|  |  |
| --- | --- |
| *voenmeh* | МИНОБРНАУКИ РОССИИ  федеральное государственное бюджетное образовательное учреждение  высшего образования  **«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова»**  **(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова»)** |
| БГТУ.СМК-Ф-4.2-К5-01 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Факультет |  | И |  | Информационные и управляющие системы |
|  |  | шифр |  | наименование |
| Кафедра |  | И9 |  | Систем управления и компьютерных технологий |
|  |  | шифр |  | наименование |
| Дисциплина |  | Программирование на языке высокого уровня | | |

КУРСОВАЯ РАБОТА

на тему

|  |
| --- |
| Разработка игры «Arkanoid» |
|  |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил студент группы | | | |  | И967 |
| Масанов Илья Вячеславович | | | | | |
| Фамилия И.О. | | | | | |
| **РУКОВОДИТЕЛЬ** | | | | | |
| Вальштейн К.В. | |  |  | | |
| Фамилия И.О. Подпись | | | | | |
| Оценка |  | | | |  |
| «\_\_\_\_\_» |  | | | | 2017г. |

САНКТ-ПЕТЕРБУРГ

2017 г.

Содержание

[ВВЕДЕНИЕ 3](#_Toc485902989)

[1 Формулировка задания 4](#_Toc485902990)

[2 Описание программы 6](#_Toc485902991)

[3 Игровой процесс 12](#_Toc485902993)

[ЗАКЛЮЧЕНИЕ 1](#_Toc485902994)5

[Список используемых источников 16](#_Toc485902995)

# **ВВЕДЕНИЕ**

Вдохновившись темой компьютерных игр, было решено разработать игру Arkanoid с удобным управлением и дружественным интерфейсом, где игроку предлагается управлять платформой-ракеткой и сбивать блоки шариком, отбивая его платформой.

Целью курсовой работы является создание собственной игры на основе объектно-ориентированного программирования, используя графическую библиотеку SDL 2.0.

Для достижения указанной цели требуется выполнить следующие задачи:

1. Продумать и создать несколько классов, описывающих объекты, участвующих в игровом процессе;
2. Познакомится с библиотекой SDL 2.0 и ее функциями;
3. Создать приложение на основе объектно-ориентированного программирования на языке с++.

# **1 Формулировка задания**

1. Разработать и написать игру «Arkanoid».
2. Цель игры: сбить все блоки на уровне за наименьшее время.
3. Правила игры:

* Игра продолжается до тех пор, пока не будут сбиты все блоки или пока шарик не упадёт на нижнюю грань игрового поля;
* Игра рассчитана на одного игрока;
* При попадании шариком в зелёный блок количество очков увеличивается на 1, и блок уничтожается;
* При попадании шариком в красный блок, блок превращается в зелёный;
* Во время паузы время игровой сессии не останавливается;
* При выходе из игры во время паузы, очки обнуляются;
* Уровни генерируются случайным образом;
* Ранг считается на основе полученных очков и затраченного на игровую сессию времени.

1. Требования к выполнению курсовой работы

* Программа при выполнении должна иметь дружественный интерфейс, включающий заставку, меню и удобные средства для управления программой;
* Заставка должна выводится на экран при запуске программы и давать представление об игре;
* Управление в игре осуществляется с помощью мыши и клавиатуры;
* Необходимо обеспечить достаточную скорость перерисовки динамически перемещаемых элементов;
* Во время игры должна быть возможность поставить её на паузу или завершить. Если реализована пауза, то из неё или идет продолжение, или выход в главное меню с завершением;
* Выполнение курсовой работы должно быть основано на принципах объектно-ориентированного программирования[1];
* Программа должна являться графическим приложением для Windows XP, написанным на языке C++ (с элементами C) с использованием библиотеки SDL 2.0.

## **2 Описание программы**

Программа написана на языке высокого уровня С++ (с элементами С) с использованием библиотеки SDL 2.0[3]. Среда разработки – CodeBlocks выбрана потому, что она является, в первую очередь, полностью бесплатной, а также легкой в освоении и отладке написанной программы.

