Programação Orientada a Objetos

Trabalho Prático - Jogo de Memória Genius

Arthur Marciano Pires - 3019
André Elias - 3013

Florestal, 25 de novembro 2019

SUMÁRIO

1. INTRODUÇÃO

Este trabalho consiste na implementação do jogo da memória chamado Genius. O jogo busca estimular a memorização de cores. Possui botões coloridos (azul, vermelho, amarelo e verde) que se iluminam em seqüência. Cabe aos jogadores repetir o processo sem errar. O desenvolvimento foi feito utilizando a linguagem de programação Java.

2. DESENVOLVIMENTO

A interface desenvolvida nesse sistema possui um botão de iniciar e outros quatro botões das cores azul, amarelo, vermelho e verde.

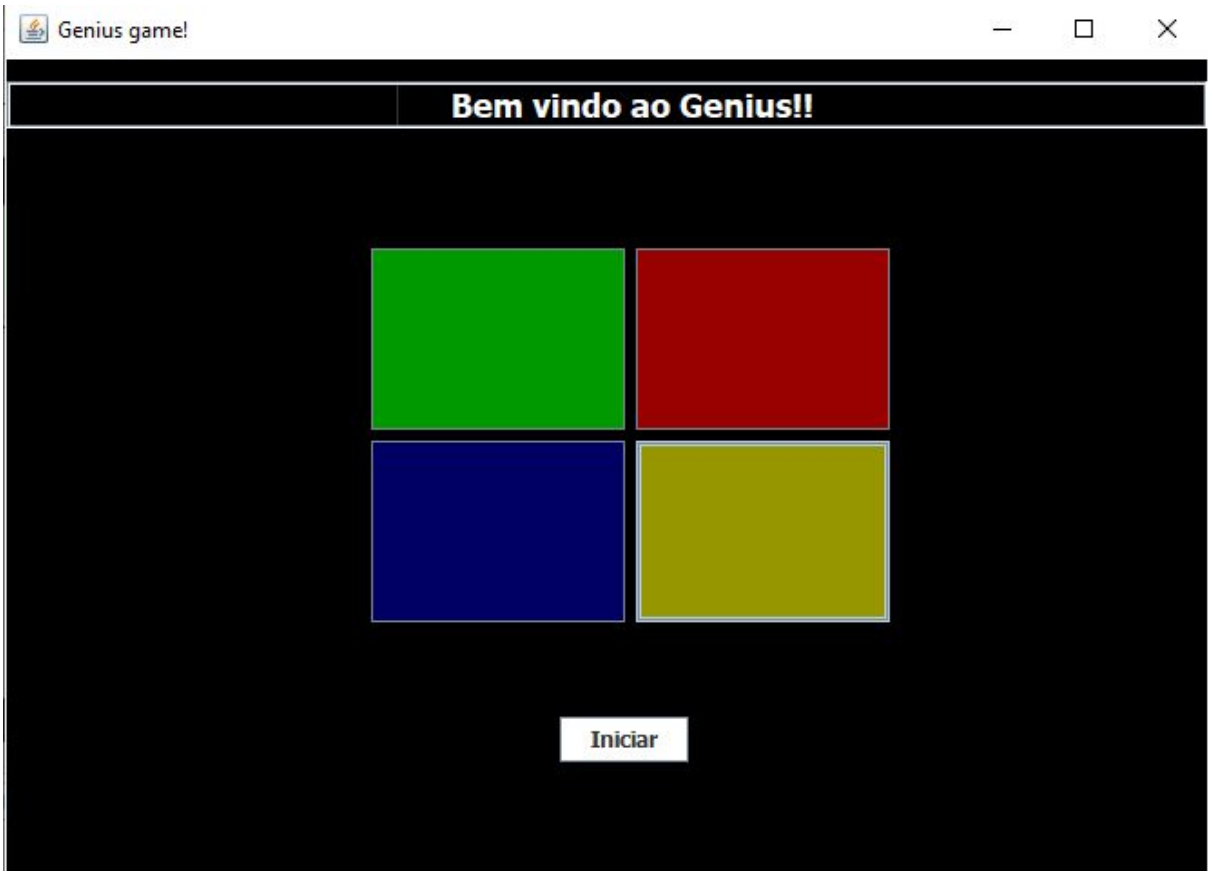

Imagem 1: Interface do jogo

Basta o usuário clicar em iniciar para começar o jogo. O jogo começa piscando uma luz aleatória, e o usuário tem cinco segundos para clicar no botão que corresponde à cor piscada. Caso ele exceda o tempo máximo ou clique no botão errado uma mensagem de erro será exibida e o jogo será zerado.

