Guia de Estudo: Jogo da Memória SMB

Visão Geral do Código

Este guia de estudo foi elaborado para revisar o seu entendimento sobre o código do Jogo da Memória SMB. Ele aborda a estrutura do programa, a função de cada seção e as interações entre as diferentes partes do código.

Conteúdo Programático

Estrutura do Projeto:

- Inclusão de Bibliotecas: Graficos, Teclado, Util, Mouse, Sons.
- Constantes do Jogo: LARGURA_JANELA, ALTURA_JANELA, TAMANHO_CARTA_L, TAMANHO_CARTA_A, MARGEM_CARTA, TEMPO_ESPERA, NUM_CARTAS.
- IDs de Imagens: Variáveis para carregar e armazenar imagens das cartas.
- IDs de Sons: Variáveis para carregar e armazenar sons do jogo.
- Arrays Paralelos para Cartas: id_tipos, viradas, encontradas, pos_x, pos_y.
- Variáveis de Controle do Jogo: tentativas, erros, vitorias, cartas_viradas_indices, num_cartas_viradas, tempo_inicio_espera, aguardando, jogo_ganho.

Funções Principais:

- inicio(): Ponto de entrada do programa, inicialização gráfica, carregamento de recursos, loop principal do jogo, encerramento.

Funções Auxiliares:

- embaralhar_cartas(): Define e embaralha os tipos de cartas, inicializa o estado das cartas e calcula suas posições.

- verificar_clique_carta(): Detecta cliques do mouse nas cartas, vira a carta e chama verificar_combinacao() se duas cartas forem viradas.
- verificar_combinacao(): Compara duas cartas viradas, marca pares encontrados ou inicia o temporizador de espera.
- processar_espera(): Gerencia o tempo de espera para desvirar cartas não combinadas e decrementa tentativas, reiniciando o tabuleiro se as tentativas acabarem.
- verificar_vitoria(): Conta cartas encontradas e declara vitória se todos os pares forem formados.
- processar_entrada(): Lida com a entrada do usuário após a vitória (reiniciar jogo).
- desenhar_interface(): Desenha o painel de informações do jogo (tentativas, erros, vitórias, avisos).
- desenhar_carta(inteiro indice): Desenha uma carta específica, mostrando a frente ou o verso conforme o estado.
- reiniciar_cartas(): Reseta o estado do tabuleiro e embaralha as cartas para um novo jogo.

Perguntas para Autoavaliação

Quiz (10 Perguntas de Resposta Curta)

1. Qual a principal função da biblioteca Graficos e como ela é utilizada na função inicio()?
2. Explique o propósito das constantes TAMANHO_CARTA_L e TAMANHO_CARTA_A. Como elas afetam as imagens no jogo?
3. Descreva a finalidade dos arrays paralelos viradas e encontradas. Como eles diferem em seu uso?
4. O que acontece se o jogador virar duas cartas que não combinam? Quais variáveis são afetadas e quais funções são chamadas?
5. Qual o papel da variável TEMPO_ESPERA no jogo? Em qual função ela é principalmente utilizada?

6. Como a função verificar_vitoria() determina se o jogo foi ganho? O que ocorre após a vitória ser detectada?
7. Explique a funcionalidade da função embaralhar_cartas(). Que algoritmo é mencionado para o embaralhamento e qual sua importância?
8. Qual a condição para que a função verificar_clique_carta() processe um clique do mouse em uma carta?
9. O que acontece quando o número de tentativas do jogador chega a zero? Cite as ações tomadas pelo jogo.
10. Descreva como a desenhar_carta(inteiro indice) decide qual imagem exibir para uma dada carta.