В представленной программе используется пять файлов: главный файл «main» и подключенные к нему файлы “bblock”, “blocks”, “image” и “shar”.

В файле “bblock” находятся класс “bblock.

class b\_block

{

SDL\_Rect nblock;

SDL\_Surface \*img;

SDL\_Texture \*timg;

SDL\_Renderer \*render;

public:

b\_block(SDL\_Renderer \*);

~b\_block();

int rety();

void setmain();

int retx();

void Draw\_Block();

void setpos(int);

};

Класс[4] bblock состоит из полей: SDL\_Rect nblock, где хранятся координаты платформы, поля SDL\_Surface \*img – поверхность для хранения изображения, SDL\_Texture \*timg – текстура, полученная из поверхности img и поля SDL\_Renderer \*render, содержащего указатель на рендер.

**Краткое описание методов.**

1. Функции rety и retx возвращают текущие координаты платформы на игровом поле.
2. Метод setmain отвечает за установку начальных значений.
3. Функция setpos отвечает за установку значения координат платформы.

В файле «blocks» находится класс “blocks”, который описывает блоки, которые надо сбивать.

class blocks

{

struct qwe

{

SDL\_Rect r;

int var;

};

vector <qwe> a;

qwe temp;

int map[26][13];

public:

blocks();

void Draw\_Blocks(SDL\_Renderer \*ren);

int rety(int i);

int retx(int i);

int retvar(int i);

void load\_level();

void minus(int i);

void setmain();

int bsize();

void del(int i);

};

Состоит из полей:

* vector <qwe> a – массив блоков, каждый блок описывается структурой qwe, состоящей из полей: r – координаты блока и var – тип блока;
* qwe temp – вспомогательная переменная;
* int map[26][13] – карта расположения блоков на игровом поле.

Также состоит из функций:

* Draw\_Blocks – отрисовка блока;
* rety – возвращает координату “у” блока;
* retx – возвращает координату “х” блока;
* retvar – возвращает значение, соответствующее типу блока;
* load\_level – генерирует уровень;
* minus – функция, изменяющая тип блока;
* setmain – установка начальных значений;
* bsize – возвращает количество оставшихся на поле блоков;
* del – удаляет блок

В файле “image” находится класс “image”, предназначенный для отрисовки элементов игры, таких как: пауза, меню, начало и конец игры.

class image

{

SDL\_Surface \*background, \*menu, \*text;

SDL\_Texture \*tbackground, \*tmenu, \*ftext;

SDL\_Renderer \*render;

public:

image(SDL\_Renderer \*ren);

~image();

void draw\_begin();

void draw\_pause();

void draw\_background();

void level\_cleared(double a);

void draw\_end();

void draw\_menu(int menu);

};

Состоит из полей:

* SDL\_Surface \*background, \*menu, \*text – поверхности с изображениями заднего фона, фона меню и вспомогательного поля text, для рендера текста;
* SDL\_Texture \*tbackground, \*tmenu, \*ftext – текстуры, полученные из поверхностей \*background, \*menu, \*text;
* SDL\_Renderer \*render – указатель на рендер.

Также состоит из функций:

* draw\_begin – отрисовка интерфейса начала игры;
* draw\_pause – отрисовка интерфейса паузы;
* draw\_background – отрисовка заднего фона;
* level\_cleared – отрисовка интерфейса при удачном завершении игры;
* draw\_end – отрисовка интерфейса при неудачном завершении игры;
* draw\_menu – отрисовка интерфейса в меню.

В файле “shar” находится класс “shar”, описывающи й объект “шарик”.

class shar

{

SDL\_Rect RECT;

int gy, gx, alf, fx, radius;

public:

shar();

void setgy();

void setgx();

void setfx(int x);

void setalf(int x);

void setmain();

void Draw(SDL\_Renderer \*rend);

void SDL\_RenderFillCircle(SDL\_Renderer\* rend, int rad);

int rety();

int retx();

int retgx();

int retalf();

int retgy();

int retfx();

void next\_step();

};

Состоит из полей:

* SDL\_Rect RECT – содержит координаты шарика.
* int gy – скорость шарика по “у”
* int gx – скорость шарика по “х”
* int alf – угол отклонения шарика
* int fx – вспомогательная переменная
* int radius – радиус шарика.

