

# Universidade de São Paulo

## Relatório do Trabalho 4

Professor: Adenilso da Silva Simão

Feito por:

Maurício Gabriel Garcia Catellan; N° USP: 10716592

# Introdução

Este trabalho se trata de um programa, implementado em C, onde se encontra a lógica e a sequência de instruções para que se realize o jogo Rummikub.

Rummikub é um jogo que pode ser jogado com peças ou cartas; no caso, vetores de caracteres, que contêm uma numeração hexadecimal e um símbolo, cumprem essa função, e eles são chamados de peças pelo programa. Já as interações são feitas por entradas através de um console.

O trabalho foi feito, individualmente, por Maurício Gabriel Garcia Catellan (Nº USP: 10716592).

## Descrição do projeto

O programa do projeto foi compilado pelo GNU GCC Compiler, em Windows x64. Os códigos-fonte (3 arquivos .c) e um cabeçalho com declaração de registro e de funções (1 arquivo .h) estão em anexo, bem como o executável gerado após a compilação. Já as bibliotecas padrão usadas foram a “stdio.h”, a “stdlib.h” e a “string.h”.

## Tutorial

Como o programa foi compilado originalmente pelo GNU GCC Compiler, esse é o compilador ideal para compilá-lo. Já a sua execução pode ser feita através do executável gerado após a compilação.

Ao executá-lo, aparecerá o seguinte menu:

```
>>>RUMMIKUB<<<
```

- 1- Jogar - modo aleatorio
- 2- Jogar - modo controlado
- 3- Sair

Entre com 1 ou 2 para jogar. O que difere essas opções é como estará o baralho durante o jogo: na opção 1, o deck é

embaralhado (então as peças que aparecem são aleatórias); já na opção 2, o deck não é embaralhado e a ordem de suas peças é definida por um arquivo (arquivo “baralho.txt”, definido previamente).

Após escolher 1 ou 2, o programa perguntará em quantos jogadores será o jogo. Entre com o número desejado, sabendo que o mínimo é 1, e o máximo, 5.

Tendo escolhido, aparecerá a seguinte mensagem:

“As pecas estao na mesa! Escolham uma peca para que seja decidido o primeiro a jogar (peca de 0 a 105)”.

Essa parte da execução simula o momento em que cada jogador escolhe uma peça (com todas viradas para baixo) para que, então, seja decidido quem joga primeiro. Faça um número de entradas igual ao número de jogadores, e então, serão reveladas as peças escolhidas e quem é o primeiro a jogar (baseado nos valores das peças escolhidas, sendo que o coringa é a peça de maior valor). Exemplo, com 3 jogadores:

“0

1

2

Pecas escolhidas:

Jogador 1: 7#

Jogador 2: B#

Jogador 3: 7@

Jogador 2 joga primeiro!”

No exemplo, as primeiras 3 linhas foram as entradas. A seguir, apareceram as peças escolhidas, e, como esperado, o jogador 2 foi escolhido como o primeiro a jogar (B, que em decimal é 11, é maior que 7).

Após essa parte, inicia-se o jogo de fato, com os jogadores obtendo as mãos iniciais. A mensagem que comprova essa obtenção é:

“Jogadores preencheram as maos! Inicio das jogadas!”

Após ela, inicia-se o primeiro turno do primeiro jogador a jogar. Nesse momento, observa-se o menu de jogadas que será

visto ao longo de todo o jogo. Ele representa as possíveis jogadas que um jogador pode realizar, e aparece da seguinte forma:

“Veza do jogador 2

O que deseja fazer?

- 1- Usar pecas de sua mao na mesa
- 2- Manipular as pecas da mesa
- 3- Comprar uma peca
- 4- Encerrar a vez

Mao: 5\$ 9\$ 1\$ 6! 9# 8@ D! C\$ 3\$ B\$ 4! B@ A\$ 7@”

Sendo que o jogador da vez e a mão são para a continuação do exemplo acima. Note que sua mão foi mostrada na tela. Ela aparecerá a cada repetição do menu. Com o menu sendo mostrado, siga escolhendo uma das opções de jogada.

## Tutorial de uso de peças da mão na mesa

Caso a opção 1 seja escolhida, aparecerá a seguinte mensagem:

“Deseja jogar uma nova combinacao (1) ou apenas uma peca (2)?”