A imagem a seguir exemplifica o que foi falado:

Imagem 2: Exemplo de quando o usuário perde

Caso o jogador acerte toda a sequência de oito cores uma mensagem de parabéns será exibida e o jogo também será zerado.
A imagem abaixo retrata o que ocorre em caso de vitória:

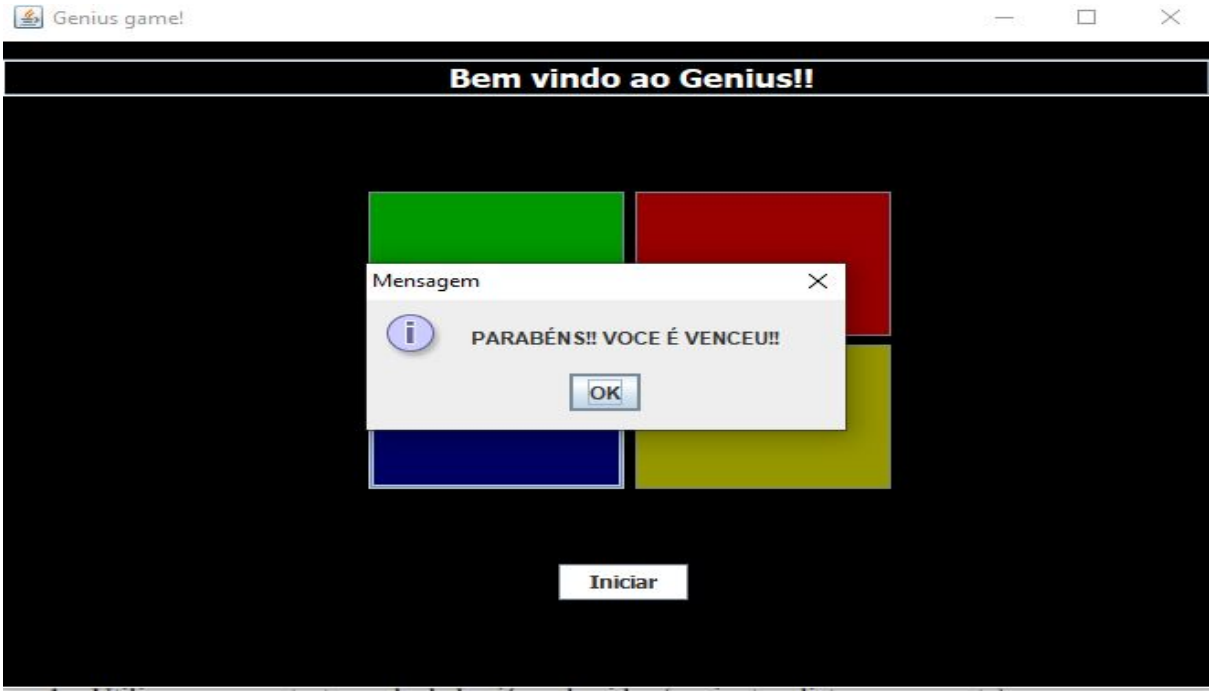

Imagem 3: Exemplo em caso de vitória

2.1. Decisões tomadas

Na implementação do sistema, certas decisões tomadas tiveram uma certa importância.

Para fazer os botões piscarem utilizamos threads que servem para executarmos duas ou mais tarefas simultaneamente. A função foi implementada em runnable, que permite que seja executado em sequência. A imagem abaixo mostra um exemplo da implementação:

```
public Runnable acendeVermelho(){

    Runnable r = new Runnable(){
        @Override
        public void run(){
            try {
                Thread.sleep(500);
                BotaoVermelho.setBackground(Color.red);
                Thread.sleep(500);
                BotaoVermelho.setBackground(new java.awt.Color(153, 0, 0));

            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    };
    return r;
}
```

Imagem 4: Exemplo do uso de threads

Para fazê-los piscarem em sequência uma outra função foi criada, essa função possui um loop com o level do jogo, que será a quantidade de vezes que os botões serão piscados. O botão a ser piscado é escolhido randomicamente utilizando a função “Random()”.

```
public List<Botao> criarLista(){
    int num;
    Random random = new Random();

    for (int x = count;x < lvl; x++){
        num = random.nextInt(4);

        switch(num){
            case 0:
                lista.add(novoBotao1);
                break;
            case 1:
                lista.add(novoBotao2);
                break;
            case 2:
                lista.add(novoBotao3);
                break;
            case 3:
                lista.add(novoBotao4);
                break;
```

Imagem 5: Função que seleciona os botões randomicamente

3. CONCLUSÃO

Este trabalho foi importante para o aperfeiçoamento e revisão de conceitos vistos anteriormente em sala de aula. Além disso nos proporcionou o aprendizado de novas coisas como uso de threads para o monitoramento de ações, como piscar um botão. Concluímos portanto que houve um bom aproveitamento e ganho de conhecimento por parte dos integrantes do grupo.