Научно-исследовательская работа

Информатика

Разработка и реализация игровой механики “Голодный крестьянин” с использованием символьной графики средствами языка программирования C++

Выполнил(а):

Ведров Лев Александрович

учащийся 8 класса

ЧОУ «Школа «Шамир», Россия, г. Санкт-Петербург

Содержание

Введение

История компьютерных игр насчитывает уже множество лет, и современные игры далеко ушли от своих предшественников, как в графическом, так и геймплейном плане. В прошлом мощности компьютерной техники не позволяли использовать современный уровень графического оформления, что не мешало создавать интересные и увлекательные игры.

Язык программирования C++ это компилируемый, статически типизированный язык программирования общего назначения. Область его применения включает создание операционных систем, разнообразных прикладных программ, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также развлекательных приложений [1].

Актуальность: в реалиях современного общества уделяется большое внимание рабочим процессам, человек тратит много сил под грузом ответственности, которая ложится на его плечи. Причем интеллектуальная нагрузка растет быстрее чем физическая. За долгие годы эволюции человечества были изобретены множество игр, направленных на организацию мыслительного отдыха. Последние несколько десятков лет информационные технологии все плотнее входят в нашу жизнь. Язык программирования C++ является одним из самых популярных в мире [2].

Цель: получение практических навыков в разработке игровых механик и программировании на языке C++.

Для достижения поставленной цели требуется решить следующие **задачи**:

1. Изучить основы программирования на языке C++.
2. Разработать игровую механику.
3. Создание своего игрового движка с текстовой графикой на языке программирования C++.
4. Реализация игровой механики на основе разработанного игрового движка.

Игровая механика

Игрок берет на себя управление голодным “крестьянином”, целью которого является поедание яблок, появляющихся в случайных местах игрового поля. При съедании яблока существует шанс что главный герой отравится. Отравленный “крестьянин” имеет возможность сделать ограниченное количество ходов до “смерти”. При отравлении на каждый ход игрока существует вероятность появления лекарства на поле. Движение в игре возможно в четырех направлениях: вверх, вниз, влево, вправо. На каждое перемещение главного героя тратится один ход. Игровое поле закольцовано: при переходе через границы поля игрок появляется с другой стороны.

Во время игры изредка появляются бонусы: золотые и серебряные монеты.

Аркадная составляющая игрового процесса подчеркивается наличием системы подсчета очков. Вышеупомянутые бонусные монеты дают определенное количество очков, которые приведены в таблице 1.

Таблица 1. Игровые бонусы.

Название	Обозначение	Очки
золотая монета	$	+100 очков
серебряная монета	+	+50 очков
лекарство	!	Излечение от отравления & +10 очков
яблоко	@	+1 очко

В игре реализована система рангов. При достижении граничных значений, игроку присваивается ранг.

Таблица 2. Соответствие очков игрока к получаемым рангам.

Ранг	Очки
Нет ранга	0
PROSPECT I	200
PROSPECT II	400
PROSPECT III	600
PROSPECT ELITE	800
CHALLENGER I	1000
CHALLENGER II	1200
CHALLENGER III	1400
CHALLENGER ELIE	1600
RISIG STAR	1800
SHOOTING STAR	2000
ALL-STAR	2200
SUPERSTAR	2400
CHAMPION	2600
SUPER CHAMPION	2800
GRAND CHAMPION	3000

В меню можно выбрать уровень сложности игры, который влияет на вероятность отравления при съедании яблока и появления лекарства, которые приведены в таблице 3.

Таблица 3. Уровни сложности.

Уровень сложности	Вероятность отравления при съедании яблока	Вероятность в ход появления лекарства при отравлении
Салага	5%	25%
Новичок	10%	20%
Любитель	15%	15%
Профессионал	20%	10%
Легенда	30%	0.01%

Описание структуры данных

В данной работе я использую типы данных: char, int, bool и string.

Тип данных int – это целочисленный тип данных способный содержать положительные и отрицательные значения используются для представления чисел [3].

Тип данных string– класс с методами и переменными для организации работы со строками в языке программирования C++. Он включён в стандартную библиотеку C++ [3].

Тип данных bool – это логический тип данных способный содержать одно из двух значений: true и false [3].

Тип данных char – это целочисленный тип данных, который соответствует одному символу [3].

