# Exercise 1: TicTacToe

Write a two-player tic-tac-toe program. Prompt the players one at a time for their move (a number 1 through 9). Print the board after each move. Make sure moves are legal and detect a winning move and congratulate the winner!

### Suggestions:

1. Make a board-printing function that takes an int pointer and prints a nine-entry array treating the values:
   1. 0 means blank
   2. 1 means “x”
   3. 2 means “o”
2. Make a main and try printing various hard-coded array values.
3. Add “cin” and set “1” in the array. Check for range errors.
4. Add “cin” and set “2” in the array. Check for range errors.
5. Add check to make sure cell is empty.
6. Add a loop-forever.
7. Add checks for wins.
8. Spruce up the game messages and board printing.
9. Enhance board-printing to have “3” mean win-X and “4” mean win-O. When someone wins modify the board to show the winning move.

# Solution

#include <iostream>

using namespace std;

void printBoard(int\* board) {

char print[5] = {' ', 'x', 'o', 'X', 'O'};

// 3 rows to print

for(int y=0;y<3;++y) {

// 3 columns to print

for(int x=0;x<3;++x) {

cout << print[\*board];

// Print a divider line between columns 0 and 1

if(x==0 || x==1) {

cout << "|";

}

// Next square

++board;

}

// Print a line return

cout << endl;

// Print a divider line between rows 0 and 1

if(y==0 || y==1) {

cout << "-+-+-" << endl;

}

}

// Blank row at bottom of board

cout << endl;

}

void printDirections() {

cout << "2 player TIC TAC TOE" << endl << endl;

cout << "Square numbering:" << endl;

cout << " 1|2|3" << endl

<< " -+-+-" << endl

<< " 4|5|6" << endl

<< " -+-+-" << endl

<< " 7|8|9" << endl;

}

bool checkLine(int\* board, int start, int offset, int winMark) {

if(board[start]==0) return false;

if(board[start]==board[start+offset] && board[start]==board[start+offset\*2]) {

board[start] = winMark;

board[start+offset] = winMark;

board[start+offset\*2] = winMark;

return true;

}

return false;

}

bool checkForWin(int\* board, int winMark) {

// Check the rows

if(checkLine(board,0,1,winMark)) return true;

if(checkLine(board,3,1,winMark)) return true;

if(checkLine(board,6,1,winMark)) return true;

// Check the columns

if(checkLine(board,0,3,winMark)) return true;

if(checkLine(board,1,3,winMark)) return true;

if(checkLine(board,2,3,winMark)) return true;

// Check the diagonals

if(checkLine(board,0,4,winMark)) return true;

if(checkLine(board,2,2,winMark)) return true;

// No winning lines.

return false;

}

bool turnPlayer(int\* board, int marker, int winMarker, char name) {

// The current player's move

int move;

// Show the board

cout << endl;

printBoard(board);

// Loop until we get a good move

while(true) {

// Get the move

cout << "Enter a square number player " << name << " :";

move = 0;

cin >> move;

// Make sure the move is in range and the square is empty

if(move>0 && move<10 && board[move-1]==0) {

// Move is OK ... break out of input loop

break;

}

// Tell player to try again and go back for another input

cout << "Invalid move." << endl;

}

// Place the marker on the board

board[move-1] = marker;

// Check for a win

if(checkForWin(board,winMarker)) {

cout << endl << endl;

printBoard(board);

cout << "You win player " << name << " !!" << endl;

return true;

}

// Not a win

return false;

}

int main() {

// Initialize board to empty

int board[9] = {0,0,0, 0,0,0, 0,0,0};

// Print directions

printDirections();

// Enless play loop (we "return" on a win)

while(true) {

bool win = turnPlayer(board,1,3,'X');

if(win) return 0;

win = turnPlayer(board,2,4,'O');

if(win) return 0;

}

}