Также состоит из функций:

* setgy – установка значения gy;
* setgx – установка значения gx;
* setfx – установка значения fx;
* setalf – установка значения alf;
* setmain – установка начальных значений;
* Draw и SDL\_RenderFillCircle – отрисовка шарика;
* rety – возвращает координату “у” шарика;
* retx – возвращает координату “х” шарика;
* retgx – возвращает gx;
* retgy – возвращает gy;
* retalf – возвращает alf;
* retfx – возвращает fx;
* next\_step – установка новых значений координат шарика.

Полный листинг программы представлен на внешнем носителе.

## **3 Игровой процесс**

При запуске программы пользователь в главное меню, показано на рисунке 1.

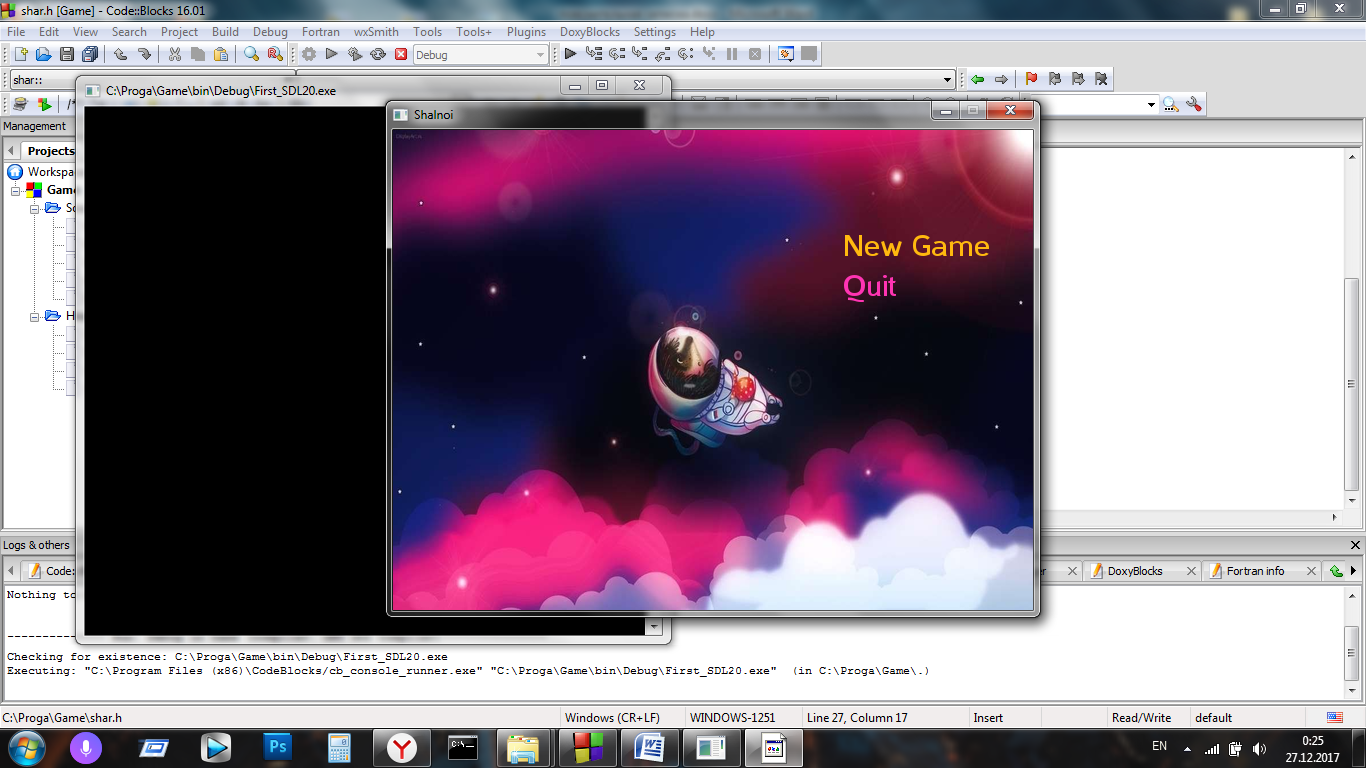


Рисунок 1 – Главное меню

После нажатия на поле “New Game” отрисовывается интерфейс начала игры (рисунок 2).

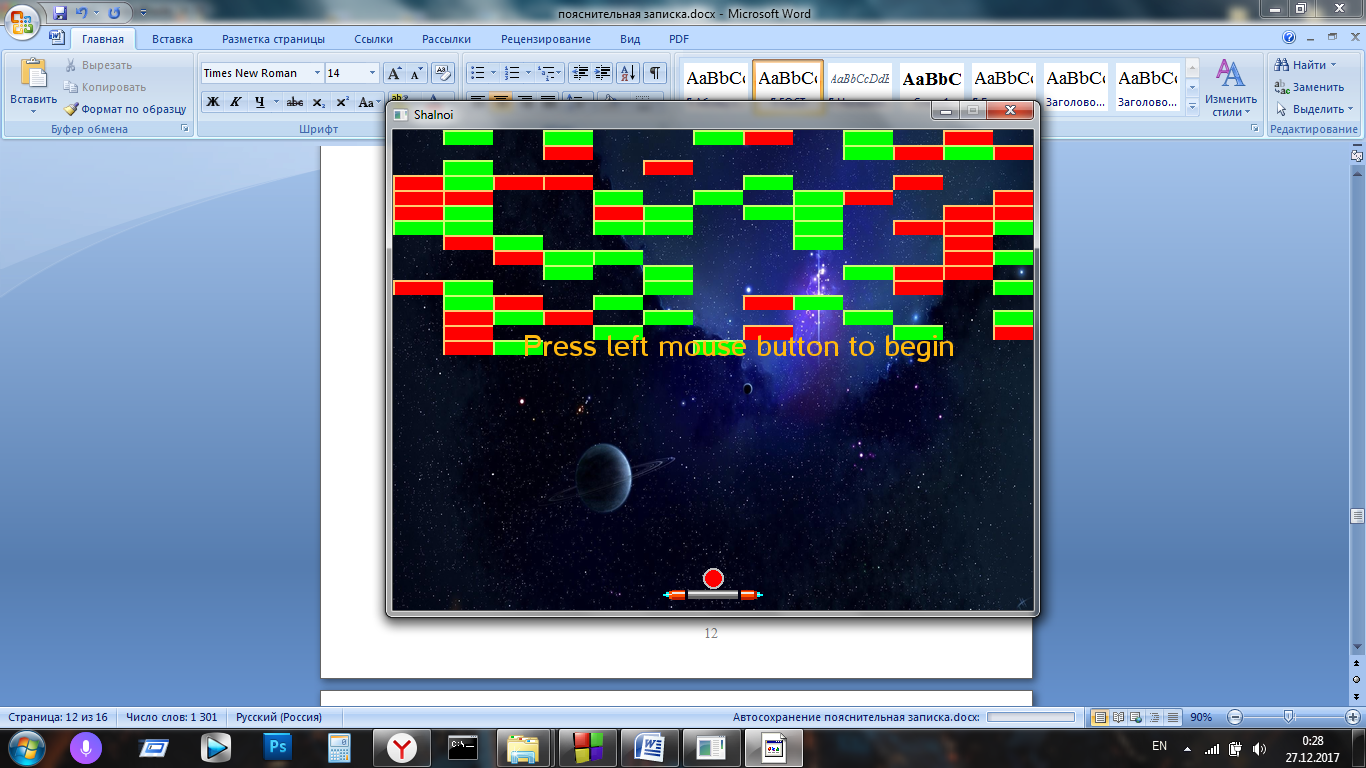


Рисунок 2 – Интерфейс начала игры

При нажатии на левую кнопку мыши начинается игра. Вид игрового поля во время игры показан на рисунке 3.

