

TCSS 435 Programming Assignment 2

NOTE: Be sure to adhere to the University's **Policy on Academic Integrity** as discussed in class. Programming assignments are to be written individually and submitted programs must be the result of your own efforts. Any suspicion of academic integrity violation will be dealt with accordingly

Assignment Details:

You are to write a program that will engage the user in a 2-person game of **Pentago**. Your program should use a minimax approach with several-move look-ahead and alpha-beta pruning. The rules of the game are summarized as follows:

Pentago is a 2-player game played on a **6x6** grid. The players alternate turns. The two players are referred to here as "**W**" and "**B**", which also signifies the colors of the tokens (white and black) they place on the board. The rules are summarized below or you can refer to this [Pentago Strategy!](#)

STAR: Start with an empty board and decide who starts, and who's playing what color.

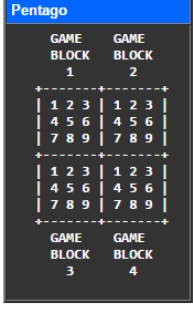
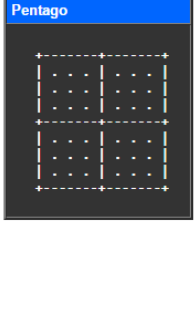
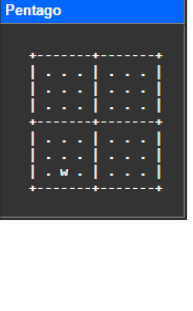
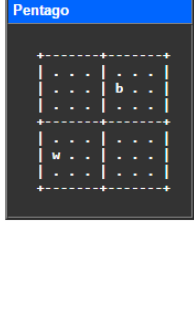
GOAL: The objective is to get five marbles in a row, in any direction, before your opponent does.

PLAY: Each turn consists of placing one marble, anywhere on the board and twisting any of the game blocks 90 degrees, in either direction. You can place your marble on one game block and twist any other game block.

WIN: First to five in a row wins!

Notes:

- If the board is filled without a winner declared, the game ends in a tie.
- Twisting the board can cause two players to win simultaneously, which is also a tie.
- It is possible to win the game by placing a token before twisting a game block
 - in this case the game block does not need to be twisted.

			
<p>Cell numbering. Each cell is numbered by its game block and its position, e.g., 3/8</p>	<p>Empty Board.</p>	<p>W moves first in this game, with the move 3/8 1R. Because game block 1 is empty, rotating it has no effect</p>	<p>B moves next, with the move 2/4 3R.</p>

Your program should adhere to the following conventions:

- Programs may be written in Java, Python (version 2.x).
- Your program should be able to make either the first or second move.
- Your program should be able to be either the player with w or b tokens.
- Your program should display a reasonable representation of a Pentago board after each of its moves and twists, and each of its opponent's moves and twists, with appropriate labeling (see below figure).
- Your program should be able to recognize and declare a winner. This should be checked after each token is placed and again after each board is twisted. Because it is possible that the opponent may have one or more lines with 5 tokens in a row after a twist, the opponent may also win after a twist! In the example below, player B gets 5 tokens in a row after the twist, but player W also gets 5 tokens in a row in two separate locations. While the rules do not discuss this circumstance explicitly, we will consider it a tie.
- **Specifying Moves:** The squares of the board are numbered by game *block* and *position*. The square **3/8** refers to Game **Block 3**, **position 8** (see figure). A move will have the form: **b/p bd**, where **b/p** is the block and position describing the location in which a token is placed, and **bd** is a block and direction for rotation. Unlike the game as described online, a player must provide a block and rotation for each move, even if empty blocks are present. Thus, the sample complete moves shown above: **3/8 1R** and **2/4 3L**. Your program should be able to accept input as either upper-case or lower-case letters interchangeably
- Your program should be able to start with an empty board. Your program should implement the game using both min-max and alpha-beta pruning.

You can choose if Human or AI should go first (randomly). Your program output (see below) should show the board configuration and the player moves for each turn. Write this information to the file.

Player 1 Name (player who moves first)	Alphonse
Player 2 Name	Gaston
Player 1 Token Color (B or W)	W
Player 2 Token Color (B or W)	B
Player to Move Next (1 or 2)	1
Next 6 lines: board state
Remaining lines: list of moves made,	...w..
in order from first to last,
alternating players.
	b.....

	2/4 1R
	3/4 4L

Pentago	Pentago	Pentago
<pre> +-----+ w . b . b w . w . b . w b . w . . w +-----+ b w . . . b w . b b b . w w +-----+ </pre>	<pre> +-----+ w . b . b w . w . b . w b . w . . w +-----+ b . . b . w . . . b w . b b b . w w +-----+ </pre>	<pre> +-----+ w . b . b w . w . b . w b . w . . w +-----+ b . . w . w w w b b b b b . +-----+ </pre>
<p>It is B's move. B makes the move 4/1 4L .</p>	<p>After B places a token in position 4/1 .</p>	<p>After B rotates Game Block 4 to the Left. Notice that B has 5 tokens in a row. However, W now also has <i>two</i> lines with 5 tokens in a row. The game is declared a tie.</p>

Submission Guidelines:

Submit your files on Canvas using the Programming Assignment 2 submission Link. **You will submit a zip file containing:**

- Source code with necessary source documentation for your program.
- Instruction.txt: Text file with instructions on how to run your program.
- Output.txt: A text file containing your output in the following format:
 - Player Info (Human/AI)
 - Player token (w/b)
 - Player move (includes – board/position, board rotation)
 - Configuration of board

This is the same console output that I should be able to see when I run your code.

- Readme.txt: Another text file containing the following information:
 - Minmax Algorithm: Number of nodes expanded, depth level for look-ahead, time and space complexity of your algorithm. Show me the number for nodes expanded for at-least two different depth level.
 - Minmax with Alpha-Beta Pruning: Number of nodes expanded, depth level for look-ahead, time and space complexity of your algorithm. Show me the number for nodes expanded for at-least two different depth level.