Centro Universitário Unieuro

Guilherme Braga Rios da Costa

Trabalho de Estrutura de Dados

BRASÍLIA - DF

2023

Guilherme Braga Rios da Costa

JOGO DA MEMÓRIA

Trabalho de Estrutura de dados
com o tema “Jogo da memória”
como requerimento para a
avaliação formativa.

BRASÍLIA - DF

2023

SUMÁRIO

INTRODUÇÃO

Neste trabalho, apresentarei o desenvolvimento de um jogo da memória em linguagem de programação C. O jogo consiste em um tabuleiro com diversas cartas viradas para baixo, onde o jogador deve encontrar as cartas iguais em um determinado número de jogadas.

Serão apresentados os conceitos de programação utilizados no desenvolvimento do jogo, além da descrição detalhada de cada etapa do processo de criação, desde a implementação do tabuleiro até a verificação das cartas iguais e o fim do jogo. Além disso, serão discutidos os desafios encontrados durante o desenvolvimento e as soluções aplicadas para superá-los.

Espero que este trabalho possa contribuir para o aprendizado dos conceitos básicos de programação em C e para a compreensão de como esses conceitos podem ser aplicados na criação de jogos simples, porém divertidos e desafiadores.

DESENVOLVIMENTO

1. Descrição do projeto:

1.1. Objetivo: O jogo da memória é um passatempo que tem como objetivo principal exercitar e estimular a memória e a concentração do jogador. Como um projeto de entretenimento, o objetivo é criar um jogo simples, porém desafiador, que pode ser jogado em um terminal de linha de comando. Um jogo da memória em C pode ser divertido para jogadores de todas as idades, ajudando a melhorar a memória, o pensamento crítico e ,em questão a quem está desenvolvendo o jogo, no aprendizado em conceitos básicos da linguagem de programação C

1.2. Escopo:

- O objetivo do projeto é criar um jogo da memória em C, que possa ser jogado por uma pessoa em um terminal de linha de comando.
- O jogo deve permitir que o jogador escolha entre diferentes conjuntos de cartas, cada um com um tema diferente, como animais, frutas, carros, etc.
- O jogo deve ter um número fixo de cartas, por exemplo, 16 cartas no nível fácil, distribuídas em uma matriz de 4x4.
- O jogo deve permitir que o jogador escolha duas cartas por jogada e verifique se elas são correspondentes.
- O jogo deve contar o número de tentativas ou acertos que o jogador fez para completar o jogo.
- O jogo deve mostrar as cartas temporariamente no começo da partida para o jogador poder memorizar posições.
- O jogo deve mostrar as cartas temporariamente em situações de “Bônus” no jogo.
- O jogo deve permitir que o jogador jogue novamente após terminar uma partida.
- O jogo deve ter uma interface de usuário simples e fácil de usar, que inclua instruções sobre como jogar.
- O projeto não incluirá recursos adicionais, como animações, gráficos

avançados ou pontuações de jogadores.

- O projeto deve ser bem documentado, com comentários claros e instruções de compilação e uso.

2. Tecnologias e ferramentas: A linguagem utilizada nesse projeto foi a linguagem de programação C*,* ela é uma linguagem de programação de propósito geral que permite a criação de sistemas operacionais, drivers de dispositivos, jogos, softwares de sistemas, compiladores e muitos outros aplicativos. As bibliotecas que foram utilizadas no projeto foram:

I. stdio.h

- Para usar funções básicas da linguagem de programação C, como printf (imprime algo na tela).

II. stdlib.h

- Para embaralhar as cartas do jogo usando as funções rand() e srand() contidas nessa biblioteca.

III. locale.h

- Para usar a função “setlocale(LC_ALL,”Portuguese”) dessa biblioteca, que permite escrever com acentuação a e colocar sinais como º ou ª dentro de printfs, putchars ou puts utilizados no código.

IV. time.h

- Para definir o valor de semente do gerador de números aleatórios com base no tempo atual, o que aumenta a aleatoriedade dos números gerados.

