

Universidade de São Paulo

SSC0600 - Introdução a ciência da computação I

Trabalho 4 : Rummikub

julho/2018

# Índice

Introdução.....	
Descrição do Projeto.....	
Tutorial.....	
Outras informações.....	
Referências.....	

# 1 - Introdução

## Participantes:

P=0 : Victoria Helena Ianni 10255900

P=1 : Antonio Sebastian Fernandes Rabelo 10797781

P=2 : Renata Oliveira Brito 10373663

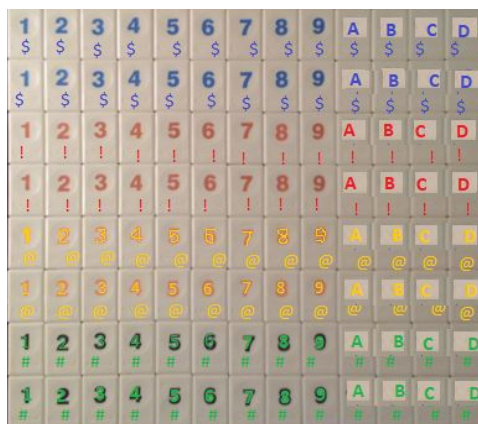
P=3 : Helbert Moreira Pinto 10716504

P=4 : Gustavo Machado Mendes 1082115

P=5 : Nicolas Ribeiro Batistuti 10408351

O jogo Rummikub foi inventado por Ephraim Hertzano no início dos anos de 1930, sendo popularizado em 1977. O inventor propôs três diferentes regras para o jogo sendo adotada a regra Sabra para esse trabalho.

O baralho possui 106 peças, sendo 2 coringas (marcados por ‘\*\*’) e as outras peças divididas em quatro grupos de símbolos ('!', '@', '#', '\$'), também representados pelas cores (vermelho, amarelo, verde, azul), contendo cada cor duas sequências de peças enumeradas em hexadecimal (1..9, A..D).



O número de jogadores participantes varia de um a cinco e quatro sendo que no início cada jogador recebe 14 peças e o objetivo é combinar as peças entre si ou com as peças que já estão na mesa, formando combinações, o vencedor será aquele que primeiro conseguir colocar todas as suas peças sobre a mesa, em combinações aceitáveis.

Sendo essas combinações, entre grupos (conjuntos de 3 ou 4 peças com o mesmo número e necessariamente com cores diferentes) ou entre sequências (conjuntos de 3 a 13 peças de mesma cor, com números em sequência). O coringa pode assumir representar qualquer peça de uma combinação (em que ela está inserida).

Dessa forma, não é aceitável adicionar dois coringas à uma combinação compostas por três peças de mesmo valor numérico e de diferentes símbolos, pois não os coringas podem assumir valores que satisfaçam a combinação.

A primeira jogada de cada jogador deve colocar sobre a mesa pelo menos 30 pontos (somatório dos valores numéricos representados pelas peças, o coringa assume o lugar da peça que está substituindo).

Em cada rodada que um jogador não puder efetuar nenhuma jogada, uma peça será adicionada à sua mão.

Ao final de sua jogada, todas as peças sobre a mesa devem fazer parte de combinações válidas.

### **Legenda :**

mesa : peças que estão na mesa disponíveis para manipular.

temp : mão onde se manipula as combinações.

mão : peças disponíveis para manipular.

mão -> temp : enviar peças da mão para temp.

mesa -> temp : enviar as peças da mesa para manipular na temp.

Exemplos de manipulações possíveis feitas na temp:

**Cortar uma sequência (editar temp):** o jogador pode retirar a peça inicial ou final de uma sequência para usá-la em outra combinação, desde que a sequência permaneça com pelo menos três peças.

**Cortar uma quadra (editar temp):** o jogador pode retirar uma das peças de uma quadra (que se torna, então, uma trinca) para usá-la em outra combinação. O mesmo não pode ser feito em uma trinca, já que uma combinação de apenas duas peças não seria válida.

**Deslocar uma sequência de três (juntar temp + editar temp):** ao se colocar uma peça na ponta (início ou final) de uma sequência de três peças, esta passa a ter quatro peças, e portanto a outra ponta pode ser retirada para uso em outra combinação.

**Substituir em uma trinca (juntar temp + editar temp):** em uma trinca sobre a mesa, o jogador pode adicionar a peça de mesmo número e da cor que falta, transformando-a uma quadra, e portanto podendo retirar uma das outras peças da trinca para usá-la em outra combinação.

**Dividir uma sequência (juntar temp + editar temp):** um jogador pode dividir uma sequência longa e colocar as peças correspondentes no meio, desde que as sequências resultantes tenham, cada uma, um mínimo de três peças. Por

exemplo, se há na mesa uma sequência azul de 6 a 10 e o jogador tem na mão um 8 azul, ele pode colocar a sua peça no meio e formar as sequências 6-7-8 e 8-9-10.

