Лабораторная работа по ООП. Нужно создать игру крестики-нолики на Си++ размером 10 на 10. Игра с компьютером. Код написать как можно легче, с комментариями. Желательно использовать код, данный ниже, модернизировав его

#include "stdafx.h"

#include <iostream>

#include <windows.h>

#include <time.h>

#define SIZE 3

using namespace std;

void startGame();

void printField();

int fieldValidation();

void movePlayer();

void movePC();

int field[SIZE][SIZE];

int main(void)

{

int choice;

while (1)

{

cout << "Tic Tac Toe" << endl << "Play? (1-yes, 0 - no)" << endl;

cin >> choice;

switch (choice)

{

case 1:

for (int i = 0; i<SIZE; i++)

for (int j = 0; j<SIZE; j++)

field[i][j] = 0;

startGame();

break;

case 0:

return 0;

default:

cout << "Invalid command" << endl;

break;

}

}

}

void startGame()

{

int k = 1;

printField();

while (k)

{

movePlayer();

printField();

k = fieldValidation();

if (k)

{

Sleep(1000);

movePC();

printField();

k = fieldValidation();

}

}

}

void printField()

{

cout << "\_\_\_\_\_\_\_" << endl;

cout << " 0 1 2" << endl;

for (int i = 0; i<SIZE; i++)

{

cout << i << " ";

for (int j = 0; j<SIZE; j++)

{

switch (field[i][j])

{

case 0:

cout << "\_ ";

break;

case 1:

cout << "X ";

break;

case 2:

cout << "O ";

break;

}

}

cout << endl;

}

cout << "\_\_\_\_\_\_\_" << endl;

}

void movePlayer()

{

int i, j;

while (1)

{

cout << "Enter numbers of cell" << endl;

cin >> i;

cin >> j;

if (field[i][j] == 0)

{

field[i][j] = 1;

return;

}

else

cout << "Cell is busy" << endl;

}

}

void movePC()

{

for (int i = 0; i<SIZE; i++)

if (field[i][0] == field[i][1] && field[i][0] != 0 && field[i][0] == 2 && field[i][2] != 1)

{

field[i][2] = 2;

return;

}

else if (field[i][1] == field[i][2] && field[i][1] != 0 && field[i][1] == 2 && field[i][0] != 1)

{

field[i][0] = 2;

return;

}

else if (field[i][0] == field[i][2] && field[i][0] != 0 && field[i][0] == 2 && field[i][1] != 1)

{

field[i][1] = 2;

return;

}

else if (field[0][i] == field[1][i] && field[0][i] != 0 && field[0][i] == 2 && field[2][i] != 1)

{

field[2][i] = 2;

return;

}

else if (field[1][i] == field[2][i] && field[1][i] != 0 && field[1][i] == 2 && field[0][i] != 1)

{

field[0][i] = 2;

return;

}

else if (field[0][i] == field[2][i] && field[0][i] != 0 && field[0][i] == 2 && field[1][i] != 1)

{

field[1][i] = 2;

return;

}

if (field[0][0] == field[1][1] && field[0][0] != 0 && field[0][0] == 2 && field[2][2] != 1)

{

field[2][2] = 2;

return;

}

else if (field[0][0] == field[2][2] && field[0][0] != 0 && field[0][0] == 2 && field[1][1] != 1)

{

field[1][1] = 2;

return;

}

else if (field[1][1] == field[2][2] && field[1][1] != 0 && field[1][1] == 2 && field[0][0] != 1)

{

field[0][0] = 2;

return;

}

else if (field[0][2] == field[1][1] && field[0][2] != 0 && field[0][2] == 2 && field[2][0] != 1)

{

field[2][0] = 2;

return;

}

else if (field[0][2] == field[2][0] && field[0][2] != 0 && field[0][2] == 2 && field[1][1] != 1)

{

field[1][1] = 2;

return;

}

else if (field[1][1] == field[2][0] && field[1][1] != 0 && field[1][1] == 2 && field[0][2] != 1)

{

field[0][2] = 2;

return;

}

/////////////Проверка если есть 2 крестика/////////////////////////

for (int i = 0; i<SIZE; i++)

if (field[i][0] == field[i][1] && field[i][0] != 0 && field[i][2] == 0)

{

field[i][2] = 2;

return;

}

else if (field[i][1] == field[i][2] && field[i][1] != 0 && field[i][0] == 0)

{

field[i][0] = 2;

return;

}

else if (field[i][0] == field[i][2] && field[i][0] != 0 && field[i][1] == 0)

{

field[i][1] = 2;

return;

}

else if (field[0][i] == field[1][i] && field[0][i] != 0 && field[2][i] == 0)

{

field[2][i] = 2;

return;

}

else if (field[1][i] == field[2][i] && field[1][i] != 0 && field[0][i] == 0)

{

field[0][i] = 2;

return;

}

else if (field[0][i] == field[2][i] && field[0][i] != 0 && field[1][i] == 0)

{

field[1][i] = 2;

return;

}

if (field[0][0] == field[1][1] && field[0][0] != 0 && field[2][2] == 0)

{

field[2][2] = 2;

return;

}

else if (field[0][0] == field[2][2] && field[0][0] != 0 && field[1][1] == 0)

{

field[1][1] = 2;

return;

}

else if (field[1][1] == field[2][2] && field[1][1] != 0 && field[0][0] == 0)

{

field[0][0] = 2;

return;

}

else if (field[0][2] == field[1][1] && field[0][2] != 0 && field[2][0] == 0)

{

field[2][0] = 2;

return;

}

else if (field[0][2] == field[2][0] && field[0][2] != 0 && field[1][1] == 0)

{

field[1][1] = 2;

return;

}

else if (field[1][1] == field[2][0] && field[1][1] != 0 && field[0][2] == 0)

{

field[0][2] = 2;

return;

}

int i, j;

srand(time(0));

i = rand() % (SIZE);

j = rand() % (SIZE);

while (1)

{

if (field[i][j] == 0)

{

field[i][j] = 2;

return;

}

else

{

i = rand() % (SIZE);

j = rand() % (SIZE);

}

}

}

int fieldValidation()

{

int full = 1, win = 0;

for (int i = 0; i<SIZE; i++)

if (field[i][0] == field[i][1] && field[i][1] == field[i][2]) // Проверка по горизонтали

{

win = field[i][0];

}

else if (field[0][i] == field[1][i] && field[1][i] == field[2][i]) // Проверка по вертикали

{

win = field[0][i];

}

if (field[0][0] == field[1][1] && field[1][1] == field[2][2]) // Проверка по диагонали

{

win = field[0][0];

}

else if (field[0][2] == field[1][1] && field[1][1] == field[2][0]) // Проверка по диагонали

{

win = field[0][2];

}

if (win)

{

switch (win)

{

case 1:

cout << "Player win!" << endl << "Congratulations!!!!" << endl;

return 0;

case 2:

cout << "PC win!" << endl << "Try again." << endl;

return 0;

}

}

for (int i = 0; i<SIZE; i++)

for (int j = 0; j<SIZE; j++)

if (field[i][j] == 0) // Проверка, на наличие свободных ячеек в поле

full = 0;

if (full)

{

cout << "Full field!" << endl;

return 0;

}

return 1;

}