V. unistd.h

- Para usar a função sleep() dentro do código, a qual suspende a execução do programa pela quantidade de segundos que a função recebe dentro dos parênteses. Exemplo: sleep(5) suspende o programa por 5 segundos.

3. Arquitetura do projeto:

3.1. Estrutura de arquivos: No código está implementada diversas funções que fazem coisas específicas, as quais são chamadas dentro de outras funções para trabalharem em conjunto. O código contêm as seguintes funções:

1. Void Menu_Jogo(): Responsável por criar o menu do jogo, onde o jogador poderá selecionar entre Jogar, Instruções ou sair do jogo.

2. Void Desenho_Menu(): Função que será utilizada por fins estéticos no menu do jogo, ela foi criada para não ser necessário ficar escrevendo vários printfs dentro da função menu, então a cada vez que se deseja fazer o desenho essa função será chamada.

3. Void Instrucoes(): Como o próprio nome diz, essa função é responsável por dar as instruções de como o jogo funciona.

4. Void Escolha_Dificuldade(): Função que será responsável por perguntar ao jogador qual dificuldade ele deseja jogar: fácil, média ou difícil.

5. Void Asterisco_Revelacoes(): Essa função será responsável por imprimir as cartas escondidas no jogo e, caso o jogador acerte um par de cartas, elas permanecerem viradas até o jogo acabar.

6. Void Cartas_Facil(): Essa função será responsável por inicializar a semente de números aleatórios no nível fácil, para que toda vez que o jogo começar, as cartas estejam em ordens diferentes.

7. Void Cartas_Medio(); Essa função seria responsável por inicializar a semente de números aleatórios do nível médio, porém como não foi possível concluir os níveis de dificuldade do jogo, caso o jogador queira jogar no nível médio ela servirá para imprimir uma mensagem de aviso .

8. Void Cartas_Dificil (); Essa função seria responsável por inicializar a semente de números aleatórios do nível difícil, porém como não foi possível concluir os níveis de dificuldade do jogo, caso o jogador queira jogar no nível

difícil ela servirá para imprimir uma mensagem de aviso.

9. Void Embaralhar(): Nessa função é onde inicializa o gerador de números aleátorios e embaralha as cartas (para que a cada vez que o jogador iniciar um novo jogo as cartas estejam em posições diferentes).

10. Void Imprime_Cartas_Facil(): Essa função é responsável por imprimir as cartas do jogo em suas posições, ela será usada para o jogador memorizar as cartas no início do jogo e também quando ele adquirir alguns bônus no jogo.

11. Void Jogar(): Essa função é onde está a maioria das outras funções, é onde acontece a maior parte do jogo.

12. Int Main (): Função principal, que deve estar em todos os códigos para que funcione. Essa função está começando o jogo chamando a função Menu_Jogo() e também está com a função setlocale(LC_ALL, "Portuguese") para que possa haver a utilização de acentos no decorrer do código.

3.2. Estrutura e vetores: Foram usadas no jogo duas estruturas, uma para armazenar a variável Cartas_Corretas do tipo J, que será utilizada mais a frente no jogo para virar o par de cartas que foi achado. A estrutura já é utilizada logo em seguida para definir uma variável global acertadas. A outra estrutura é usada para armazenar as coordenadas x e y que o jogador irá digitar para virar as cartas 1 e 2 respectivamente. Logo em seguida já são criadas variáveis globais do tipo Coordenadas (2ª struct).