**Substituir um coringa (juntar temp + editar temp):** se o jogador possui a peça que substitui um coringa em uma combinação sobre a mesa, ele pode trocá-la, podendo a seguir usar o coringa em qualquer outra combinação.

## 2 - Descrição do projeto

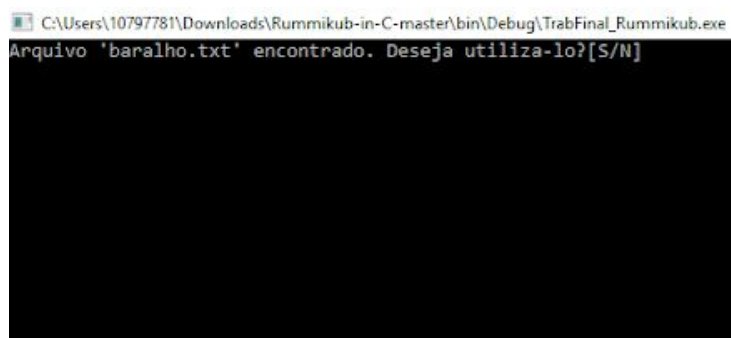
Para desenvolver o projeto, utilizamos a IDE de ambiente de desenvolvimento CodeBlocks (versão 17.12 - x86x64) e para compilar o fonte utilizamos o GCC. O sistema operacional utilizado foi o Windows 10 (versão 64bits). O código fonte está no arquivo main.c, é necessário que esteja junto ao código fonte do projeto o arquivo "ConioC/conio.c", pois ele é utilizado para personalizar o programa. O arquivo compilado é o main.o. As bibliotecas utilizadas foram:

- `stdlib.h`
- `stdio.h`
- `time.h`
- `stdbool.h`

## 3 - Tutorial

Inicialmente é necessário setar o intervalo de tempo em segundos entre as jogadas de cada jogador, para que um não veja as peças do outro, por meio do: “`#define TIME_ENTRE_JOG`”, que inicialmente está definida para 0 segundos.

Se existir o arquivo `baralho.txt` no diretório do código fonte do programa, é exibida uma mensagem perguntando ao usuário se deseja utilizar o baralho advindo do arquivo ou se é necessário gerar um baralho dinâmico. (caso não encontre o arquivo, a mensagem não será exibida)



Iniciando o programa: digite a quantidade número de jogadores

```
C:\Users\10255900\Downloads\Rummikub-in-C-master\bin\Debug\TrabFinal_Rummikub.exe
Quantidade de jogadores [1-5]: 2
```

Informe o nome dos jogadores :

```
C:\Users\10255900\Downloads\Rummikub-in-C-master\bin\Debug\TrabFinal_Rummikub.exe
Quantidade de jogadores [1-5]: 2
Nome do jogador 1 [MAX 20 carac]: jogador 1
Nome do jogador 2 [MAX 20 carac]: jogador 2
```

Aparecerá a mão do jogador da vez:

```
C:\Users\10255900\Downloads\Rummikub-in-C-master\bin\Debug\TrabFinal_Rummikub.exe
Vez de jogador 1
Baralho contem 78 cartas
MESA:
(Vazio)
MAO:
2! 2! 9! 1@ A@ C@ 3# 6# 3$ 4$ 5$ 7$ 8$ 9$
a b c d e f g h i j k l m n

*****|
Opcoes |
*****|
Mao -> Temp      | 4 |
Encerar jogada   | 5 |
*****|
Selecione
```

Para colocar as peças na mesa, deve-se colocar as peças na mão temporária e depois colocar na mesa: no exemplo o jogador 1 colocar as peças ijk (em letras minúsculas, como a sua representação, e sem espaço).

```
C:\Users\10255900\Downloads\Rummikub-in-C-master\bin\Debug\TrabFinal_Rummikub.exe
Vez de jogador 1
Baralho contem 78 cartas
MESA:
(Vazio)
MAO:
2! 2! 9! 1@ A@ C@ 3# 6# 3$ 4$ 5$ 7$ 8$ 9$
a b c d e f g h i j k l m n

*****
| Opcoes
| *****
| Mao -> Temp          | 4
| Encerrar jogada     | 5
| *****
| Selecione 4
| Selecione os indices das pecas:
```

```
C:\Users\10255900\Downloads\Rummikub-in-C-master\bin\Debug\TrabFinal_Rummikub.exe
Vez de jogador 1
Baralho contem 78 cartas
MESA:
(Vazio)
MAO:
2! 2! 9! 1@ A@ C@ 3# 6# 3$ 4$ 5$ 7$ 8$ 9$
a b c d e f g h i j k l m n

*****
| Opcoes
| *****
| Mao -> Temp          | 4
| Encerrar jogada     | 5
| *****
| Selecione 4
| Selecione os indices das pecas: ijk
```

