

# CONNECT FOUR

## Board Representation

- 2D array
- Output new board after every move
  - “\_” for empty
  - “0” for player 1
  - “X” for player 2

Player 1: 0

Player 2: X

Player 1's turn...

1	2	3	4	5	6	7
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	X	-	-	-	-	-
-	0	X	-	X	-	-
-	0	0	X	0	X	-
X	0	0	0	X	X	-

Enter a column:

-----



Player 1 has won!

Main Menu

-----

1. Vs Player
2. Vs CPU
3. Quit

## Game Initialization

- start a timer to keep track of how long game takes or track number of turns
- single player vs cpu or multiplayer
- let users choose
  - board size(option for 3 sizes: small (5x4), big(8x7), biggest(10x7)
- bool playerOneTurn
  - keep track of turns
  - true: player 1
  - false: player 2

## Functionality

- void printBoard(const string board[ ][COLUMN\_SIZE])
- bool checkWin()
  - Loop through 2D array checking for neighbors in a line
  - Check for 3 in a row in every direction (up, down, diagonals)
  - if true then output which player made the winning move and prompt the user if they want to start a new game
- void playPiece(int col)
  - User inputs the number of the column they want to place a piece into
  - Board reprints with updated state
- void nextTurn()
  - switch turn to false/true
  - increments turn counter
- void checkTie()
  - if board is filled, then print the final board and statement that the game is tied

## CPU

Easy:

Random placement

Hard:

Not random placement

Blocks player