Chave de Respostas do Quiz

1. A biblioteca Graficos (g) permite manipular a parte gráfica do jogo, como a janela, imagens e textos. Na função inicio(), ela é usada para inicializar o modo gráfico, definir as dimensões e título da janela, definir a cor de fundo, carregar e redimensionar imagens e, finalmente, renderizar tudo na tela.
2. TAMANHO_CARTA_L define a largura e TAMANHO_CARTA_A define a altura de cada carta no jogo. Elas afetam as imagens porque todas as imagens carregadas para as cartas são redimensionadas para essas dimensões na função inicio(), garantindo um tamanho padrão e consistente para todas as cartas visíveis.
3. O array viradas (booleano) indica se uma carta está atualmente virada para cima (verdadeiro/falso). O array encontradas (booleano) indica se um par de cartas já foi encontrado, significando que elas permanecerão viradas para cima e não poderão mais ser clicadas.
4. Se o jogador virar duas cartas que não combinam, a função verificar_combinacao() é chamada. Ela inicia o tempo_inicio_espera com o tempo atual e ativa a flag aguardando. Posteriormente, a função processar_espera() usará esse tempo para desvirar as cartas após o TEMPO_ESPERA.
5. TEMPO_ESPERA define o período em milissegundos (1 segundo) que duas cartas diferentes permanecem viradas antes de serem desviradas automaticamente. Ela é principalmente utilizada na função processar_espera() para controlar o tempo que o jogo aguarda antes de esconder as cartas não combinadas.

6. A função verificar_vitoria() determina se o jogo foi ganho contando quantas cartas estão marcadas como encontradas. Se essa contagem for igual ao NUM_CARTAS total, significa que todos os pares foram formados. Após a vitória, a flag jogo_ganho é ativada, o contador de vitorias é incrementado e um som de vitória é reproduzido.
7. A função embaralhar_cartas() prepara o tabuleiro do jogo. Ela define os tipos de cartas, embaralha sua ordem usando o algoritmo de Fisher-Yates (garantindo aleatoriedade), inicializa todas as cartas como viradas para baixo e não encontradas, e calcula suas posições X e Y na tela.
8. Para que verificar_clique_carta() processe um clique do mouse, as seguintes condições devem ser verdadeiras: o botão esquerdo do mouse deve ter sido pressionado, o jogo não pode estar em modo de espera (aguardando deve ser falso) e menos de duas cartas devem estar viradas (num_cartas_viradas menor que 2). Além disso, o clique deve estar dentro dos limites de uma carta que não esteja viradas ou encontradas.
9. Quando o número de tentativas chega a zero na função processar_espera(), o tabuleiro é reiniciado. As tentativas são resetadas para 3, o contador de erros é incrementado, um som de erro é reproduzido, todas as cartas são desviradas e marcadas como não encontradas, e um som de reset é tocado.
10. A função desenhar_carta(inteiro indice) decide qual imagem exibir verificando o estado da carta. Se a carta no indice estiver viradas para cima ou já encontradas, ela usa uma estrutura escolha (equivalente a switch-case) para desenhar a imagem correspondente ao id_tipos da carta. Caso contrário (se a carta estiver virada para baixo), ela desenha a imagem do verso da carta (img_card_back).

Sugestões de Perguntas em Formato de Ensaio

- Analise a importância da seção de "Constantes do Jogo". Discuta como a modificação de algumas dessas constantes (como TEMPO_ESPERA ou NUM_CARTAS) afetaria a jogabilidade e a experiência do usuário.
- Discorra sobre o papel do loop principal na função inicio(). Explique como ele orquestra as diferentes funcionalidades do jogo, desde a detecção de entrada até a renderização gráfica, e por que a função u.aguarde(50) é crucial nesse contexto.
- Compare as funções verificar_combinacao() e processar_espera(). Detalhe como elas interagem para gerenciar tanto os acertos quanto os erros do jogador, e qual a importância da flag aguardando nesse fluxo.

- Explique a relevância dos "Arrays Paralelos para Cartas" na organização e gestão do estado de cada carta no jogo. Dê exemplos de como a informação em um índice específico em cada array se correlaciona e é utilizada por diferentes funções.
- O jogo da memória tradicional geralmente possui dois pares de cada carta. O comentário na função embaralhar_cartas() aponta uma inconsistência na distribuição dos tipos de cartas. Discuta o impacto dessa inconsistência na jogabilidade e na dificuldade percebida pelo jogador. Proponha como a distribuição dos tipos poderia ser corrigida para aderir a um modelo mais tradicional ou para variar a dificuldade de forma intencional.