Табл. 4. Основные переменные.

Переменная	Тип	Байт [4]	Диапазон принимаемых значений [4]
Буфер игрового поля (массив)	Char	1	от 0 до 255
Буфер ввода пользователя			
Итерационные счетчики	Int	4	От -2 147 483 648 до 2 147 483 647

Координаты игрока			
Координаты яблока			
Счет игрока			
Координаты лекарства			
Координаты бонусных монет			
Уровень сложности			
Отравлен ли игрок	Bool	1	False или true
Существует ли на поле лекарство			
Существует ли на поле бонусные монеты			
Работа программы			
Массив с названиями рангов игрока	String	Динамическое выделение памяти	
Логин игрока			
Строка с ASCII графикой MENU			

Описание интерфейса пользователя

При запуске программы, перед пользователем появляется меню выбора, представленное на рис. 1.

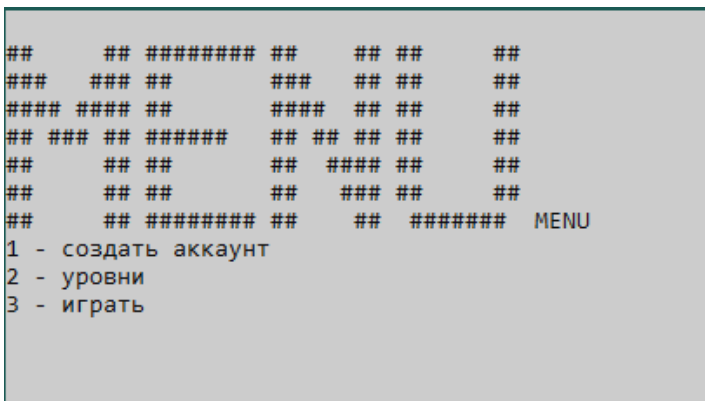

Рисунок 1. Меню программы.

У пользователь есть возможность задать свое игровое имя, выбрать уровень сложности игры и начать игру.

При задании игрового имени у пользователя будет возможность ввести его в поле, которое изображено на рис. 2.

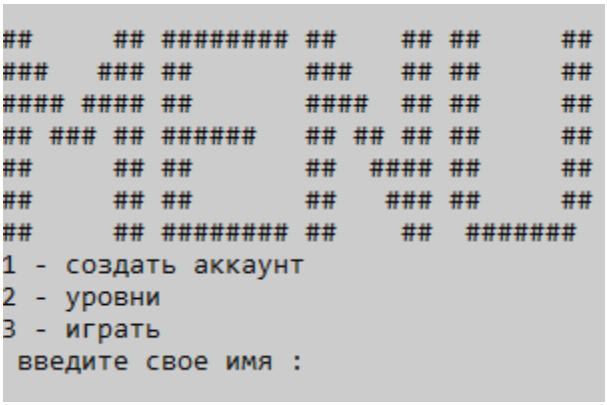

Рисунок 2. Меню выбора имени пользователя.

Для выбора сложности пользователю отображается меню, изображённое на рис.3.

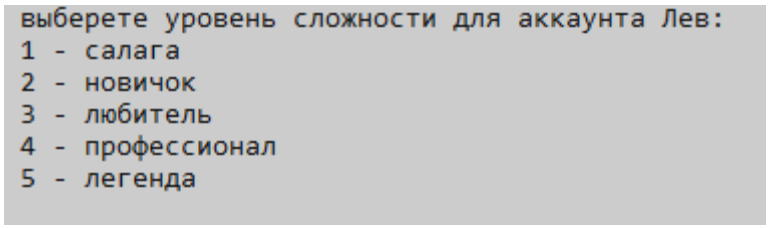

Рисунок 3. Меню выбора сложности.

При старте игрового процесса игрок видит в правой части экрана легенду обозначений и его текущий ранг. В левой половине экрана отображается игровое поле, на котором изображены игрок (#) и яблоко (@). Для перемещения к яблоку пользователю предлагается использовать клавиши “WASD”. В левом нижнем углу экрана отображается текущее количество набранных очков.

