Федеральное агентство связи

Сибирский государственный университет телекоммуникация и информатики

КУРСОВАЯ РАБОТА

по дисциплине «Программирование»

ВАРИАНТ 24

Выполнил: студент группы ИП-711

Иванов А. А.

Проверили:

доцент кафедры ПМиК Перцев И.В.

Новосибирск 2018

**Содержание**

1.Введение

2.Техническое задание

3.Код программы

4.Результат программы

**Введение**

**Задача**: создание приложения для игры в морской бой

**Описание:** «Морской бой» — игра для двух участников, в которой игроки по очереди называют координаты на неизвестном им поле соперника. Если у соперника по этим координатам имеется корабль (координаты заняты), то корабль или его часть «топится», а попавший получает право сделать ещё один ход. Цель игрока — первым потопить все корабли противника.

Игровое поле — обычно квадрат 10×10 у каждого игрока, на котором размещается флот кораблей. При размещении корабли не могут касаться друг друга сторонами и углами.

Размещаются:

1 корабль — ряд из 4 клеток

2 корабля — ряд из 3 клеток

3 корабля — ряд из 2 клеток

4 корабля — 1 клетка

**Техническое задание**

Реализация в консоли, написано на языке C.

Основная цель: потопить корабли противника раньше, чем твои.

При запуске появится консоль c пунктами меню: 1) Start 2) Exit.

Если выбран 1 пункт, то начнется игра, если 2, то выход.

Сначала игроку предлагается выбрать соперника, то есть заранее готовое поле.

После выбора будет предложена возможность выбрать уже созданное поле или создать свое.

При выборе второго пункта игрок должен расставить корабли по полю. Сначала нужно выбрать какой корабль он хочет поставить. Затем координаты начала и конца корабля. При этом программа будет проверять правильность введенных данных, то есть чтобы длина корабля по координатам была равна выбранному кораблю, пределы поля, соприкосновение кораблей, то что корабль это прямая, а также будет вестись учет уже использованных кораблей.

Далее игроку выводится на экран два поля. Одно это его, а другое это пустое поле в котором будут отмечаться ходы.

Игрок поочередно ходит с компьютером, вводя координаты выстрелов, при попадание отображается ‘X’, а промахе ‘o’.

Компьютер ходит при помощи рандома, то есть на вход ему подаются случайные числа от 0 до 9, но он не должен ходить по клеткам с попаданием или промахом.

Игра происходит до тех пор пока на каком-либо поле не останется целых кораблей(клеток).

**Код программы**

# include <stdio.h>

# include <conio.h>

# include <time.h>

# include <stdlib.h>

# include <ctype.h>

# include <Windows.h>

# include <math.h>

# include <string.h>

char symbolupfield = 205;

char symbolsidefield = 199;

char symbolship = 178;

char symbolhit = 88;

char symbolmiss = 111;

char fieldplayer[10][11] =

{

" ",

" ",

" ",

" ",

" ",

" ",

" ",

" ",

" ",

" ",

};

char fieldplayer1[10][11] =

{

" 1111",

" 1 ",

"1 ",

" ",

"1 ",

" ",

"1 ",

" 111 ",

"1 11 1",

"1 111 1",

};

char fieldenemy1[10][11] =

{

"1 ",

" ",

"1 ",

" 1",

"1 1",

" 1",

"1 1",

" 111 ",

"1 1",

"1 11 111 1",

};

char fieldenemy2[10][11] =

{

" ",

" 1111 ",

" ",

"1 1 ",

"1 1 1 1 1 ",

"1 1 1 1 1 ",

" ",

" 1 1 1 ",

" ",

" 1 ",

};

char fieldenemy3[10][11] =

{

"1 1 ",

" ",

" 1 1111 ",

"1 1 ",

"1 1 ",

" 1 ",

" 11 1 1 ",

" ",

" ",

"1 111",

};

char fieldmainenemy[10][11];

char fieldofassistance[10][11] =

{

" ",

" ",

" ",

" ",

" ",

" ",

" ",

" ",

" ",

" ",

};

void init(char field[][11]);

void visual(char field[][11]);

int test(int, int, int, int, int);

void positionoffield(char field[][11]);

void positionofenemyfield();

int main()

{

srand(time(NULL));

int i = 0, j = 0, n;

printf("1.Start\n2.Exit\nEnter the menu item:");

scanf("%d", &n);

while (n != 2)

{

if (n == 1)

{

printf("\nStart?\n");

\_getch();

for (i = 0; i < 5; i++)

{

printf(". ");

Sleep(500);

}

printf("\nGo!\n");

Sleep(500);

system("CLS");

positionofenemyfield();

init(fieldmainenemy);

system("CLS");

system("CLS");

int select, trigger = 0;

while (trigger == 0)

{

printf("\n\t1.Use already made field\n\t2.Make a field yourself\n\t");

scanf("%d", &select);

system("cls");

if (select >= 1 && select <= 2)

{

switch (select)

{

case 1:

{

for (i = 0; i < 10; i++)