Glossário de Termos Chave

- **Graficos (g):** Biblioteca responsável por toda a manipulação visual do jogo, incluindo janela, imagens, textos e renderização.
- **Teclado (t):** Biblioteca que permite detectar o pressionamento de teclas do teclado para interações do usuário.
- **Util (u):** Biblioteca que oferece funções utilitárias como sorteio de números aleatórios (sorteia) e controle de tempo (tempo_decorrido, aguarde).
- **Mouse (m):** Biblioteca para detectar cliques e a posição do ponteiro do mouse na tela.
- **Sons (s):** Biblioteca que gerencia a reprodução de arquivos de áudio no jogo.
- **Constantes:** Valores fixos que não mudam durante a execução do jogo, usados para configurar dimensões (LARGURA_JANELA, ALTURA_JANELA, TAMANHO_CARTA_L, TAMANHO_CARTA_A, MARGEM_CARTA), tempos (TEMPO_ESPERA) e quantidades (NUM_CARTAS).
- **IDs de Imagens (img_card_back, img_flower, etc.):** Variáveis inteiras que armazenam identificadores para as imagens carregadas, permitindo que sejam desenhadas na tela.
- **IDs de Sons (cartas_som, erros_som, etc.):** Variáveis inteiras que armazenam identificadores para os arquivos de áudio carregados, permitindo sua reprodução.
- **Arrays Paralelos (id_tipos, viradas, encontradas, pos_x, pos_y):** Conjuntos de arrays onde cada índice corresponde à mesma carta, mas armazena um tipo

diferente de informação sobre ela (tipo da imagem, estado de virada, estado de encontrada, posição X e Y).

- **tentativas:** Contador que armazena o número de chances restantes antes que o tabuleiro seja reiniciado devido a erros.
- **erros:** Contador do número total de vezes que o jogador falhou em um tabuleiro e ele foi resetado.
- **vitorias:** Contador do número de vezes que o jogador completou um tabuleiro com sucesso.
- **cartas_viradas_indices:** Array que armazena os índices das duas cartas que foram viradas pelo jogador no turno atual.
- **num_cartas_viradas:** Contador que mantém a contagem de quantas cartas estão viradas no momento (0, 1 ou 2).
- **tempo_inicio_espera:** Variável que registra o momento em que o temporizador de espera é iniciado (quando duas cartas diferentes são viradas).
- **aguardando:** Flag booleana que indica se o jogo está em um estado de espera, aguardando o tempo para desvirar cartas não combinadas.
- **jogo_ganho:** Flag booleana que se torna verdadeira quando o jogador encontra todos os pares e vence a partida.
- **inicio():** A função principal do programa, onde a execução começa e o loop do jogo é gerenciado.
- **embaralhar_cartas():** Função que randomiza a ordem dos tipos de cartas e define suas posições no tabuleiro.
- **verificar_clique_carta():** Função que detecta cliques do mouse dentro dos limites de uma carta e a vira.
- **verificar_combinacao():** Função que compara os tipos de duas cartas viradas para determinar se formam um par.
- **processar_espera():** Função que lida com o tempo que as cartas não combinadas permanecem viradas e gerencia a contagem de tentativas.
- **verificar_vitoria():** Função que checa se todas as cartas foram encontradas e o jogador venceu o jogo.
- **processar_entrada():** Função que trata a entrada do teclado, especificamente após uma vitória, para reiniciar o jogo.

- **desenhar_interface():** Função que desenha informações adicionais na tela, como contadores e mensagens de status.
- **desenhar_carta(indice):** Função que desenha uma única carta na tela, exibindo seu verso ou a imagem correspondente ao seu tipo.
- **reiniciar_cartas():** Função que reseta o estado de todas as cartas no tabuleiro para uma nova partida.
- **Algoritmo Fisher-Yates:** Um algoritmo de embaralhamento utilizado para randomizar a ordem dos elementos em um array de forma eficiente e imparcial.