

Introdução à Ciência da Computação I - SSC0600

# **Universidade de São Paulo**

## **Trabalho 1 - Quarto**

André Rennó de Campos 10298864

# 1. Introdução

## 1.1 Membros da Equipe

O projeto foi feito individualmente por André Rennó de Campos (Nº USP 10298864)

## 1.2 Descrição do Trabalho

O trabalho é uma implementação do jogo de tabuleiro "Quarto" criado pelo matemático Blaise Müller, na linguagem de programação C. O jogo é composto por um tabuleiro 4x4 e 16 peças únicas, com quatro características binárias, cor (vermelho ou azul), tamanho (alto ou baixo), forma (circular ou quadrado) e característica do topo (com buraco ou liso).

Para essa implementação, são usados no lugar das peças números de 0 a 15, e as características binárias das peças são representados pelos bits do número.

Durante o jogo, os jogadores se alternam em colocar na tabela o número que o outro escolheu outro e escolher um número (o primeiro a jogar somente escolhe um número). Ganha quem colocar um número que fizer uma linha que tenha determinada "característica" (bit) igual, por exemplo, a linha 13 9 11 3 (**1101 1001 1011 0011**) pois o último bit (em negrito) é igual em todos os números.

Essa implementação permite jogar o jogo com os números em hexadecimal, decimal, e binário.

# 2. Descrição do Projeto

## 2.1 Ambiente de Desenvolvimento

O projeto foi desenvolvido na distribuição "Arch Linux" do sistema operacional Linux, usando o editor de texto Vim.

## 2.2 Compilação e versões compiladas

O projeto foi compilado com o compilador gcc 7.3.1 em um sistema Linux x64 e com o MinGW (gcc 6.3.0) em um sistema windows x64. Para compilar foi necessário incluir a biblioteca *ncurses* com a flag `-lncurses` no linux e a biblioteca *pdcurses* com a flag `-lpdcurses` no windows.

A versão compilada no linux tem o nome "Quarto" e a versão compilada no windows tem o nome "Quarto.exe". Ambas estão na pasta `/bin/`.

## 2.3 Códigos Fonte

O projeto está dividido nos seguintes arquivos:

- main.c
- init.c
- print\_table.c
- check\_victory.c
- ai.c
- quarto.h

## 3. Tutorial

### 3.1 Compilação

Para compilar no linux, basta usar, em um sistema Linux com gcc instalado, o comando `gcc -o jogo main.c init.c print_table.c check_victory.c ai.c -lncurses` onde "jogo" é um nome qualquer para o arquivo compilado (por exemplo: Quarto)

No windows é necessário instalar o MinGW e a biblioteca `pdcurses` e usar o comando `gcc -o jogo.exe main.c init.c print_table.c check_victory.c ai.c -lpdcurses` no powershell, sendo que "jogo" é um nome qualquer para o arquivo compilado.

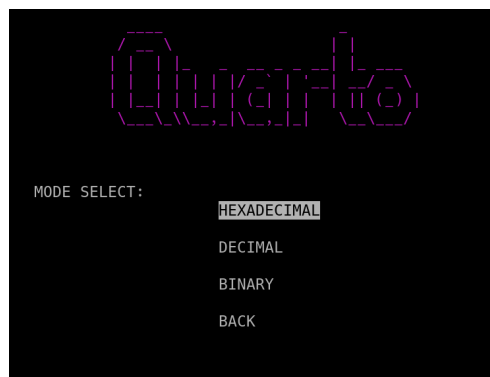
### 3.2 Execução

Antes de executar o programa, verifique que você tem instalado a livreria *pdcurses* (windows) ou *ncurses* (linux);

Para executar o programa basta usar o comando `./jogo` (ou clicar duas vezes no arquivo no windows) sendo "jogo" o nome do arquivo compilado em 3.1

Se você estiver no linux, o programa irá pedir para você apertar `ctrl +` até que o texto cubra o terminal inteiro. Isso é opcional, mas é melhor fazer isso agora do que durante o jogo. Aperte `enter` para ir para o menu.

No menu podemos selecionar entre as opções `1 jogador`, em que o jogador joga contra o computador, e `2 jogadores`, em que dois jogadores jogam um contra o outro, e depois disso



é possível, caso o modo 1 jogador tenha sido selecionado, selecionar quem começa primeiro, o computador ou o player.

Depois, para ambas as opções de número de jogadores, é possível selecionar o modo de jogo: Hexadecimal, Decimal e Binário; Essa opção determina em qual base os números serão mostrados no jogo.

C		2	
		3	A
	D	0	
F		E	

12		2	
		3	10
	13	0	
15		14	

1100		0010	
		0011	1010
	1101	0000	
1111		1110	

Exemplos de tabelas no modo hexadecimal, decimal e binário.

Depois de iniciado o jogo, as jogadas são feitas selecionando a peça ou posição desejada com as setas do teclado e apertando enter. O primeiro jogador deve selecionar apenas o número a ser colocado e, a partir daí, cada jogador coloca o número e seleciona outro.

O jogo acaba quando uma linha, coluna ou diagonal tiver o mesmo bit em determinada posição. Quando isso acontecer, o programa irá dizer quem ganhou e com quais posições, e será dada a opção de jogar novamente.

## 4. Outras informações.

### 4.1 Limitações da inteligência artificial

A "inteligência artificial" presente no programa são uma série de funções extremamente básicas que apenas verifica se algum movimento irá resultar em vitória (e retorna ele se houver) e para escolher um numero para o player verifica antes se o número não vai permitir que o player ganhe. Tudo além disso é feito aleatoriamente.