{

for (j = 0; j < 11; j++)

fieldplayer[i][j] = fieldplayer1[i][j];

}

break;

}

case 2:

{

positionoffield(fieldplayer);

break;

}

}

trigger = 1;

}

else

{

Sleep(1000);

system("cls");

printf("\n\tINVALID INPUT\n");

Sleep(1000);

system("cls");

}

}

init(fieldplayer);

visual(fieldplayer);

printf("\nAre you ready for battle?\n");

\_getch();

for (i = 0; i < 5; i++)

{

printf(". ");

Sleep(500);

}

printf("\nGo!\n");

Sleep(500);

system("CLS");

int x1, y1, numbercells1 = 20,numbercells2=20, triggerfind = 0;

do

{

int triggerplayer = 0;

while (triggerplayer == 0)

{

visual(fieldplayer);

printf("\n");

visual(fieldofassistance);

printf("X Y of the shot:");

scanf("%d %d", &x1, &y1);

if (fieldmainenemy[y1][x1] != 'X' && fieldmainenemy[y1][x1] != 'o' && x1 <= 9 && y1 <= 9 && x1 >= 0 && y1 >= 0) triggerplayer = 1;

else

{

Sleep(1000);

system("cls");

printf("\n\tINVALID INPUT\n");

Sleep(1000);

system("cls");

}

}

if (fieldmainenemy[y1][x1] == symbolship)

{

do

{

fieldmainenemy[y1][x1] = 'X';

fieldofassistance[y1][x1] = 'X';

system("cls");

visual(fieldplayer);

printf("\n");

visual(fieldofassistance);

printf("\n\tYou hit!");

numbercells1--;

printf("\nPress any key to continue\n");

\_getch();

system("cls");

int triggerplayer1 = 0;

while (triggerplayer1 == 0 && numbercells1)

{

visual(fieldplayer);

printf("\n");

visual(fieldofassistance);

printf("X Y of the shot:");

scanf("%d %d", &x1, &y1);

if (fieldmainenemy[y1][x1] != 'X' && fieldmainenemy[y1][x1] != 'o' && x1 <= 9 && y1 <= 9 && x1 >= 0 && y1 >= 0) triggerplayer1 = 1;

else

{

Sleep(1000);

system("cls");

printf("\n\tINVALID INPUT\n");

Sleep(1000);

system("cls");

}

}

} while (fieldmainenemy[y1][x1] == symbolship && numbercells1);

}

if (numbercells1 <= 0) break;

if (fieldmainenemy[y1][x1] == ' ')

{

fieldmainenemy[y1][x1] = 'o';

fieldofassistance[y1][x1] = 'o';

system("cls");

visual(fieldplayer);

printf("\n");

visual(fieldofassistance);

printf("\n\tYou missed");

printf("\nPress any key to continue\n");

\_getch();

}

system("cls");

int triggercomp = 0;

while (triggercomp == 0)

{

x1 = rand()\*(9) / RAND\_MAX;

y1 = rand()\*(9) / RAND\_MAX;

if (fieldplayer[y1][x1] != 'X' && fieldplayer[y1][x1] != 'o') triggercomp = 1;

}

if (fieldplayer[y1][x1] == symbolship)

{

do

{

fieldplayer[y1][x1] = 'X';

visual(fieldplayer);

printf("\n");

visual(fieldofassistance);

printf("\n\tEnemy`s shot:%d %d\n\tEnemy had hit\n\tPress any key to continue\n", x1, y1);

numbercells2--;

\_getch();

system("cls");

int triggerenemy1 = 0;

while (triggerenemy1 == 0 && numbercells1)

{

x1 = rand()\*(9) / RAND\_MAX;

y1 = rand()\*(9) / RAND\_MAX;

if (fieldplayer[y1][x1] != 'X' && fieldplayer[y1][x1] != 'o') triggerenemy1 = 1;

else

{

Sleep(1000);

system("cls");

printf("\n\tINVALID INPUT\n");

Sleep(1000);

system("cls");

}

}

} while (fieldplayer[y1][x1] == symbolship && numbercells2);

if (numbercells2 <= 0) break;

}

if (fieldplayer[y1][x1] == ' ')

{

fieldplayer[y1][x1] = 'o';

visual(fieldplayer);

printf("\n");

visual(fieldofassistance);

printf("\n\tEnemy`s shot:%d %d\n\tEnemy had missed\n\tPress any key to continue\n", x1, y1);

\_getch();

}

system("cls");

} while (numbercells1 && numbercells2);

system("cls");

if (numbercells1 <= 0) {

visual(fieldplayer);

printf("\n");

visual(fieldofassistance);

printf("\n\tWINNER WINNER CHICKEN DINER!!!\n\tPress any key to continue");

}

else printf("\n\tYOU LOSE");

\_getch();

}

system("cls");

printf("1.Start\n2.Exit\nEnter the menu item:");

scanf("%d", &n);

}

}

void init(char field[][11])

{

int i = 0, j = 0;

for (i = 0; i<12; i++)

{

for (j = 0; j<13; j++)

{

char symb = field[i][j];

switch (symb)

{

case '1':