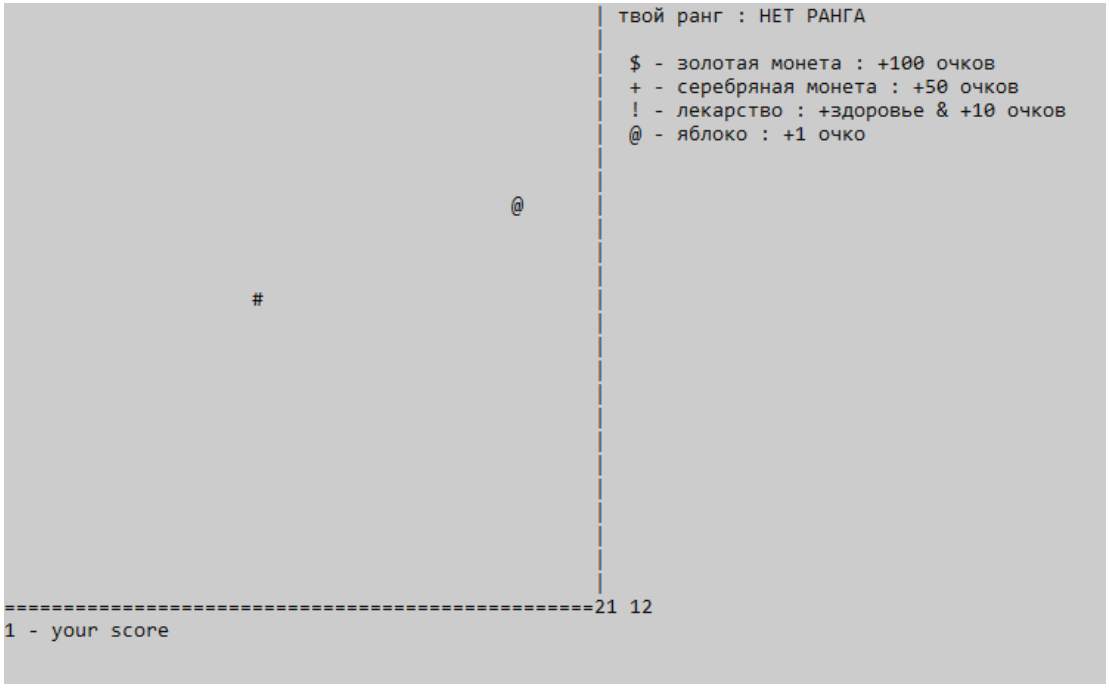

Рисунок 4. Начало игры.

На рисунке 5 изображен момент отравления игрока. При дальнейшем поедании яблок количество ходов до смерти увеличивается на 16. Также на поле появилось лекарство (!), съев которое, главный герой вылечится. На каждый ход игрока существует не большая вероятность, что лекарство изменит свое место нахождения.