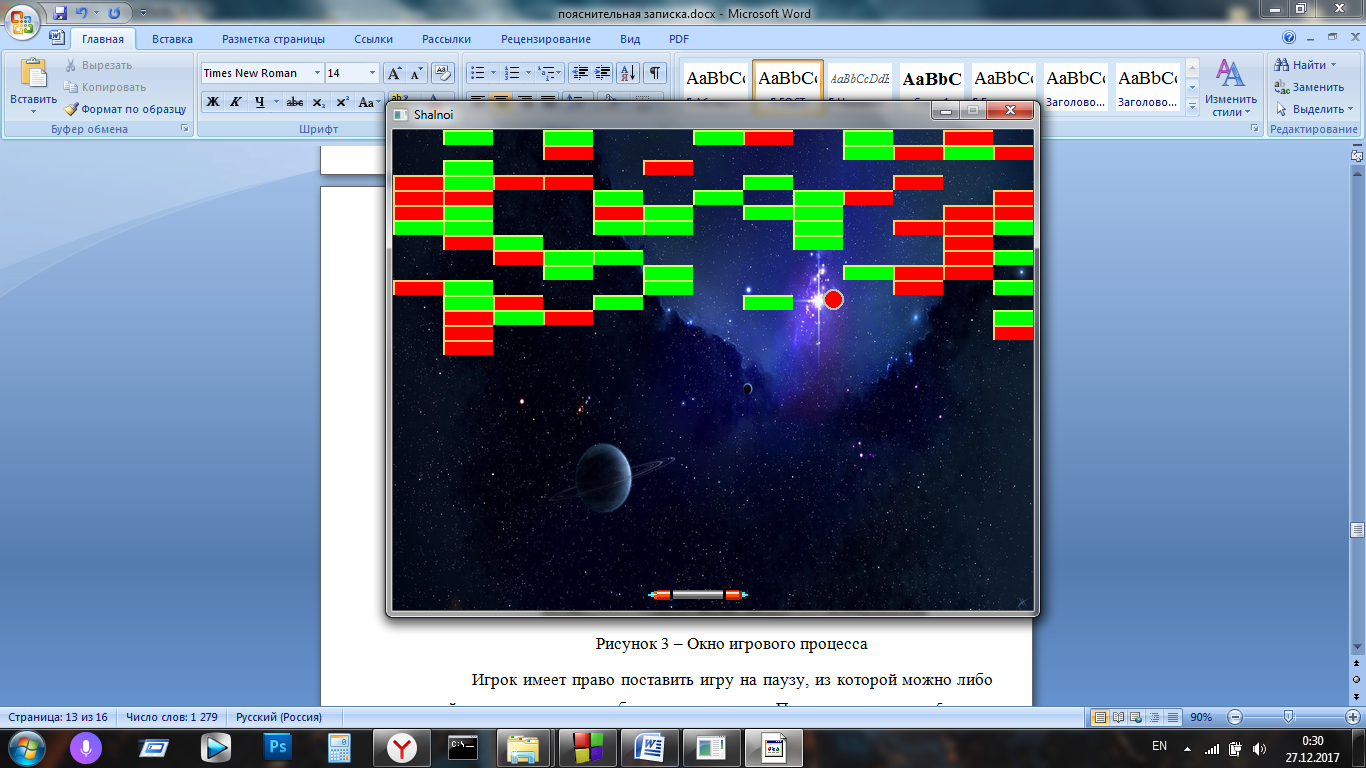


Рисунок 3 – Окно игрового процесса

Игрок имеет право поставить игру на паузу, из которой можно либо выйти в главное меню, либо продолжить игру. Показано на рисунке 4.