Entre com 1 ou com 2 para prosseguir. No caso da primeira jogada do jogo, não haverá peças na mesa, então apenas a opção 1 será possível de ser executada.

Escolhendo a opção 1, aparecerá:

“Diga que pecas deseja jogar (as pecas estao, da esquerda para a direita, em ordem crescente, e a primeira peca e a 0). Entre com -1 para terminar as escolhas”

Então, escolha as peças conforme explicado na mensagem, e finalize entrando com -1. Após a escolha, o programa falará se ela foi válida ou não. Se foi, as peças serão colocadas na mesa e a mão diminuirá de tamanho, e, a partir daí, a mesa começará a ser mostrada na tela, nos momentos de mostra do menu. Se não, nada ocorrerá. Exemplo, feito após os exemplos acima e após a compra de uma peça “C!”, uma peça “4#” e uma peça “C#”, nessa ordem:

“0

2

10

-1

Pecas escolhidas:

C# C! C\$

Grupo posto na mesa!”

E, após essa execução, abaixo, do menu, aparece:

“Mesa: C# C! C\$”

Agora, escolhendo a opção 2, aparece a seguinte mensagem:

“Que peca deseja jogar? (as pecas estao, da esquerda para a direita, em ordem crescente, e a primeira peca e a 0)”

Selecione a peça de forma similar a o que é feito na opção 1, e, então, aparece a seguinte mensagem:

“A que posicao da mesa deseja leva-la? (as pecas serao deslocadas para a direita)”

Selecione uma peça da mesa da mesma forma, e, se combinações válidas forem formadas, aparece a seguinte mensagem:

“Peca colocada na mesa!”

Tomando como exemplo uma mesa no seguinte estado:

“Mesa: 9@ 8@ 7@”

E escolhendo uma peça “6@”, a se colocar na posição 0 da mesa, as mensagens acima aparecem, e a mesa fica da seguinte forma:

“Mesa: 6@ 9@ 8@ 7@”

E a mão fica sem a peça utilizada, o que comprova o funcionamento da função.

## Tutorial de manipulação de peças da mesa

Ao selecionar a opção 2, aparecerá a seguinte mensagem:

“Que peca deseja manipular? (as pecas estao, da esquerda para a direita, em ordem crescente, e a primeira peca e a 0)”

Selecione uma peça da mesa seguindo a regra estabelecida no parênteses e, então, aparecerá a mensagem:

“A que posicao da mesa deseja leva-la? (as pecas serao deslocadas para a direita”

Entre com a posição seguindo a mesma regra usada acima, e, então, se combinações válidas foram formadas, as peças trocarão de lugar.

Tomando como exemplo uma situação em que a mesa está da seguinte forma:

“Mesa: 7! 6! \*\* 5! 5# 5\$”

Nessa situação, há duas combinações (“7! 6! \*\* 5!” e “5! 5# 5\$”). Trocando o coringa com a peça “5!”, através do uso das operações acima, a mesa fica da seguinte forma:

“7! 6! 5! 5# 5\$ \*\*”

O que comprova o funcionamento da função (já que tanto “7! 6! 5!” quanto “5# 5\$ \*\*” são combinações válidas).

## Tutorial de compra de peça

Ao selecionar 3, uma nova peça aparecerá na sua mão (à direita). Ela será considerada a nova posição 0 no uso das outras funções. Caso esteja jogando no modo 1, ela será aleatória; já no modo 2, poder-se-á saber qual será ela ao observar o arquivo lido.

Após terminar de executar as jogadas desejadas, entre com “4” para passar o turno para o próximo jogador. Nas primeiras jogadas de cada jogador, as peças jogadas devem somar 30 pontos, então, caso essa quantia não tenha sido atingida, não será possível passar o turno. Já nas jogadas restantes, poder-se-á passar o turno sempre que desejar.

## Outras Informações

O programa possui algumas limitações, como a não ordenação das peças de uma sequência na mesa, a eventual falha da segunda função da função 1 para o caso de coringas e algumas eventuais incompatibilidades entre o que é mostrado na

mesa e a lógica de uma função, contudo a lógica geral do jogo Rummikub, bem como as jogadas básicas a se executar, estão presentes.

Outro detalhe a se observar é que, na obtenção das mãos iniciais de cada jogador, as peças não são distribuídas alternadamente. Contudo, usando uma lógica sequencial, pode-se ainda utilizar, no modo de jogo 2, o arquivo lido para o acompanhamento das peças obtidas.