Briefing Detalhado: Jogo da Memória SMB

Este documento detalha as principais funcionalidades e a estrutura do código do "Jogo da Memória SMB", com base nos comentários e explicações fornecidas no documento "explicacao_codigo.pdf". O jogo é uma implementação do clássico jogo da memória, incorporando elementos visuais e sonoros do universo Super Mario Bros.

1. Visão Geral do Jogo

O Jogo da Memória SMB é um jogo de concentração onde o jogador deve encontrar pares de cartas idênticas. Ele apresenta uma interface gráfica com cartas temáticas do Mario, sons para interações e um sistema de pontuação que acompanha tentativas, erros e vitórias.

2. Estrutura e Componentes Essenciais

O código é modularizado, utilizando diversas bibliotecas e organizando as variáveis e funções para gerenciar diferentes aspectos do jogo.

2.1. Bibliotecas Essenciais

O jogo depende de cinco bibliotecas principais para sua funcionalidade, cada uma com um propósito específico:

- **Graficos (g):** "Permite manipular a parte gráfica (janela, imagens, textos)". Essencial para a exibição visual do jogo.
- **Teclado (t):** "Permite detectar o pressionamento de teclas do teclado". Usada para interações como reiniciar o jogo.
- **Util (u):** "Oferece funções utilitárias, como sorteio e tempo". Crucial para o embaralhamento das cartas e o controle de tempo de espera.

- **Mouse (m):** "Permite detectar cliques e a posição do mouse". Fundamental para a interação do jogador com as cartas.
- **Sons (s):** "Permite a reprodução de arquivos de áudio". Adiciona feedback sonoro às ações do jogo.

2.2. Constantes do Jogo

Valores fixos que definem as características visuais e de tempo do jogo:

- **Dimensões da Janela:** LARGURA_JANELA = 900 e ALTURA_JANELA = 750.
- **Dimensões das Cartas:** TAMANHO_CARTA_L = 120 e TAMANHO_CARTA_A = 150.
- **Espaçamento:** MARGEM_CARTA = 10.
- **Tempo de Espera:** TEMPO_ESPERA = 1000 (1 segundo), para cartas que não combinam.
- **Total de Cartas:** NUM_CARTAS = 18.

2.3. Recursos Visuais e Sonoros

O jogo utiliza identificadores inteiros para carregar e gerenciar as imagens e sons:

- **IDs de Imagens:**
- img_card_back: Verso da carta.
- img_flower, img_mushroom, img_star, img_1up, img_coin10, img_coin20: Imagens da frente das cartas, representando diferentes elementos do universo Mario.
- **IDs de Sons:**
- cartas_som: Som de virar uma carta.
- erros_som: Som de erro (quando as tentativas acabam e o tabuleiro reseta).
- vitoria_som: Som de vitória.
- resete_som: Som de reinício do tabuleiro.

2.4. Arrays Paralelos para Cartas

São os dados centrais que representam o estado de cada carta no tabuleiro:

- **id_tipos[NUM_CARTAS]:** Armazena o tipo de cada carta (qual imagem ela representa).
- **viradas[NUM_CARTAS]:** Booleano que indica se a carta está virada para cima (verdadeiro/falso).
- **encontradas[NUM_CARTAS]:** Booleano que indica se o par da carta já foi encontrado (verdadeiro/falso).
- **pos_x[NUM_CARTAS], pos_y[NUM_CARTAS]:** Armazenam as coordenadas X e Y de cada carta na tela.

2.5. Variáveis de Controle do Jogo

Variáveis dinâmicas que gerenciam o fluxo e o estado da partida:

- **tentativas = 3:** Número de chances antes do tabuleiro reiniciar.
- **erros = 0:** Contador de reinícios de tabuleiro por erros.
- **vitorias = 0:** Contador de jogos completos com sucesso.
- **cartas_viradas_indices[2]:** Armazena os índices das 2 cartas viradas no turno atual.
- **num_cartas_viradas = 0:** Contador de cartas atualmente viradas (0, 1 ou 2).
- **tempo_inicio_espera = 0:** Marca o início do temporizador para desvirar cartas.
- **aguardando = falso:** Flag que indica se o jogo está em modo de espera.
- **jogo_ganho = falso:** Flag que indica se o jogador venceu a partida atual.

3. Fluxo Principal do Jogo (funcao inicio())

A função inicio() é o ponto de partida e o coração do jogo, orquestrando todas as operações:

- **Inicialização:** Configura a janela gráfica (g.iniciar_modo_grafico, g.definir_dimensoes_janela, g.definir_titulo_janela, g.definir_cor).
- **Carregamento e Redimensionamento de Recursos:** "Carrega as imagens dos arquivos para as variáveis" e "Redimensiona as imagens para o tamanho definido pelas constantes". Também carrega os sons.

