Управление образование Администрация Сосновского муниципального района

Муниципальное бюджетное общеобразовательное учреждение

«Сосновская средняя школа №1»

**Создание компьютерной игры средствами языка программирования PascalABC.Net**

проектная работа

Автор

Сергунин Родион Эдуардович

10 класс

Руководитель

Е.С.Ильина, учитель информатики

п. Сосновское

2019

Оглавление

[Введение 3](#_Toc5304465)

[I. Что такое компьютерная игра 4](#_Toc5304466)

[1. Зарождение компьютерных игр 4](#_Toc5304467)

[2. Классификация компьютерных игр 6](#_Toc5304468)

[II. Разработка игр с помощью программирования 8](#_Toc5304469)

[1. Этапы проектирования игры 8](#_Toc5304470)

[2. Языки программирования для разработки игр 8](#_Toc5304471)

[3. Способы графического изображения в PascalABC.NET 10](#_Toc5304472)

[III. Описание создания игры «The Maze» 13](#_Toc5304473)

[Заключение 21](#_Toc5304474)

[Источники информации 22](#_Toc5304475)

# Введение

В современном мире компьютерные игры актуальны, возможно, они уже стали неотъемлемой частью жизни молодого поколения. Плохо это или хорошо – вопрос, но ответ на него никак не влияет на реальную действительность. Множество игр и программ создаются каждый день, они совершенствуются, а впоследствии некоторые становятся популярными

Людей, осознающих, что игры являются не только развлечением, но и частью искусства в целом, становится больше. В последнее время игры доказывают, что могут являться способом самовыражения, не уступающим произведениям литературы и живописи. Погружаясь во вселенную компьютерной игры, я задумываюсь, можно ли самостоятельно создать её с помощью каких-либо программ. Попытка ответить на этот вопрос привела меня к мысли провести исследование по этой теме. Для своего исследования я сформулировал **проблемный вопрос:** как создать компьютерную игру?

**Цель проекта:** создать компьютерную игру.

**Задачи:**

* Изучить и обобщить информацию по созданию компьютерных игр;
* Выбрать наиболее подходящую программу для создания компьютерной игры;
* Создать проектный продукт – свою компьютерную игру.

# I. Что такое компьютерная игра

Компьютерная игра – компьютерная программа, служащая для организации игрового процесса (геймплея), связи с партнёрами по игре, или сама выступающая в качестве партнёра.

По некоторым играм проводятся любительские и профессиональные соревнования, называемые киберспортом. Компьютерные игры также с 2011 года официально признаны отдельным видом искусства.

Развитие компьютерных игр тесно связано с совершенствованием компьютерного обеспечения и технологий. Многие составные части компьютеров создаются и разрабатываются чуть ли не для игр в первую очередь. Игры разрабатываются с учётом последних новинок компьютерных технологий, подходя всё ближе к реальности. Уже сейчас есть игры, поражающие своей правдоподобностью, почти полностью имитирующие реальную жизнь.

## 1. Зарождение компьютерных игр

Эдвард Кондон был назначен заместителем директора по исследованиям Westinghouse Electric в 1937 году. Идея создания игры ним родилась у Кондона и его коллеги во время обеда, и произошло это от осознания того, что такие же схемы, которые использовались в счётчиках Гейгера, могут использоваться для представления чисел, которые задают состояние игры. Эдвард Кондон разработал на этой основе компьютерную игровую электронно-релейную машину, которую собрал со своими ассистентами Джеральдом Тоуни и Уиллардом Дерром. В своем изобретении он отметил как то, что компьютер предназначен для развлечения, так и то, что он является примером «как можно для принятия решений собрать набор электрических реле в соответствии с довольно простой математической процедурой».

Разработанное устройство было предназначено для представления Westinghouse Electric на Всемирной выставке 1939—1940 годов в Нью-Йорке. Это была машина с электромеханическими реле, которая весила более тонны и занимала одну комнату. Посетителям выставки предлагалось обыграть машину. Согласно собранным данным, за всё время выставки в игру сыграло около 100 000 человек, из которых 90 000 не смогли обыграть машину, и при этом значительная часть победивших людей составляют операторы машины, которые демонстрировали, что у компьютера можно выиграть.

В последний раз компьютер участвовал в выставке в 1942 году в Нью-Йорке, что проходило при соглашении с Союзной ассоциацией общественных наук и спонсорстве Института математической статистики и Американской статистической ассоциации. В последующем Nimatron был перемещён в научные коллекции Планетария в Питтсбурге, где также представлялся публике.

26 апреля 1940 года авторы изобретения подали заявку на патент, которая 24 сентября того же года была удовлетворена.

В предлагаемой игре ним участвует два игрока, роль одного из которых берёт на себя компьютер. Интерфейс текущего состояния партии представляет собой четыре столбца по семь ламп, каждая из которых либо горит, либо погашена. Игрок может погасить несколько ламп в одном из столбцов, и после этого наступает ход компьютера, который может делать то же самое. Выигрывает тот, кто погасит последнюю лампу. Если машина проигрывала, то она выдавала игроку жетон, на котором было напечатано Nim Champ.