{

field[i][j] = symbolship;

break;

}

default:

{

field[i][j] = symb;

break;

}

}

}

}

}

void visual(char field[][11])

{

int i = 0, j = 0;

printf("\n\t");

for (i = 0; i<13; i++)

printf("%c", 205);

printf("\n\t%c 0123456789%c\n", 199, 199);

for (i = 0; i<10; i++)

{

printf("\t");

printf("%c%d", 199, i);

for (j = 0; j<10; j++)

printf("%c", field[i][j]);

printf("%c", 199);

printf("\n");

}

printf("\t");

for (i = 0; i<13; i++)

printf("%c", 205);

printf("\n\t");

}

int test(int x1, int y1, int x2, int y2, int type)

{

int i = 0, j = 0;

if (x1>9 || y1>9 || x2>9 || y2>9 || x1<0 || y1<0 || x2<0 || y2<0)

{

return 0;

}

if (x1 != x2 && y1 != y2)

{

return 0;

}

if (x1 == x2)

{

if (((y2 - y1) + 1) != type) return 0;

}

if (y1 == y2)

{

if (((x2 - x1) + 1) != type) return 0;

}

for (i = y1; i <= y2; i++)

{

for (j = x1; j <= x2; j++)

{

if (i + 1 >= 0 && i + 1 <= 9 && fieldplayer[i + 1][j] == symbolship) return 0;

if (i - 1 >= 0 && i - 1 <= 9 && fieldplayer[i - 1][j] == symbolship) return 0;

if (j + 1 <= 9 && j + 1 >= 0 && fieldplayer[i][j + 1] == symbolship) return 0;

if (j - 1 <= 9 && j - 1 >= 0 && fieldplayer[i][j - 1] == symbolship) return 0;

if (i + 1 >= 0 && i + 1 <= 9 && j + 1 <= 9 && j + 1 >= 0 && fieldplayer[i + 1][j + 1] == symbolship) return 0;

if (i + 1 >= 0 && i + 1 <= 9 && j - 1 <= 9 && j - 1 >= 0 && fieldplayer[i + 1][j - 1] == symbolship) return 0;

if (i - 1 >= 0 && i - 1 <= 9 && j - 1 <= 9 && j - 1 >= 0 && fieldplayer[i - 1][j - 1] == symbolship) return 0;

if (i - 1 >= 0 && i - 1 <= 9 && j + 1 <= 9 && j + 1 >= 0 && fieldplayer[i - 1][j + 1] == symbolship) return 0;

}

}

return 1;

}

void positionoffield(char field[][11])

{

int i = 0, j = 0;

int x1, y1, x2, y2, type, k = 10;

int ships[4];

for (i = 0; i<4; i++)

{

ships[i] = 4 - i;

}

while (k)

{

visual(fieldplayer);

printf("You have:\n");

for (j = 0; j < 4; j++)

{

printf("Ship%d: %d\n", j + 1, ships[j]);

}

printf("Select ship(1,2,3,4):");

scanf("%d", &type);

if (ships[type - 1]!=0 && type>=1 && type<=4)

{

printf("Beginning x,y of the ship:");

scanf("%d %d", &x1, &y1);

printf("End x,y of ship:");

scanf("%d %d", &x2, &y2);

if (y1 > y2)

{

int tmp = y1;

y1 = y2;

y2 = tmp;

}

if (x1 > x2)

{

int tmp = x1;

x1 = x2;

x2 = tmp;

}

if (test(x1, y1, x2, y2, type))

{

ships[type - 1]--;

for (i = y1; i <= y2; i++)

{

for (j = x1; j <= x2; j++)

fieldplayer[i][j] = '1';

}

system("CLS");

init(fieldplayer);

test(x1, y1, x2, y2, type);

k--;

}

else

{

Sleep(1000);

system("cls");

printf("\n\tINVALID INPUT\n");

Sleep(1000);

system("cls");

}

}

else

{

Sleep(1000);

system("cls");

printf("\n\tINVALID INPUT\n");

Sleep(1000);

system("cls");

}

}

}

void positionofenemyfield()

{

int select, i = 0, j = 0, trigger = 0;

while (trigger == 0)

{

printf("Select your enemy(1,2,3):");

scanf("%d", &select);

if (select >= 1 && select <= 3)

{

switch (select)

{

case 1:

{

for (i = 0; i < 10; i++)

{

for (j = 0; j < 11; j++)

fieldmainenemy[i][j] = fieldenemy1[i][j];

}

break;

}

case 2:

{

for (i = 0; i < 10; i++)

{

for (j = 0; j < 11; j++)

fieldmainenemy[i][j] = fieldenemy2[i][j];

}

break;

}

case 3:

{

for (i = 0; i < 10; i++)

{

for (j = 0; j < 11; j++)

fieldmainenemy[i][j] = fieldenemy3[i][j];

}

break;

}

}

trigger = 1;

}

else

{

Sleep(1000);

system("cls");

printf("\n\tINVALID INPUT\n");

Sleep(1000);

system("cls");

}

}

}

**Результат программы**