- **Primeiro Embaralhamento:** Chama embaralhar_cartas() para configurar o primeiro tabuleiro.
- **Loop Principal do Jogo:** "Ele se repete enquanto o usuário não pressionar a tecla ESC". Dentro deste loop, o jogo:
- processar_entrada(): Lida com a entrada do teclado (ex: reiniciar após vitória).
- verificar_clique_carta(): Detecta interações do mouse com as cartas.
- processar_espera(): Gerencia o temporizador para desvirar cartas.
- g.limpar(): Limpa a tela.
- desenhar_carta(i): Desenha todas as NUM_CARTAS.
- desenhar_interface(): Desenha a HUD com informações do jogo.
- g.renderizar(): Atualiza a tela para exibir as mudanças.
- u.aguarde(50): Pausa o loop para controlar a taxa de quadros.
- **Encerramento:** g.encerrar_modo_grafico() é chamado quando o loop principal termina.

4. Funções Auxiliares do Jogo

Essas funções implementam a lógica específica de diferentes ações no jogo:

4.1. embaralhar_cartas()

- **Propósito:** Prepara o tabuleiro para uma nova partida.
- **Lógica:**
- Define um array tipos_para_embaralhar com os IDs das imagens. O documento nota uma "inconsistência na distribuição dos tipos de cartas em relação a um jogo da memória tradicional", onde alguns tipos têm 4 cartas e outros 2, ao invés de todos terem 2.
- Usa o algoritmo de Fisher-Yates para "Embaralha o array".
- Inicializa id_tipos, viradas (todas falso), e encontradas (todas falso).
- Calcula as pos_x e pos_y para organizar as cartas em um grid 6x3.

4.2. verificar_clique_carta()

- **Propósito:** Detecta e processa o clique do mouse em uma carta.
- **Lógica:**
- Só age se o "botão esquerdo do mouse foi pressionado, se o jogo não está em espera e se menos de 2 cartas estão viradas".
- Itera sobre todas as cartas para verificar se as coordenadas do mouse estão sobre uma carta que "não esteja virada ou já encontrada".
- Se uma carta válida é clicada: a carta é virada (viradas[i] = verdadeiro), seu índice é armazenado, num_cartas_viradas é incrementado, e s.reproduzir_som(cartas_som,falso) é executado.
- Se num_cartas_viradas == 2, chama verificar_combinacao().

4.3. verificar_combinacao()

- **Propósito:** Compara as duas cartas viradas.
- **Lógica:**
- Compara os id_tipos das duas cartas viradas.
- Se são iguais (par encontrado):
    - encontradas[carta1_idx] = verdadeiro e encontradas[carta2_idx] = verdadeiro.
    - num_cartas_viradas = 0.
    - Chama verificar_vitoria().
- Se são diferentes (par não encontrado):
    - Inicia o temporizador tempo_inicio_espera = u.tempo_decorrido().
    - Ativa a flag aguardando = verdadeiro.

4.4. processar_espera()

- **Propósito:** Gerencia o tempo de espera para desvirar cartas diferentes.
- **Lógica:**
- Ativa quando aguardando é verdadeiro e o TEMPO_ESPERA passou.
- Desvira as duas cartas (viradas = falso).

- Reseta num_cartas_viradas e aguardando.
- **Gerenciamento de Tentativas:** Decrementa tentativas.
- Se tentativas <= 0:
    - Reseta tentativas = 3.
    - Incrementa erros++.
    - Reproduz erros_som e resete_som.
    - Reinicia completamente o tabuleiro: todas as cartas são desviradas e marcadas como não encontradas.

4.5. verificar_vitoria()

- **Propósito:** Verifica se todos os pares foram encontrados.
- **Lógica:**
- Conta quantas cartas estão marcadas como encontradas.
- Se cartas_encontradas_count == NUM_CARTAS:
    - Ativa jogo_ganho = verdadeiro.
    - Incrementa vitorias++.
    - Reproduz vitoria_som.

4.6. processar_entrada()

- **Propósito:** Lida com entrada do usuário após uma vitória.
- **Lógica:**
- Se jogo_ganho é verdadeiro e a tecla ESPAÇO é pressionada:
    - Desativa jogo_ganho.
    - Chama reiniciar_cartas() para um novo jogo.
    - Reseta tentativas = 3.
    - Reproduz resete_som.

4.7. desenhar_interface()

- **Propósito:** Desenha os elementos da interface do usuário (HUD).
- **Lógica:**

- Desenha um painel inferior escuro.
- Exibe os contadores de "Tentativas", "Erros" e "Vitórias".
- Exibe "ÚLTIMA TENTATIVA!" em vermelho se tentativas == 1.
- Se jogo_ganho é verdadeiro, exibe "PARABÉNS! VOCÊ GANHOU!" e "Pressione ESPAÇO para continuar" em verde.

4.8. desenhar_carta(inteiro indice)

- **Propósito:** Desenha uma única carta na tela.
- **Lógica:**
- Se a carta está viradas ou encontradas, desenha a imagem da frente correspondente ao seu id_tipos (usando uma estrutura escolha / switch-case).
- Caso contrário (carta virada para baixo), desenha img_card_back.

4.9. reiniciar_cartas()

- **Propósito:** Reseta o estado das cartas no tabuleiro.
- **Lógica:**
- Define viradas[i] e encontradas[i] para falso para todas as cartas.
- Chama embaralhar_cartas() para redistribuir e ocultar os tipos de cartas.