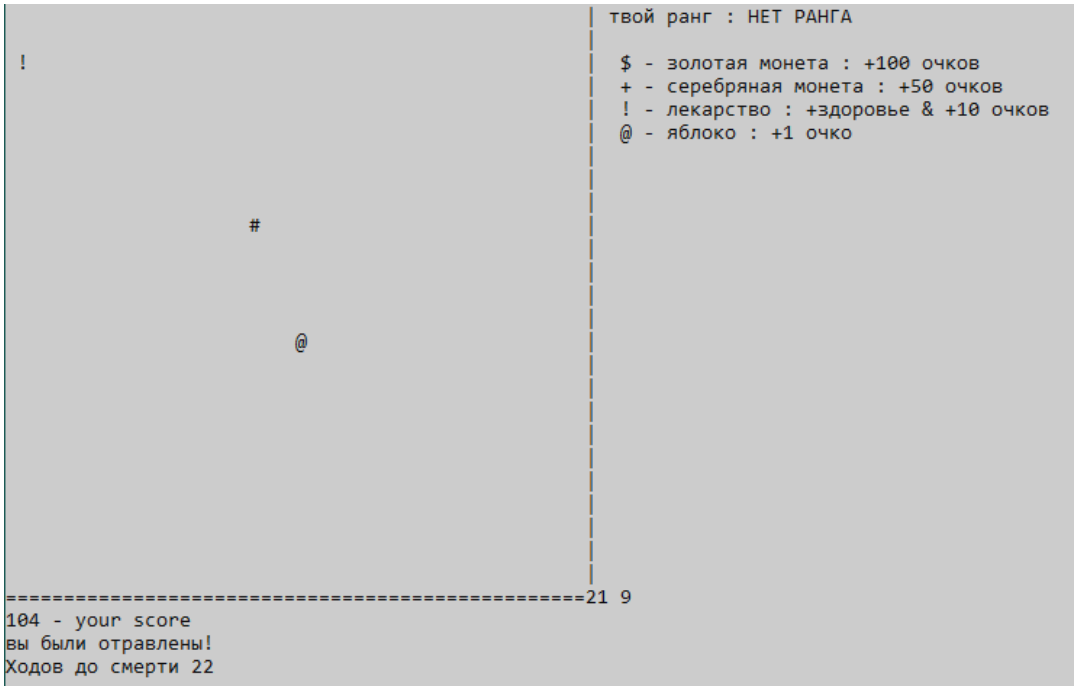

Рисунок 5. Отравление главного героя.

Когда счетчик ходов до смерти обнуляется, игроку отображается экран окончания игры. На нем зафиксирован счет пользователя и его ранг.

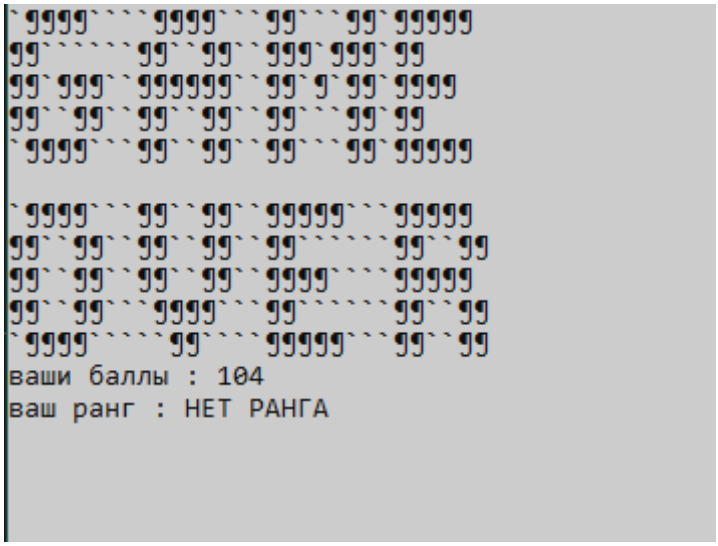

Рисунок 6. Окончание игры.

Заключение

В результате разработки и реализации игровой механики “Голодный крестьянин” с использованием символьной графики средствами языка программирования C++ были успешно освоены основы программирования на языке C++, получены практические навыки разработки игровых механик и алгоритмов.

Подобные задачи разработки и реализации игровых механик с использованием современных языков программирования позволяют развивать познания в области алгоритмизации, изучении языков программирования и получить практические навыки. Поэтому это будет полезно каждому при изучении нового, для него, языка программирования.

Список литературы

1. C++ (язык программирования) [Электронный ресурс] // Национальная библиотека им. Н. Э. Баумана: [сайт]. [2016]. URL: https://ru.bmstu.wiki/C%2B%2B_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) (дата обращения: 15.11.2020).

2. TIOBE Index for November 2020 [Электронный ресурс] // TIOBE: [сайт]. [2020]. URL: https://www.tiobe.com/tiobe-index// (дата обращения: 15.11.2020).

3. Типы данных [Электронный ресурс] // METANIT.COM Сайт о программировании: [сайт]. [2020]. URL: https://metanit.com/cpp/tutorial/2.3.php (дата обращения: 13.11.2020).

4. Диапазоны типов данных [Электронный ресурс] // Microsoft | Docs: [сайт]. URL: https://docs.microsoft.com/ru-ru/cpp/cpp/data-type-ranges?view=msvc-160 (дата обращения: 2.11.2020).

Приложение 1. Листинг кода игры “Голодный крестьянин”

```
#include <conio.h>
#include <iostream>
#include <windows.h>
#include <cstdlib>
#include <ctime>
using namespace std;
#define MAXX 50
#define MAXY 25
void printmypole(char** pole)
{
	int i=0, b=0;
	while (i <= 19)
	{
		while (b <= 19)
		{
			cout << pole[i+b*19];
			b = b + 1;
		}
		cout << "\n";
		i = i + 1;
	}
}
void init (int* x, int* y)
{
	*x = rand() % MAXX;
	while(*x%2==0)
		*x = rand() % MAXX;
	*y = rand() % MAXY;

}
void main(void) {
	void* handle = GetStdHandle(STD_OUTPUT_HANDLE);
	CONSOLE_CURSOR_INFO structCursorInfo;
	GetConsoleCursorInfo(handle, &structCursorInfo);
	structCursorInfo.bVisible = FALSE;
	SetConsoleCursorInfo(handle, &structCursorInfo);
	char GameSpace[MAXX][MAXY], UserInput;
	string gameover =
"`¶¶¶¶```¶¶¶¶```¶¶```¶¶`¶¶¶¶¶\n¶¶`````¶¶``¶¶``¶¶¶`¶¶¶`¶¶\n¶¶`¶¶¶``¶¶¶¶¶¶``¶¶`¶`¶¶`¶¶¶¶\n¶¶``¶¶``¶¶``¶¶``¶¶
```¶¶`¶¶\n`¶¶¶¶```¶¶``¶¶``¶¶```¶¶`¶¶¶¶¶\n
\n`¶¶¶¶```¶¶``¶¶`¶¶¶¶¶```¶¶¶¶¶\n¶¶``¶¶``¶¶``¶¶``¶¶`````¶¶``¶¶\n¶¶``¶¶``¶¶``¶¶``¶¶¶¶````¶¶¶¶¶\n¶¶``¶¶```¶¶¶¶
```¶¶`````¶¶``¶¶\n`¶¶¶¶````¶¶````¶¶¶¶¶```¶¶``¶¶";
	int i,b,User_x=MAXX/2,User_y=MAXY/2, FoodY, FoodX, score, deathcount=-1, drugsX=-1,
drugsY=-1, goldcoinX=-1, goldcoinY=-1, silvercoinX=-1, silvercoinY=-1, DifficultyLevel=0;
	bool poisoned = false, MedicineExist=false, goldcoin = false, silvercoin = false, on = true;
	string Ranks[16] = {"НЕТ РАНГА","PROSPECT I","PROSPECT II","PROSPECT
III","PROSPECT ELITE","CHALLENGER I","CHALLENGER II","CHALLENGER III","CHALLENGER
ELIE","RISIG STAR","SHOOTING STAR","ALL-STAR","SUPERSTAR","CHAMPION","SUPER
CHAMPION","GRAND CHAMPION"};
	string login;
	string menu;
	SetConsoleOutputCP(1251);
	SetConsoleCP(1251);

	srand(time(NULL));
```

```
	i = 0;
	score = 0;
	menu = "\n##    ## ######## ##    ## ##    ## \n###   ### ##      ###   ## ##    ## \n#### #### ## ####  ## ##    ## \n## ### ## ######   ## ## ## ##    ## \n##    ## ##      ##  #### ##    ## \n##    ## ##    ## \n##    ## ######## ##    ##  ######  ";



	system("color 70");
	while (on)
	{
		cout << menu;
		cout << "MENU\n1 - создать аккаунт\n2 - уровни\n3 - играть\n ";

		UserInput = _getch();
		switch (UserInput)
		{
		case '1':
			cout << "введите свое имя : ";
			cin >> login;
			cout << "добро пожаловать " << login << " , теперь вы можете играть.\n";
			break;
		case '2':
			if (login == "")
				cout << "создайте аккаунт чтобы играть\n";
			else
			{
				cout << "выберете уровень сложности для аккаунта " << login << ": \n 1 - салага\n 2 - новичок\n 3 - любитель\n 4 - профессионал\n 5 - легенда\n";

				UserInput = _getch();
				switch (UserInput)
				{
				case '1':
					DifficultyLevel = -5;
					break;
				case '2':
					DifficultyLevel = 0;
					break;
				case '3':
					DifficultyLevel = 5;
					break;
				case '4':
					DifficultyLevel = 10;
					break;
				case '5':
					DifficultyLevel = 20;
					break;

				}
			}
			break;
		case '3':
			if (login == "")
				cout << "создайте аккаунт чтобы играть\n";
			else
			{
```

```
while (i < MAXY)
{
	b = 0;
	while (b < MAXX)
	{
		GameSpace[b][i] = ' ';
		b = b + 1;


	}
	i = i + 1;
}
init(&FoodX, &FoodY);
while (deathcount != 0)

{
	i = 0;
	while (i < MAXY)
	{
		b = 0;
		while (b < MAXX)
		{
			if (b == User_x && i == User_y)
			{
				cout << '#';
				if (User_x == FoodX && User_y ==
FoodY)

				{
					init(&FoodX, &FoodY);
					score = score + 1;
					if (poisoned)
						deathcount = deathcount +
MAXX / 3;

					if (rand() % 100 >= 90 -
DifficultyLevel && !poisoned)

					{
						deathcount = MAXX / 2;
						poisoned = true;
					}
					if (rand() % 100 >= 98)
					{
						//создаем золотую монету
						init(&goldcoinX,
&goldcoinY);

						goldcoin = true;
					}
					if (rand() % 100 >= 75)

					{
						init(&silvercoinX,
&silvercoinY);

						silvercoin = true;

					}

				}
```

```
                                        if (User_x == drugsX && User_y ==
drugsY && MedicineExist)
                                        {
                                                score = score + 10;
                                                poisoned = false;
                                                MedicineExist = false;
                                        }
                                        if (User_x == goldcoinX && User_y ==
goldcoinY && goldcoin)
                                        {

                                                score = score + 100;
                                                goldcoin = false;
                                        }
                                        if (User_x == silvercoinX && User_y ==
silvercoinY && silvercoin)
                                        {
