

# SME0230 - Introdução à Programação de Computadores

## Primeiro semestre de 2021

**Professoras:** Marina Andretta (andretta@icmc.usp.br) e

Franklina Toledo (fran@icmc.usp.br)

**Estagiário PAE:** Pedro Regattieri Rocha (pedro.regattieri.rocha@usp.br)

**Monitora:** Beatriz Martins Bassi (beatrizmb@usp.br)

### Trabalho: jogo *TateDrez* (jogo da velha + xadrez)

## 1 Grupos

- O trabalho será dividido em duas etapas, que poderão ser feitas em **dupla** ou **individualmente** (recomendamos fortemente que trabalhem em duplas).
- As duas etapas do trabalho devem ser feitas pelo mesmo grupo.
- Caso alguém queira mudar de grupo, deverá entrar em contato previamente alguma das professoras.

## 2 Submissão

- As duas etapas do trabalho deverão ser submetidas no Run Codes (código de matrícula M71D) por apenas um dos integrantes do grupo.
- Apenas o arquivo fonte deve ser entregue, em um arquivo `.c`.
- No início do arquivo `(.c)` deverá existir um comentário com o nome e o número USP de todos os integrantes do grupo.

## 3 Como funciona o jogo

O *TateDrez* (do espanhol, *Tatetí* - jogo da velha - e *ajedrez* - xadrez) é um jogo que mistura o jogo da velha com o jogo de xadrez. Nele, há um tabuleiro de xadrez  $3 \times 3$ , um bispo, uma torre e um cavalo brancos, e um bispo, uma torre e um cavalo pretos. Dois jogadores (**Jogador 1** e **Jogador 2**) alternam suas jogadas, o primeiro (**Jogador 1**) usando as peças brancas, e o segundo (**Jogador 2**) as peças pretas.

O tabuleiro começa vazio e, inicialmente, os jogadores, alternadamente, colocam cada uma de suas peças no tabuleiro, em uma casa vazia. O **Jogador 1** coloca sua primeira peça em uma casa vazia, em seguida o **Jogador 2** coloca sua primeira peça em uma casa vazia, depois o **Jogador 1** coloca mais uma de suas peças em uma casa vazia, e assim por diante. Depois de todas as 6 peças colocadas no tabuleiro, começa o jogo de fato.

Começando com o **Jogador 1**, ele escolhe uma de suas peças e faz o movimento como em um jogo de xadrez. Ou seja, a torre pode andar quantas casas quiser na vertical ou na horizontal; o bispo pode andar quantas casas quiser na diagonal; o cavalo anda em “L”, ou seja, uma casa na horizontal ou vertical e depois mais uma na diagonal. Os movimentos devem ser tais que a casa final esteja dentro do tabuleiro e esteja livre. Ao contrário do jogo de xadrez, aqui não é possível capturar peças do adversário.

Depois de feita a jogada de um jogador, passa-se a vez para o outro jogador, que escolhe uma peça e executa um movimento.

O jogo acaba quando ou um dos jogadores vence, ou acontece um empate.

Um jogador vence como no jogo da velha: quando há 3 de suas peças alinhadas, ou em uma linha, ou em uma coluna, ou em uma diagonal.

O empate acontece quando um jogador não tem jogadas possíveis para serem feitas. Ou quando a mesma configuração de tabuleiro acontece por 3 vezes. Para facilitar a implementação, vamos considerar o jogo empatado quando um jogador não tem jogadas possíveis ou quando ele já fez 30 jogadas e ninguém venceu.

Se quiser jogar uma versão eletrônica deste jogo, existe este aplicativo disponível para Android: [https://play.google.com/store/apps/details?id=com.ninedevelopments.tatedrez&hl=pt\\_BR&gl=US](https://play.google.com/store/apps/details?id=com.ninedevelopments.tatedrez&hl=pt_BR&gl=US)

## 4 Objetivo

O objetivo deste trabalho é desenvolver um programa em linguagem C que implementa duas versões do jogo *TateDrez*: uma em que dois jogadores usuários (humanos) jogam um contra o outro e uma em que um jogador humano joga contra o computador.

No início do programa, o usuário deverá escolher qual das versões será jogada. Deve ficar claro como os jogadores devem fazer para especificar suas jogadas. Na versão em que os dois jogadores (**Jogador 1** e **Jogador 2**) são usuários, o programa deverá mostrar na tela alguma representação dos diferentes personagens por meio de caracteres.

O **Jogador 1** sempre começa o jogo (ou seja, ele tem as peças brancas). No caso usuário *versus* computador, **considera-se o computador como Jogador 1**. Após a primeira exibição do tabuleiro, o programa irá aguardar a inserção da instrução de movimento no terminal. A instrução pode ser válida ou não, e a verificação deverá ser feita após a inserção; caso não seja válida, o programa imprime uma mensagem de erro e aguarda uma nova inserção.

Em cada inserção de jogada, a representação do tabuleiro será naturalmente modificada conforme a jogada feita e isso deve ser mostrado ao jogador. Depois da jogada, o programa deverá verificar se o jogador venceu o jogo ou se houve empate. Neste caso, uma mensagem é impressa na tela e o jogo termina. Caso contrário, passa-se a vez para o outro jogador.

## 5 Práticas obrigatórias e proibidas

É necessário realizar a indentação do código. É uma prática que torna o código legível e permite a manutenção com mais facilidade. Sugere-se que sejam utilizadas as duas formas comuns de indentação, como descrito a seguir por meio de exemplos arbitrários na Figura 1:

É necessário também comentar o código. Comentários que apenas descrevem em português, literalmente, o que o programa faz, geralmente não são muito úteis. São desejáveis comentários que

Figura 1: Exemplo de indentações para o código em linguagem C.

```
#include <stdio.h>

void imprimir_valor(int n) {
    printf("%d\n", n);
    return;
}

int main() {
    int i = 0;

    while (i < 10) {
        imprimir_valor(i);
        i++;
    }

    return 0;
}
```

```
#include <stdio.h>

void imprimir_valor(int n)
{
    printf("%d\n", n);
    return;
}

int main()
{
    int i = 0;

    while (i < 10)
    {
        imprimir_valor(i);
        i++;
    }

    return 0;
}
```

explicam o que trechos do programa fazem, facilitando o entendimento do código por outras pessoas (além dos próprios autores).

Não é necessário dividir o código da primeira parte do trabalho em funções (se você não sabe o que são funções, não se preocupe: isso será ensinado mais adiante na disciplina). Porém, caso o grupo opte por utilizar funções, todas deverão possuir uma instrução **return**, mesmo as funções declaradas como **void**.

Com o objetivo de promover as boas práticas de programação, **é expressamente proibida a utilização dos seguintes recursos de programação**: comando **goto** e variáveis globais. Se você não sabe o que são estes dois recursos, pode perguntar às professoras como eles funcionam. Mas só por curiosidade, já que eles **não** podem ser usados.

## 6 Datas de entrega

As duas etapas devem ser entregues no Run Codes, nas seguintes datas:

- Versão usuário x usuário: 27 de junho de 2021.
- Versão computador x usuário: 18 de julho de 2021.