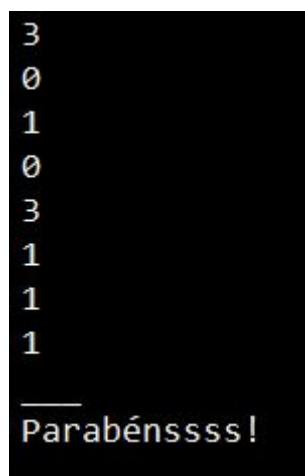


Fig. 4.12: Exibição do acerto das sequência pelo console.

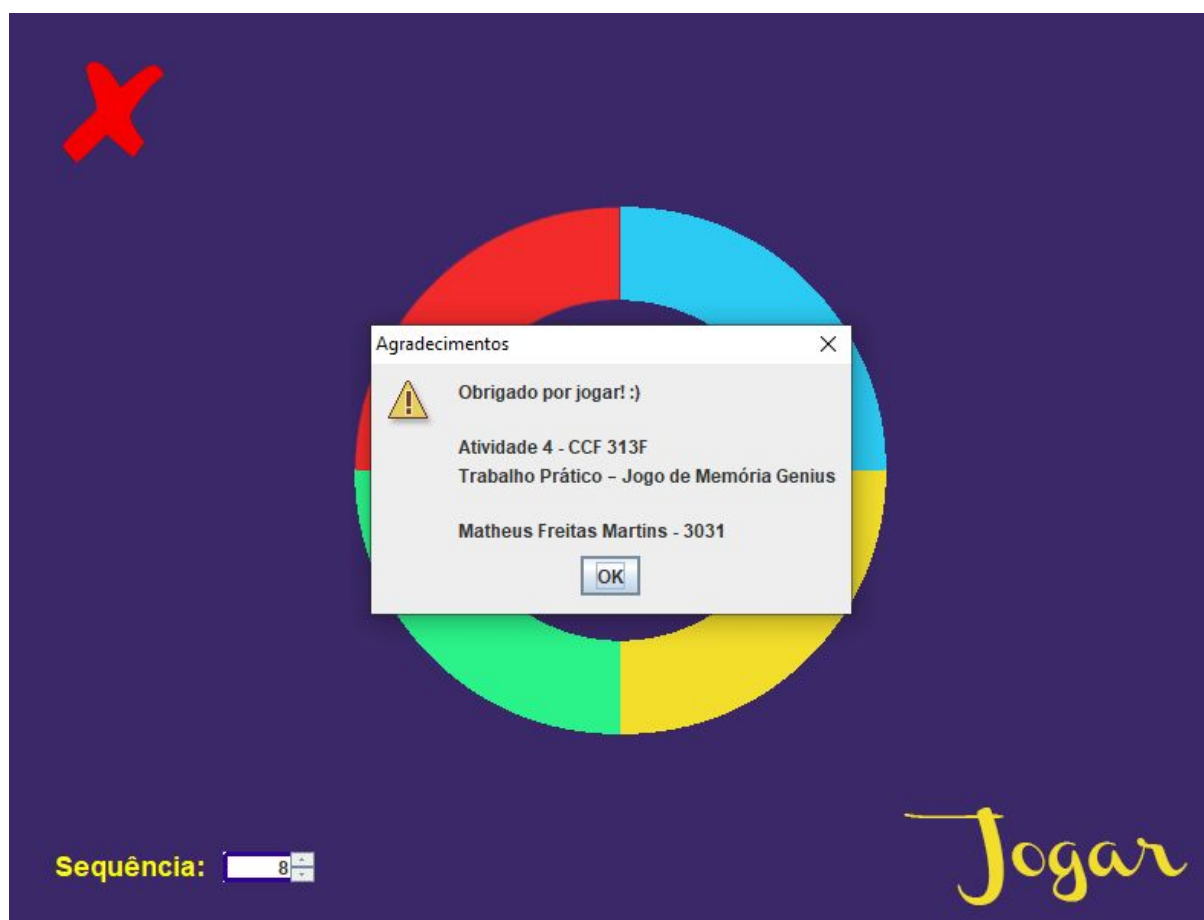


Fig. 4.13: Exibição de agradecimentos por usar o software ao clicar no botão vermelho “X”.

5 - Conclusão

Fica evidente, portanto, que a proposta da Atividade Avaliativa 4 foi atendida plenamente, o desenvolvimento do software na linguagem Java, utilizando a orientação objetos, pode-se obter o aperfeiçoamento da mesma, possui seus pacotes devidamente organizados, inclui também o tratamento de possíveis erros que um determinado usuário possa eventualmente cometer através de estruturas de comparação.