Struct 1:

```
typedef struct sJogo_memoria {
  int Cartas_corretas_F[4][4];
}J;

int Cartas_F [4][4] = {{14,21,15,16},{15,20,19,14},{21,16,17,18},{18,19,20,17}};
J acertadas;
```

Struct 2:

```
typedef struct sJogo {
    int x;
    int y;
}Coordenadas; /* Struct que contêm as variáveis das coordenadas que serão usadas no código */

Coordenadas primeira;
Coordenadas segunda;
```

4. Funcionalidades:

4.1. Lista de funcionalidades:

- Seleção do Menu (entrada: opção selecionada; saída: jogo, instruções ou sair do jogo).
- Seleção de nível de dificuldade (entrada: nível de dificuldade selecionado; saída: lista de cartas embaralhadas).
- Tempo de demonstração das cartas (entrada: demonstração das cartas; saída: cartas escondidas).
- Escolha de cartas (entrada: posição da carta selecionada; saída: verificação de correspondência ou não).
- Finalização do jogo (entrada: término do jogo; saída: vitória ou derrota).

4.2. Implementação:

- Seleção do Menu:
  - Criação de um menu com as opções jogar, instruções e sair.
  - Recebimento da escolha do usuário.
  - Caso seja jogar, chama a função Escolha dificuldade.
  - Caso seja instruções, demonstra as instruções do jogo.
  - Caso seja sair, sai do jogo.

- Seleção de nível de dificuldade:
  - Criação de um menu com as opções de nível de dificuldade.
  - Recebimento da escolha do usuário e geração de um conjunto

correspondente de cartas.

- Aviso ao jogador.
- Definição das cartas no respectivo nível.

- Tempo de demonstração das cartas:
  - Demonstra as cartas por 10 segundos para o jogador memorizar.
  - Esconde as cartas.
  - Ao decorrer do jogo existem condições “Bônus”, em que haverá um tempo de demonstração das cartas.

- Escolha de cartas:
  - Recepção das posições das cartas selecionadas pelo usuário;
  - Verificação de correspondência das cartas, se houver.
  - Exibição do par de cartas encontrado, se houver.
  - Exibição da matriz do jogo, com as cartas encontradas e cartas escondidas.
  - Atualização do número de tentativas e acertos do jogador.

- Finalização do jogo:
  - Verificação de correspondência de todas as cartas.
  - Verificação de vitória ou derrota.
  - Exibição do resultado e opção de jogar novamente ou sair.

CONCLUSÃO

Diante do exposto, o jogo da memória nesse projeto teve uma grande relevância para um maior aprendizado da linguagem de programação C e da lógica de programação. Durante a implementação do código foram encontradas diversas dificuldades, as quais com tempo e esforço foram possíveis de contornar conseguir chegar a um resultado esperado. O resultado obtido no trabalho foi satisfatório, porém pode receber ainda muitas melhorias, como: Completar no jogo os níveis de dificuldades média e difícil, virar a carta 1 e a carta 2 na própria matriz quando o usuário escolhe-las ao invés de mostra-las em baixo da matriz, fazer mais estruturas condicionais caso o jogador entre com um valor indesejado ou incorreto no lugar da carta.

REFERÊNCIAS BIBLIOGRÁFICAS

ARAUJO, Jário. dominando a linguagem c. Rio de Janeiro: Ciência Moderna Ltda., 2004.

CELES, W.; CERQUEIRA, R.; RANGEL, J. L. introdução a estruturas de dados com técnicas de programação em C. 2. ed. Rio de Janeiro: Elsevier Editora Ltda., [2007].

DAMAS, L. linguagem c. 10. ed. Rio de Janeiro: LTC, [2015].

Intellenctuale Tecnologia & treinamento. Valores aleatórios em C com a função rand. Disponível em: http://linguagemc.com.br/valores-aleatorios-em-c-com-a-funcao-rand/. Acesso em 01 maio. 2023.

Oliveira, D. (04 de Maio de 2021). *1 Vídeo (6:42 min). JOGO DA MEMÓRIA EM LINGUAGEM C*. Fonte: Publicado pelo canal Daniel Oliveira: https://youtu.be/dolFYMwX_po

Stackoverflow. Passar matriz como parâmetro de função. Disponível em: https://pt.stackoverflow.com/questions/60065/passar-matriz-como-par%C3%A2metro-de-fun%C3%A7%C3%A3o. Acesso em: 03 maio. 2023.