Завдання:

Створити гру хрестики нулики.

Github: https://github.com/SolDima/labSP/tree/main/

Код гри:

#define \_CRT\_SECURE\_NO\_WARNINGS // Щоб Visual Studio не ругався на scanf

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

void CheckWin(char\*\* field, char Fieldrow, char \* peremoga);

void PrintField(char\*\* field, char FieldRow, char full);

void FillMasiv(char\*\* masiv, char row, char col, char element);

int main()

{

system("chcp 1251 > nul");

srand(time(NULL));

char choice = -1;

printf("Привіт! \nВітаю у моїй версії гри \"Хрестики нулики!\"\n");

do

{

printf("\n--- Меню ---\n");

printf("\nВведи 1 щоб запустити гру на двох");

printf("\nВведи 2 щоб запустити гру проти надсучасного ШІ");

printf("\nВведи 0 щоб вийти\n");

(void)scanf(" %c", &choice);

while (getchar() != '\n');

if (choice == '1' || choice == '2')

{

char Fieldrow = 3;

char\*\* field = (char\*\*)malloc(Fieldrow \* sizeof(char\*));

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

{ // створення масивів для поля і запису ходів

field[i] = (char\*)malloc(Fieldrow \* sizeof(char));

}

FillMasiv(field, 3, 3, ' ');

char hidRow = 2;

char hidCol = 9;

int\*\* hid = (int\*\*)calloc(hidRow, sizeof(int\*));

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

{

hid[i] = (int\*)calloc(hidCol, sizeof(int));

}

char hidnum = 0;

char peremoga = 0;

printf("Пропоную почати, преший крок за хрестик.\n");

printf("Шоб обрати клітинку для ходу, введи її координати.\n");

PrintField(field, Fieldrow, 1);

do

{

hidnow:

// Обробка ходу

if(choice == '2' && hidnum % 2 != 0)

{

hid[0][hidnum] = 1 + (rand() % 3);

hid[1][hidnum] = 1 + (rand() % 3); // SuperAI

}

else // Хід гравця

{

if (hidnum % 2 == 0)

printf("Зараз ходить X\n");

else

printf("Зараз ходить O\n");

(void)scanf(" %d %d", &hid[0][hidnum], &hid[1][hidnum]);

}

if (hid[0][hidnum] < 1 || hid[0][hidnum] > 3 ||

hid[1][hidnum] < 1 || hid[1][hidnum] > 3)

{

printf("Помилка вводу.\n");

hid[0][hidnum] = 0;

hid[1][hidnum] = 0;

goto hidnow; // goto

}

hid[0][hidnum]--;

hid[1][hidnum]--;

for (int i = 1; i <= hidnum; i++)

{

if (hid[0][hidnum] == hid[0][i - 1] && hid[1][hidnum] == hid[1][i - 1])

{

if (!(choice == '2' && hidnum % 2 != 0))

printf("Ой, тут вже хтось ходив. Обери іншу клітинку\n");

goto hidnow; // goto

}

}

if (hidnum % 2 == 0)

field[hid[0][hidnum]][hid[1][hidnum]] = 'X';

else

field[hid[0][hidnum]][hid[1][hidnum]] = 'O';

printf(" Хід %d \n", hidnum + 1);

PrintField(field, Fieldrow, 1);

if (hidnum >= 4)

CheckWin(field, Fieldrow, &peremoga); // перевірка перемоги і очищення поля

if (peremoga != 0)

{

FillMasiv(field, 3, 3, ' ');

printf("Хід гри:\n"); // показати хід гри

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

{

if (i % 2 == 0)

field[hid[0][i]][hid[1][i]] = 'X';

else

field[hid[0][i]][hid[1][i]] = 'O';

PrintField(field, Fieldrow, 0);

}

}

hidnum++;

if (hidnum == 9 && peremoga != 1)

printf("Перемогла дружба!");

} while (hidnum != 9 && peremoga == 0);

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

free (field[i]);

free(field);

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

free(hid[i]);

free(hid);

}

else if(choice != '0')

printf("Помилка вибору");

}while (choice != '0');

return 0;

}

void CheckWin(char\*\* field, char Fieldrow, char \* peremoga)

{

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

{

if (field[i][0] == field[i][1] &&

field[i][0] == field[i][2] &&

field[i][0] != ' ') // перевірка на вертикалі

{

printf("Кінець гри! Переміг %c \n", field[i][0]);

field[i][0] = '-';

field[i][1] = '-';

field[i][2] = '-';

PrintField(field, 3, 0);

\*peremoga += 1;

}

else if (field[0][i] == field[1][i] &&

field[0][i] == field[2][i] &&

field[0][i] != ' ') // перевірка на горизонталі

{

printf("Кінець гри! Переміг %c \n", field[0][i]);

field[0][i] = '|';

field[1][i] = '|';

field[2][i] = '|';

PrintField(field, 3, 0);

\*peremoga += 1;

}

}

if ((field[0][0] == field[1][1] &&

field[0][0] == field[2][2] &&

field[0][0] != ' ')) // головна діагональ

{

printf("Кінець гри! Переміг %c \n", field[1][1]);

field[0][0] = '\\';

field[1][1] = '\\';

field[2][2] = '\\';

PrintField(field, 3, 0);

\*peremoga += 1;

}

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

field[0][2] == field[2][0] &&

field[0][2] != ' ')) // побічна діагональ

{

printf("Кінець гри! Переміг %c \n", field[1][1]);

field[0][2] = '/';

field[1][1] = '/';

field[2][0] = '/';

PrintField(field, 3, 0);

\*peremoga += 1;

}

}

void PrintField(char\*\* field, char FieldRow, char full)

{

if (full == 1)

{

printf(" 1 2 3 \n");

printf("1 | %c | | %c | | %c |\n", field[0][0], field[0][1], field[0][2]);

printf("2 | %c | | %c | | %c |\n", field[1][0], field[1][1], field[1][2]);

printf("3 | %c | | %c | | %c |\n", field[2][0], field[2][1], field[2][2]);

}

else

{

printf("| %c | | %c | | %c |\n", field[0][0], field[0][1], field[0][2]);

printf("| %c | | %c | | %c |\n", field[1][0], field[1][1], field[1][2]);

printf("| %c | | %c | | %c |\n\n", field[2][0], field[2][1], field[2][2]);

}

}

void FillMasiv(char\*\* masiv, char row, char col, char element)

{

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

{

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

{

masiv[i][j] = element;

}

}

}