                                                score = score + 50;
                                                silvercoin = false;
                                        }
                                }
                                else
                                        if (b == FoodX && i == FoodY)
                                                cout << '@';
                                        else //тут вывод лекарства на экран
                                                if (b == drugsX && i == drugsY
&& MedicineExist)
                                                        cout << "!";
                                                else
                                                        if (b == silvercoinX && i
== silvercoinY && silvercoin)
                                                                cout << "+";
                                                        else
                                                                if (b == goldcoinX
&& i == goldcoinY && goldcoin)
                                                                        cout << "$";
                                                                else
                                                                        cout <<
GameSpace[b][i];

                                        b = b + 1;
                                }
                                cout << "|";
                                if (i == 0)
                                        cout << " твой ранг : " << Ranks[(score / 200)%16];
                                if (i == 2)
                                        cout << "  $ - золотая монета : +100 очков";
                                if (i == 3)
                                        cout << "  + - серебряная монета : +50 очков";
                                if (i == 4)
                                        cout << "  ! - лекарство : +здоровье & +10 очков";
                                if (i == 5)
                                        cout << "  @ - яблоко : +1 очко";
                                cout << "\n";
                                i = i + 1;
                        }
                        for (i = 0; i < MAXX; i++)
```

```
                                        cout << '=';
                                cout << User_x << " " << User_y << "\n";
                                cout << score << " - your score \n";
                                if (poisoned)

                                        cout << "вы были отравлены!\n" << "Ходов до смерти"
<< " " << deathcount-- << "\n";
                                if (rand() % 1000 >= 799 + DifficultyLevel * 10 && poisoned)
                                {
                                        init(&drugsX, &drugsY);
                                        MedicineExist = true;
                                }
                                UserInput = _getch();

                                if (UserInput == 'd' || UserInput == 'D' || UserInput == 'в' || UserInput
== 'В')
                                        User_x = (User_x + 2) % MAXX;
                                if (UserInput == 'w' || UserInput == 'W' || UserInput == 'ц' ||
UserInput == 'Ц')
                                {
                                        User_y = User_y - 1;
                                        if (User_y < 0)
                                                User_y = MAXY + User_y;

                                }
                                if (UserInput == 'a' || UserInput == 'A' || UserInput == 'ф' || UserInput
== 'Ф')
                                {
                                        User_x = User_x - 2;
                                        if (User_x < 0)
                                                User_x = MAXX + User_x;



                                }
                                if (UserInput == 's' || UserInput == 'S' || UserInput == 'ы' || UserInput
== 'Ы')
                                        User_y = (User_y + 1) % MAXY;

                                system("cls");
                        }
                        while (true)
                        {
                                cout << gameover << '\n';
                                cout << "ваши баллы : " << score << "\n";
                                cout << "ваш ранг : " << Ranks[(score / 200)%16];
                                _getch();
                                system("cls");
                        }
                        break;
                }
                default:
                        cout << "unkown operator\n";
                        }
        }
        }
```