Depois de selecionadas, as peças aparecerão na mesa e você deve encerrar a jogada (5) :

```
C:\Users\10255900\Downloads\Rummikub-in-C-master\bin\Debug\TrabFinal_Rummikub.exe
Vez de jogador 1

Baralho contem 78 cartas

MESA:
(Vazio)
-----
TEMP:
T[a] -> 3$ 4$ 5$

MAO:
2! 2! 9! 1@ A@ C@ 3# 6# 7$ 8$ 9$
a b c d e f g h i j k

*****
Opcoes
*****
Editar Temp      | 2 |
Juntar Temp     | 3 |
Mao -> Temp      | 4 |
Encerrar jogada  | 5 |
*****
Selecione _
```

Para uma nova combinação, digite 4 (para enviar da mão para a temp):

```
C:\Users\10255900\Downloads\Rummikub-in-C-master\bin\Debug\TrabFinal_Rummikub.exe
Vez de jogador 1

Baralho contem 78 cartas

MESA:
(Vazio)
-----
TEMP:
T[a] -> 3$ 4$ 5$

MAO:
2! 2! 9! 1@ A@ C@ 3# 6# 7$ 8$ 9$
a b c d e f g h i j k

*****
Opcoes
*****
Editar Temp      | 2 |
Juntar Temp     | 3 |
Mao -> Temp      | 4 |
Encerrar jogada  | 5 |
*****
Selecione Opcao invalida
Selecione 4
Selecione os indices das pecas: ijk
```



```
C:\Users\10255900\Downloads\Rummikub-in-C-master\bin\Debug\TrabFinal_Rummikub.exe
Vez de jogador 1

Baralho contem 78 cartas

MESA:
(Vazio)
-----
TEMP:
T[a] -> 3$ 4$ 5$
T[b] -> 7$ 8$ 9$

MAO:
2! 2! 9! 1@ A@ C@ 3# 6#
a b c d e f g h

*****
| Opcoes                                     |
|*****|
| Editar Temp                             | 2 |
| Juntar Temp                             | 3 |
| Mao -> Temp                             | 4 |
| Encerrar jogada                         | 5 |
|*****|
| Selecione _                             |
|*****|
```

Para encerrar a jogada aperte 5 e aparecerá, depois do intervalo de tempo definido, a mão do próximo jogador (imagina-se que um jogador não queira ver as cartas dos adversários:

```
C:\Users\10255900\Downloads\Rummikub-in-C-master\bin\Debug\TrabFinal_Rummikub.exe
Vez de jogador 2

Baralho contem 78 cartas

MESA:
M[a] -> 3$ 4$ 5$
M[b] -> 7$ 8$ 9$

MAO:
4! D! 9@ A@ B@ D@ A# C# C# D# 1$ 2$ 8$ D$
a b c d e f g h i j k l m n

*****
| Opcoes                                     |
|*****|
| Mao -> Temp                             | 4 |
| Encerrar jogada                         | 5 |
|*****|
| Selecione _                             |
|*****|
```

Devido ao fato da primeira jogada ser necessário somar 30 pontos, o jogador só pode manipular as combinações da mesa após a primeira jogada, dessa forma, o menu do jogador que ainda não fez a primeira jogada é restrito como na imagem seguinte.

```

C:\Users\10255900\Downloads\Rummikub-in-C-master\bin\Debug\TrabFinal_Rummikub.exe
Vez de jogador 2
Baralho contem 78 cartas

MESA:
M[a] -> 3$ 4$ 5$
M[b] -> 7$ 8$ 9$

MAO:
4! D! 9@ A@ B@ D@ A# C# C# D# 1$ 2$ 8$ D$
a b c d e f g h i j k l m n

*****
Opcoes
*****
Mao -> Temp          | 4
Encerrar jogada     | 5
*****
Selecione 4
Selecione os indices das pecas: bfjn

```

```

C:\Users\10255900\Downloads\Rummikub-in-C-master\bin\Debug\TrabFinal_Rummikub.exe
Vez de jogador 2
Baralho contem 78 cartas

MESA:
M[a] -> 3$ 4$ 5$
M[b] -> 7$ 8$ 9$
-----
TEMP:
T[a] -> D! D@ D# D$

MAO:
4! 9@ A@ B@ A# C# C# 1$ 2$ 8$
a b c d e f g h i j

*****
Opcoes
*****
Editar Temp          | 2
Juntar Temp          | 3
Mao -> Temp          | 4
Encerrar jogada     | 5
*****
Selecione _

```

## 4. Outras informações

Há no programa restrições:

- O jogador deve manipular as combinações corretamente, de forma que não é possível retornar para a mesa uma jogada inválida
- Não é possível puxar mais de uma combinação por vez da mesa para a temp.
- É necessário obedecer estritamente o carácter representante de cada peça.



