

## Introdução

Gomoku é um jogo japonês milenar, jogado na China desde 700 DC, jogado em um tabuleiro de 15x15 células e peças pretas e brancas. Um jogador joga com as peças brancas, o outro joga com as peças pretas. O objetivo do game, como o nome sugere -- go-moku em japonês quer dizer cinco peças -- é colocar cinco peças da mesma cor (brancas ou pretas) consecutivamente ou numa mesma linha, ou numa mesma coluna, ou numa diagonal do tabuleiro, as peças são colocadas não podem ser movidas ou retiradas do tabuleiro.

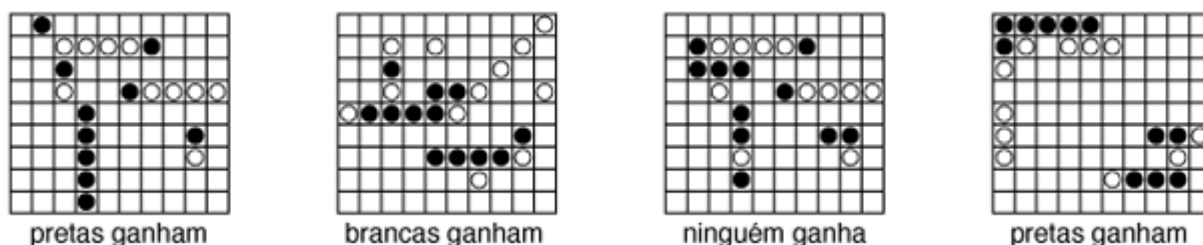


Figura 1: Configurações do game, apenas uma parte do tabuleiro é mostrada.

## Objetivo do trabalho

O objetivo deste trabalho é implementar um game de Gomoku, **em modo texto**. O tabuleiro do game deve ser armazenado em uma matriz inteiros com 15 linhas e 15 colunas, o número 1 representa a peça preta, o número 2 representa a peça branca e o número 0 indica que a posição (linha, coluna) está livre.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	2	0	0	2	2	2	1	0	0	0	0	0	0
0	0	1	2	0	0	0	0	1	0	0	0	0	0	0
0	0	0	1	2	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	2	2	0	0	0	0	0	0	0	0
0	0	1	1	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	2	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 2: Exemplo de uma matriz preenchida com um game

<sup>1</sup> **Importante:** A especificação desse trabalho pode sofrer modificações de acordo com discussões que tivermos em sala de aula.

No desenvolvimento do game as seguintes funções devem ser implementadas: **game()**, **print()**, **step()** e **status()** , caso julgue necessário você pode implementar funções.

- **game()**: É a função principal do programa e implementa um **gameloop** (while) que executa a lógica do game. No **gameloop**, dentro outras coisas, são tratados: a solicitação de uma jogada, a verificação do a jogada é válida e se alguém já conseguiu ganhar, exibir o tabuleiro na tela (a cada jogada), e também, a alternância do turno do jogador. Além disso, a função chama todos as funções descritas a seguir.
- **print()**: Essa função recebe o tabuleiro por parâmetro e imprime o tabuleiro do game na tela em modo texto, não se esqueça que a impressão deve ser o mais informativa possível, para o que jogador consiga decidir claramente qual jogada irá fazer.
- **step()**: Essa função recebe o tabuleiro, uma posição e algum outro parâmetro que julgue necessário por parâmetro, e verifica se é possível realizar a jogada, ou seja, se a posição solicitada está livre. Se a posição conseguiu ser preenchida, a função **step()** deve retornar **true** e, caso não puder ser preenchida, **false**. A função **game()** recebe o retorno e decide se troca a vez do jogador (**true**) ou solicita que o jogador refaça uma nova jogada (**false**).
- **status()**: E por fim, a função **status()** recebe o tabuleiro e algum outro parâmetro, e verifica o estado do game, ou seja, se alguém já ganhou ou houve empate. Para informar a função **game()** sobre o status do game a função pode retornar, por exemplo, **0** ainda ninguém ganhou, **1** o jogador com peças brancas ganhou, **2** o jogador com a peças pretas ganhou e **3** houve um empate

### **Critérios de avaliação:**

O programa entregue **será avaliado** de acordo com os seguintes itens:

- Funcionamento do programa;
- O programa deve estar na linguagem **Python**.
- Não se pode utilizar variáveis globais.
- O quão fiel é o programa quanto à descrição do enunciado;
- Comentários e legibilidade do código;
- Clareza na nomenclatura de variáveis e funções.

### **Grupo**

A atividade pode ser feita em grupo de no máximo dois integrantes, para entregar basta que somente um dos integrantes submeta o trabalho no Moodle com o nome dos integrantes do grupo no arquivo fonte.

### **Importante**

Como este trabalho pode ser feito em grupo, evidentemente você pode “discutir” o problema dado com outros grupos, inclusive as “dicas” para chegar às soluções, mas você deve ser responsável pela solução final e pelo desenvolvimento do seu programa. Assim não repasse para e nem copie o programa de outro grupo. Trabalhos considerados plagiados receberão nota 0 (zero).