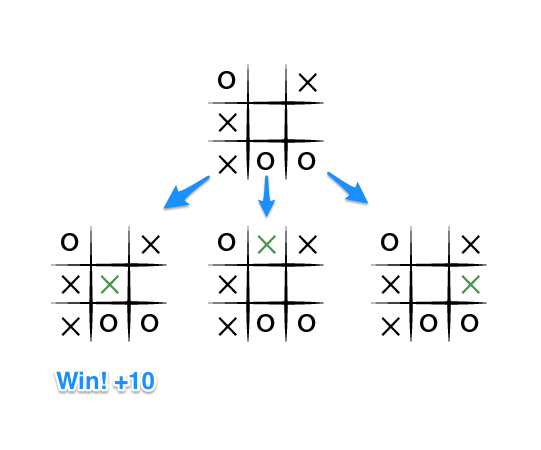
**The laboratory work 8**

**The tic-tac-toe game realization**

**Looking at a Brief Example**

To apply this, let's take an example from near the end of a game, where it is my turn. I am X. My goal here, obviously, is to *maximize* my end game score.



If the top of this image represents the state of the game I see when it is my turn, then I have some choices to make, there are three places I can play, one of which clearly results in me wining and earning the 10 points. If I don't make that move, O could very easily win. And I don't want O to win, so my goal here, as the first player, should be to pick the maximum scoring move.

At the first step we need to create the board and print it

#include "stdafx.h"

#include <iostream>

using namespace std;

void print\_board(int board[])

{

int i, j;

for (i = 0; i < 3; i++)

{

cout << " ";

for (j = 0; j < 3; j++)

{

if (board[i \* 3 + j] == 1)

cout << " X ";

else if (board[i \* 3 + j] == -1)

cout << " O ";

else if (board[i \* 3 + j] == 0)

cout << " ";

if (j != 2)

cout << " | ";

}

if (i != 2)

cout << "\n----------------" << endl;

else

cout << " ";

}

cout << "\n" << endl;

}

void print\_instructions()

{

cout << "\nThe board is laid out as folows:\n";

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

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

cout << " 4 | 5 | 6" << endl;

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

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

}

int main()

{

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

print\_instructions();

print\_board(board);

return 0;

}