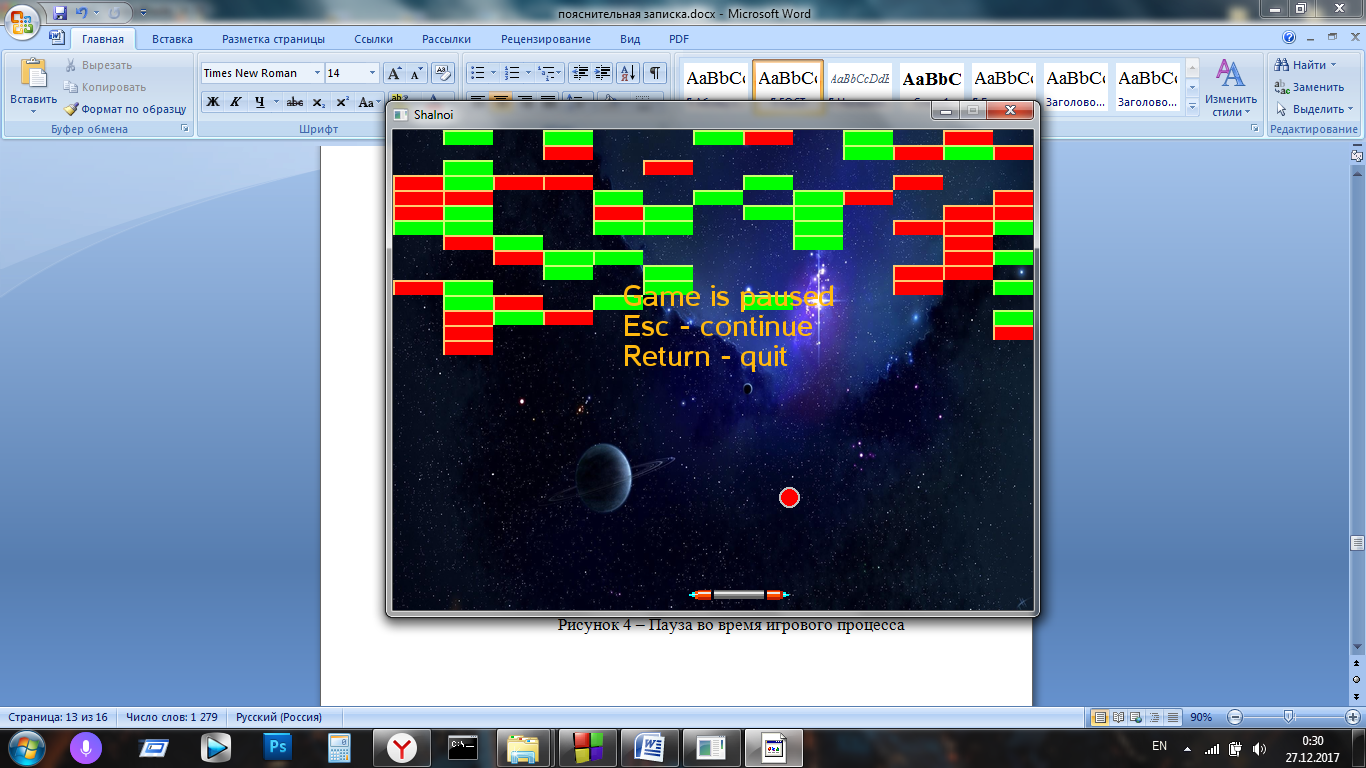


Рисунок 4 – Пауза во время игрового процесса

При падении шарика на нижнюю грань игрового поля, игра заканчивается, интерфейс при окончании игры показан на рисунке 5. При нажатии на клавишу Enter, пользователь попадёт в главное меню.

