

## **Descrição da Proposta e Diagrama de Classe**

Álvaro Luís Silva Peixoto – 202100011190

Caio Vinicius da Cruz Coelho –

202100011243

João Victor Rocha Souza Rodrigues Moiteiro – 202000024409

### **Proposta de Implementação: Jogo da Memória**

O Jogo da Memória se baseia em dispor um número de par de cartas no “tabuleiro”, todas elas viradas para baixo, ou seja, no programa estarão inicialmente com seu conteúdo oculto. O objetivo do jogo é formar todos os pares com as cartas dispostas na tela, exibindo duas cartas por vez ao clicar nelas. Caso as cartas viradas formem um par, elas permanecem com seu conteúdo exibido e o jogador segue procurando os próximos pares, caso contrário, elas são novamente ocultadas e cabe ao jogador memorizar sua posição para formar os pares nos movimentos seguintes. Esse processo se repete até que todos os pares sejam encontrados, o que vai indicar a vitória do player.

### **Detalhamento: Classes Definidas**

Para o jogo, inicialmente foram definidas onb classes, são elas: Jogo, Menu, Player, Tabuleiro e Peça. Entretanto, com o desenvolver do projeto, foi finalizado com as seguintes classes: Game, Menu, Player, GameResult, ButtonHandler, Board, Piece, ImageCompressor, Stopwatch, VictoryScreen e DefeatScreen.

Herança: No nosso projeto utilizamos a herança entre as classes GameResult, VictoryScreen e DefeatScreen, ao projetar as classes, observamos que tanto VictoryScreen quanto DefeatScreen tinham diversas partes com código semelhante, então decidimos criar a classe pai GameResult, onde nós concentramos as partes repetidas o código e deixamos nas classes filhas apenas as partes específicas delas, além de criar um método show() específico para cada um.

Game: Classe que une todas as outras classes, com o intuito de inicializar o jogo.

Menu: Classe responsável pela exibição do menu principal do jogo.

Player: Classe que tem o papel de guardar o rendimento do player durante o

andamento e o final da partida.

**Board:** Classe do tabuleiro do jogo, nela temos todas as peças do jogo, junto de métodos para obter dados sobre a partida, onde está a dificuldade e a criação das peças do jogo.

**GameResult:** Classe responsável por mostrar uma tela onde se definirá o resultado do jogo após terminado, mostrando dados sobre a partida.

**VictoryScreen:** Classe filha de GameResult utilizada para a condição de vitória no jogo, nela temos a pontuação e o bônus por tempo extra.

**DefeatScreen:** Classe filha de GameResult utilizada para a condição de derrota no jogo, com diferentes imagens e efeitos.

**Piece:** Classe responsável por cada peça do jogo, onde será guardado seu conteúdo, fundo e um botão associado a tal peça.

**ButtonHandler:** Classe responsável por comandar as ações dos botões, como virar e formatar.

**ImageCompressor:** Classe responsável por redimensionar as imagens baseado nas dificuldades do jogo.

**Stopwatch:** Classe responsável por criar um cronômetro para definir o tempo das partidas.

## Diagrama de Classe:

