

Documentação de Trabalho Bimestral

Projeto: Jogo Super Trunfo.

Linguagem e metodologia: C++, Programação Orientada a Objetos.

Curso e período: Ciência da Computação, 4º Semestre.

Responsáveis: Fernanda Bonfim Santos, Bárbara Diogo.

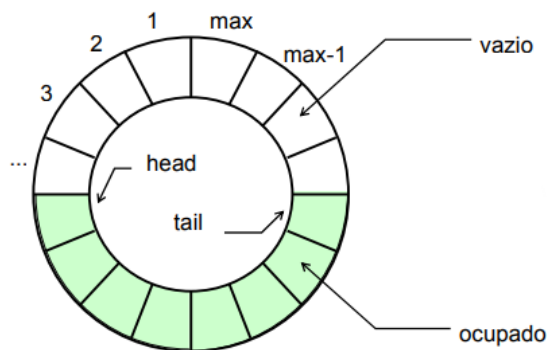
1. Introdução:

O objetivo deste trabalho é implementar um Tipo Abstrato de Dados (TAD) para a resolução de um jogo de cartas chamado “Super Trunfo”. Deverão ser implementados métodos para leitura de um arquivo .csv, embaralhamento, divisão, retirada e comparação das cartas. Com isso praticamos os conceitos básicos de estrutura de dados, visto até agora.

2. Implementação:

Estrutura de Dados

Na implementação do código foi criado um TAD Vetor circular.



Onde o campo max determina o tamanho máximo do vetor, índice head e tail indicam o início e o fim.

Métodos e Procedimentos

void Read(carro lista[]): recebe um vetor lista do tipo carro (struct com os registros das castas), abre o arquivo cards.csv e faz a leitura do mesmo. Foi necessário fazer a conversão dos registros do tipo int para string.

void Rear(carro &x): recebe por referência uma variável x do tipo carro para retornar a última carta.

void Embaralhar(carro definicao[], superTrunfo &J1, superTrunfo &J2): para embaralhar as cartas e dividir em dois decks foi necessário 3 parâmetros, um vetor do tipo carro para armazenar as cartas embaralhadas e para isso geramos números aleatórios com a função rand, e duas variáveis por referência para realizar a inserção e divisão igual das cartas nos decks.

void Serve(carro &x): operação utilizada para remoção das cartas, é preciso realizar o decremento da variável count (traz o número de elementos dentro do vetor circular), a remoção da fila sempre é feita no head, por isso a carta no índice head passa a ser a variável x a ser removida, por fim é feita o rearranjo da posição do novo head após a remoção.

void CartasDosJogadores(carro x): mostra ao usuário os registros contidos na carta.

void JogadasDoJogo(superTrunfo &J1, superTrunfo &J2, carro &a, carro &b, bool &jogadas): dentro da função teremos a chamada de outra, que é ComparacaoDeCartas(), se a carta do jogador 1 for a maior

void RodadasDoJogoTrunfo(superTrunfo &J1, superTrunfo &J2, carro &a, carro &b, bool &jogadas): nesta função é feita uma análise da carta da vez, caso seja trunfo ou não. Se a carta do jogador 1 for a carta trunfo ele vence a rodada e fica com a carta do jogador 2 ou caso a carta do jogador 2 for a carta trunfo ele vence a rodada e fica com a carta do jogador 1.

void Jogo(superTrunfo &J1, superTrunfo &J2): mostra ao usuário as cartas de cada um dos jogadores e a quantidade delas que cada um deles possui, isso enquanto o número de carta de ambos for menor que o limite máximo de cartas que são 32. Caso o número de cartas do jogador 1 for igual ao valor máximo de cartas do baralho significa que ele é o vencedor da rodada e vice versa com o jogador 2.

void Append(carro x): a função Append insere as cartas no vetor fila, recebe uma variável por valor, depois de verificar se a fila não está cheia é feito um acréscimo no número de elementos no vetor e a carta será inserida sempre no final da fila.

bool ComparacaoDeCartas(superTrunfo &J1, superTrunfo &J2, carro &a, carro &b, int escolha): na comparação das cartas precisaremos das cartas dos jogadores 1 e 2 e uma variável escolha para guardar qual registro o jogador irá querer comparar (peso do veículo, preço da carta, tamanho do motor ou cavalos de potência). Essa função será usada dentro da JogadasDoJogo().

bool Empty(): se o número de elementos for zero então a fila está vazia (sem cartas).

bool Full(): a fila só está cheia quando o número de elementos no vetor for igual ao limite máximo estabelecido.

bool ETrunfo(carro a, carro b): como foi determinado pelo enunciado do trabalho existe uma carta trunfo (1A), caso ela esteja com algum dos jogadores pode ser usada para vencer qualquer carta do baralho, exceto as do mesmo grupo A.

int Size(): retorna nosso contador count com o número de elementos contidos na fila.

Programa Principal

O programa principal traz o vetor com o tamanho limite de cartas, lê e embaralha todo o baralho e por fim chama o método Jogo(J1, J2), pois nele todas as outras funções estão contidas e serão chamadas dentro dele.

Organização do Código

O trabalho foi dividido em 3 arquivos principais: superTrunfo.cpp e superTrunfo.h fazem a implementação do TAD e no arquivo main.cpp implementa o programa principal.

Para simplificar a execução do trabalho decidimos dividi-lo em mais operações além das necessárias para gerar a estrutura da fila circular. Como a operação para embaralhar, fazer as jogadas dos jogadores, rodada trunfo, comparação das cartas, mostrar a carta na tela para o usuário e a operação Jogo() que traz todas as outras funções dentro dele, facilitando sua chamada no programa principal.

3. Testes

Após compilar o programa realizamos alguns testes para verificar a veracidade da execução do projeto. As figuras abaixo mostram a saída de uma das execuções realizadas.

```
PS C:\Users\barba\OneDrive\Área de Trabalho\Quarto Período\Estrutura de Dados\superTrunfo> ./main.exe
```

```
*****
| Bem-Vindo ao SuperTrunfo da Fernanda e da Barbara! |
*****

-----
| Chamaremos aqui, o usuário atrá da máquina de: Jogador Person |
| Chamaremos aqui, a máquina de: Jogador Machine |
-----

-----
| Grupo da carta: 3C |
| Modelo da carta: porsche3 |
| 1 - Peso do veículo: 2756 |
| 2 - Preço da carta: 32528 |
| Escolha um valor de 1 a 4 |
-----

-----
| Jogador Machine jogou sua carta! |
| Jogador Person, digite algum numero para continuar o jogo! |
-----

3
*****
| Vencedor da rodada !* o Jogador Machine! |
*****

-----
| Cartas do Jogador Person: 16 |
-----

-----
| Cartas do Jogador Machine: 16 |
-----
```

```
-----
| Grupo da carta: 8D |
| Modelo da carta: nissan2 |
| 1 - Peso do veículo: 3139 |
| 2 - Preço da carta: 18399 |
| 3 - Tamanho do Motor: 181 |
| 4 - Cavalos de potencia: 160 |
-----

-----
| Escolha um valor de 1 a 4 |
-----

-----
| Trunfo!! |
-----

-----
| Carta do Jogador Person: 8D |
| Carta do outro Jogador Machine: 1A |
-----

*****
| Carta TRUNFO do Jogador Machine !* a Campeão!! |
*****

-----
| Cartas do Jogador Person: 15 |
-----

-----
| Cartas do Jogador Machine: 17 |
-----

-----
| Grupo da carta: 2C |
| Modelo da carta: jaguar2 |
| 1 - Peso do veículo: 4867 |
| 2 - Preço da carta: 35559 |
| 3 - Tamanho do Motor: 259 |
| 4 - Cavalos de potencia: 177 |
-----
```

```
-----|
| Escolha um valor de 1 a 4 |
|-----|

|-----|
| Jogador Machine jogou sua carta |
| Jogador Person, digite algum numero para continuar o jogo! |
|-----|

2
|-----|
| Vencedor da rodada !! o Jogador Person! |
|-----|

|-----|
| Cartas do Jogador Person: 16 |
|-----|

|-----|
| Cartas do Jogador Machine: 16 |
|-----|

|-----|
| Grupo da carta: 4A |
| Modelo da carta: mercedes-benz5 |
| 1 - Peso do veiculo: 3770 |
| 2 - Preço da carta: 31600 |
| 3 - Tamanho do Motor: 183 |
| 4 - Cavalos de potencia: 128 |
|-----|

|-----|
| Escolha um valor de 1 a 4 |
|-----|

|-----|
| Jogador Person e sua VEZ! |
|-----|
```

4. Conclusão

Os resultados ficaram dentro do esperado e o prazo para realização do projeto foi suficiente para sua finalização. Uma das dificuldades encontradas foi implementar na linguagem C++ as operações extras, foi um raciocínio trabalhoso de se pôr em prática, porém a implementação transcorreu sem maiores problemas.

5. Referências

PELLISON, Felipe. Filas. 06 dez. 2021. Apresentação do Power Point. Disponível em:
https://portal.baraodemaui.br/pluginfile.php/1263353/mod_resource/content/0/3.0-Filas.pdf. Acesso em 27 ago. 2022.

6. Anexos

Listagem dos programas:

- superTrunfo.cpp
- main.cpp
- superTrunfo.h
- cards.csv