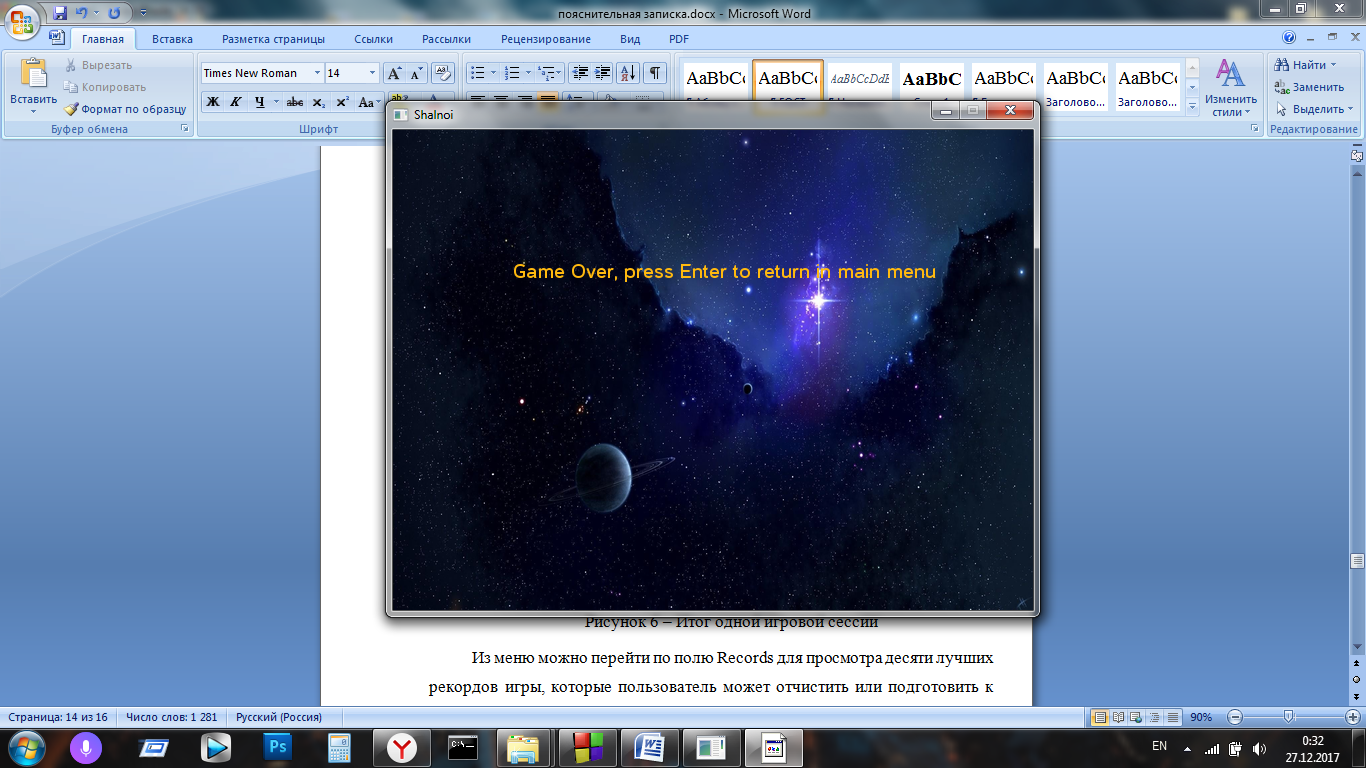


Рисунок 5 – Итог одной игровой сессии

По окончании игры происходит отрисовка интерфейса окончания игры (рисунок 6) с полученным рангом, ранг высчитывается как отношение затраченного времени к полученным очкам.

