

# PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Algoritmos e Técnicas de Programação

## TRABALHO PRÁTICO

Prof. Edwaldo Soares Rodrigues

### Instruções:

- I. O trabalho deverá ser feito em grupos de no máximo 2 estudantes.
- II. O trabalho deverá ser realizado usando a linguagem de programação C#.
- III. Deverão usar os conceitos aprendidos na disciplina Algoritmos e Técnicas de Programação, levando em consideração as melhores estruturas para representar os itens do jogo.
- IV. A data de entrega será disponibilizada em tarefa no Canvas.
- V. Somente poderão ser utilizadas estruturas e conceitos vistos em sala de aula.
- VI. A avaliação do trabalho será por meio de apresentação do código.
- VII. Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar;

### 1 - Objetivo

O Trabalho Prático deverá ser implementado usando a linguagem de programação C# e tem como objetivo colocar em prática os conceitos vistos, acerca de Algoritmos e Técnicas de Programação, ao longo do semestre.

### 2 - Descrição

Desenvolver uma aplicação que simule o jogo Ludo, onde jogadores “correm” uns contra os outros em um tabuleiro fechado. A cada rodada o jogador faz um lançamento de dado que permite mover um peão já posicionado no tabuleiro ou tirá-lo de sua casa se sorteado 6. O objetivo é ser o primeiro a completar o circuito com seus quatro peões à última casa.

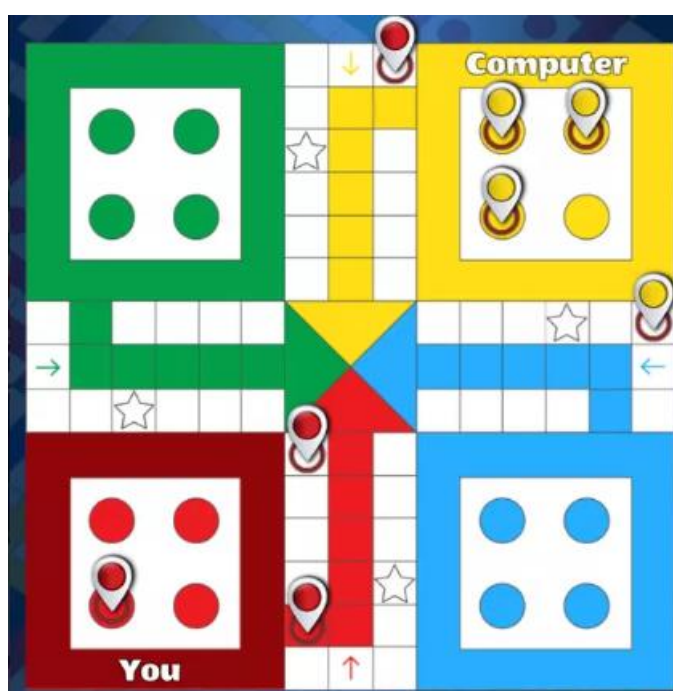
### 3 - Instruções

- Quatro (4) grandes quadrados, um com a cor de seu respectivo jogador

(vermelho, amarelo, verde, azul) que contém a posição inicial de seus quatro peões;

- Cinquenta e duas (52) casas que contornam o tabuleiro onde a corrida acontece. Destes, quatro (4) são coloridas e indicam o ponto de partida dos peões na trilha;
- Cinco (5) posições coloridas que levam ao triângulo central que indicam o caminho para o fim. Só podem ser usados pelo jogador de sua respectiva cor;
- Quatro (4) casas que contém estrelas que são pontos de segurança;

A seguir a Figura 1, apresenta uma imagem ilustrativa de um tabuleiro do jogo Ludo.



#### 4 – Possíveis Classes (São apenas sugestões)

- **Classe “Peao”** que represente cada peão, contendo atributos como “cor”, “posicao”, e métodos como: “Mover” e “RetornarCasa”.
- **Classe “Jogador”** que represente cada jogador, contendo atributos como “cor”, “peões” (um vetor de objetos Peao), e métodos como: “LançarDado”, “moverPeao”.
- **Classe “Tabuleiro”** que represente o tabuleiro do jogo, contendo atributos como “casas” (uma matriz, ou alguns vetores), e métodos como: “InicializarTabuleiro”, “VerificarCaptura”.
- **Classe “Jogo”** que representa o jogo em si, contendo atributos como “jogadores” (um vetor de objetos Jogador), “tabuleiro” um objeto da classe

Tabuleiro, “dado”, “jogoAtivo” e afins. Além de métodos como: “InicializarJogo”, “ExecutarRodada”, “AlternarJogador”, “VerificarVitoria” entre outros.

## 5 - Regras

Existem várias variações e diferentes regras para esse jogo. Para manter a uniformidade de implementação, serão dispostas as regras que a implementação deve atender. Variações não serão permitidas.

1. Ludo é jogado de 2 a 4 jogadores. Cada jogador possui 4 peões. Aquele que levar os 4 peões até o final (triângulo) primeiro, será declarado o vencedor (seu programa deverá permitir que o usuário entre com a quantidade de jogadores);
2. O peão só pode ser movido para o início da corrida, na casa de sua respectiva cor, quando um jogador lançar um 6 no dado;
3. Após colocar um peão no início do tabuleiro o jogador terá um novo lançamento do dado;
4. Se um jogador tirar 3 vezes seis em sequência deve-se passar a vez.
5. Para chegar à casa final (triângulo), deve-se tirar o valor exato remanescente e terá direito a outra rodada no dado; se não for o caso, ele deverá mover outra peça, se não for possível deve-se passar a vez;
6. É possível capturar o peão de outro jogador se cair em uma casa onde tomada por outro peão. Assim, o(s) peão(ões) adversários voltam, a posição inicial fora da trilha. Além disso quando uma captura ocorre o jogador pode lançar o dado mais uma vez;
7. As 4 casas que possuem cor são consideradas pontos seguros onde os peões adversários não podem ser capturados;
8. As 4 casas que possuem estrelas são consideradas pontos seguros onde os peões adversários não podem ser capturados;
9. Os peões devem se mover e viajar contornando o tabuleiro até chegar na última casa. A cada rodada do dado o jogador deve-se movimentar, ou seja, não é permitido abdicar do movimento; salve casos onde o movimento é ilegal;
10. Se um peão chegar ao ponto de origem, o jogador pode lançar o dado mais uma vez;
11. O programa deve permitir que o usuário escolha qual dos peões (quando possível) irá se mover;

## 6 - Requisitos

1. Comandos de repetição.
2. Utilização de métodos.
3. Utilização de Matriz(es) e/ou Vetor(es).
4. Nomes intuitivos para Métodos e Variáveis.
5. Permitir que o usuário escolha a quantidade de jogadores (2 ou 4).
6. Código indentado.
7. Contrário as boas práticas de programação, o código a ser entregue não deve conter nenhum comentário.
8. Uso de arquivos para registrar os logs da partida;

## 7 - Avaliação

O trabalho deve ser feito em grupos de no máximo 2 alunos. O trabalho deverá ser apresentado ao professor da disciplina no final do semestre. Trabalhos copiados, parcialmente ou integralmente, serão avaliados com nota **zero**, sem direito a contestação.

## 8 - Observações

Caso tenham dúvidas, recomendo que baixem o aplicativo: “Ludo Titan” que pode ser encontrado na Play Store e jogue o jogo Ludo, para entender melhor as regras do jogo.