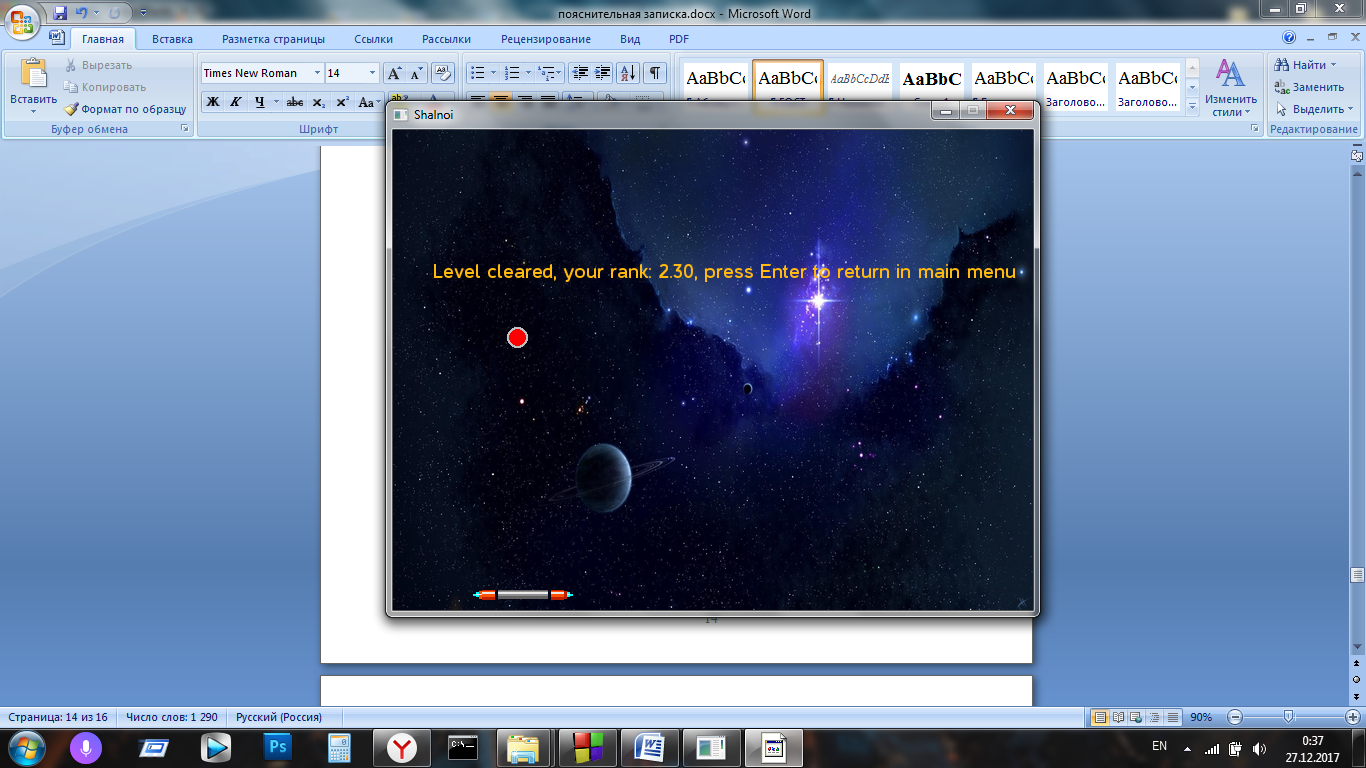


Рисунок 6 – Окончание игры

# **ЗАКЛЮЧЕНИЕ**

Были описаны объекты игры, такие, как шарик, блоки, движущаяся платформа, произошло знакомство с библиотекой SDL и ее функциями, Результатом курсовой работы стала полноценное, самостоятельно выполненное приложение на базе объектно-ориентированного программирования, с использованием библиотеки SDL 2.0. Все задачи работы выполнены, цель достигнута.

# **Список используемых источников**

1. Лафоре, Р. Объектно ориентированное программирование в С++ - 4-е издание /Москва: Питер, 2004. -922 с.
2. Скляров, В.А. Язык С++ и объектно-ориентированное программирование: учебное пособие / Минск: Вышейшая школа, 1997 – 481 с.
3. Арипова, О.В. Программирование на языке высокого уровня: лаб. практикум / О.В. Арипова, А.Н. Гущин, О.А. Палехова; Балт. гос. техн. ун-т. – СПб., 2014. – 184 с.
4. Шилдт, Г. Самоучитель С++ - 3-е издание, переработанное и дополненное / Санкт-Петербург: БВХ-Петербург, 2003 – 674 с.