Одной из особенностей игры является то, что машина делает паузу перед тем, как сделать очередной ход. Это было сделано после того, как на выставке было обнаружено, что сам факт того, что машина моментально делает ответный ход, ужасает людей. По оценке Кондона, ответный ход компьютера рассчитывался за менее чем одну сотую секунды. Для того, чтобы это не обескураживало игроков, которые обдумывали свой ход, разработчики добавили в схемы замедляющие цепи, и, соответственно, машина на пару секунд делала вид, что ей нужно подумать. По заключению Кондона, это было первое в истории намеренное замедление работы компьютера.

## 2. Классификация компьютерных игр

Компьютерные игры в основном классифицируются по жанрам, а также по количеству игроков.

Игра может принадлежать по жанру как к одному, так и несколько жанрам сразу, а иногда - быть вне любого жанра или открыть новый.

По количеству игроков игра может быть рассчитана на одного игрока – однопользовательская, или многопользовательская.

**Классификация по жанрам:**

**Action (Экшн)** - жанр компьютерных игр, в которых успех игрока в большой степени зависит от его скорости реакции и способности быстро принимать тактические решения. Действие таких игр развивается очень динамично и требует высокую концентрацию внимания и быстрой реакции на происходящие в игре события.

**Shooter (Шутер)** - в играх данного типа игрок, как правило, действуя в одиночку, должен уничтожать врагов при помощи стрелкового оружия (чаще всего огнестрельного оружия и энергетического) и оружия ближнего боя (как правило холодного) для достижения определённых целей на данном уровне. Обычно после достижения заданных целей игрок переходит на следующий уровень. ( Примеры: Counter-Strike, Far Cry, Half-Life и другие)

**Fighting** (**Файтинг**) - жанр компьютерных игр, имитирующих рукопашный бой малого числа персонажей в пределах ограниченного пространства, называемого ареной. Важной особенностью файтингов является их нацеленность на соревнование, а не на сотрудничество игроков, что делает игры этого жанра подходящими для киберспортивных чемпионатов. Обычно файтинги предоставляют игроку возможность вести бой в режиме «один на один» против компьютерного противника или другого игрока, реже - позволяют сражаться одновременно трём или четырём противникам на одной арене. (Примеры: Mortal Kombat, Street Fighter)

**Arcade (Аркада)** - жанр компьютерных игр, характеризующийся коротким, но интенсивным игровым процессом. (Пример: Super Mario)

**Simulator (Симулятор)** - имитатор (обычно механический или компьютерный), задача которого состоит в имитации управления каким-либо процессом, аппаратом или транспортным средством.(Примеры:AirCraft Simulator, FIFA, PES, GameDevTycoon)

**Strategy (Стратегия)** - жанр компьютерных игр; игры данного жанра характеризуются тем, что игроку для достижения цели необходимо применять стратегическое мышление, и оно противопоставляется быстрым действиям и реакцией, которые, как правило, не обязательны для успеха в таких играх. (Примеры:Civilization, Total War, StarCraft)

**Adventure (Приключение)** - игра-повествование, в которой управляемый игроком герой продвигается по сюжету и взаимодействует с игровым миром посредством применения предметов, общения с другими персонажами и решения логических задач.

**Puzzle** (**Головоломка**) - жанр компьютерных игр, целью которых является решение логических задач, требующих от игрока задействования логики, стратегии и интуиции. (Примеры: Portal, TheRoom)

**3. Способы создания игр**

 Большинство людей, играющих в компьютерные игры, даже не задумываются, какое множество трудов и творческих идей вложено в каждую игру. Создание игры – это продолжительный и трудоёмкий процесс.

Есть два способа создания игр. Я могу создать игру, в основе которой лежит какой-либо движок, либо игру, с помощью какого-либо языка программирования.

 Я решил создавать игру с помощью языка программирования. Я выбрал языки программирования, потому что интереснее будет «прописывать» код игры вручную.

# II. Разработка игр с помощью программирования

## 1. Этапы проектирования игры

1. Идея, жанр, сеттинг.

Игру можно придумать заранее, а можно разобраться во время разработки. Жанр нужно выбирать в первую очередь в обязательном порядке. Он будет основным направлением игры.

Сеттинг- среда, в которой происходит действие; место, время и условия действия. Например, объектами в игре «Монополия» могут быть рынки Нью-Йорка, а могут предприятия бывшего СССР — игра в обоих случаях остаётся одной и той же, меняется лишь её сеттинг.

2.Средства.

Одна из самых сложных работ – написание программного кода. Сначала я должен выбрать язык программирования, который наиболее подходит. После этого предстоит работа по написанию кода.

## 2. Языки программирования для разработки игр

* **Java:** сильно типизированный объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems, выпущен в 1995 году.

Его плюсы: обширный функционал, простота изучения, программы могут выполняться на любом оборудовании.

Минусы: чрезмерная нагрузка на оперативную память оборудования, время выполнения программ в 2 раза меньше, чем, например, у C++.

**JavaScript:** Разработан Бренданом Эйхом в 1995 году. JavaScript обычно используется как встраиваемый язык для программного доступа к объектам [приложений](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0). Наиболее широкое применение находит в [браузерах](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) как язык сценариев для придания [интерактивности](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D1%81%D1%82%D1%8C) [веб-страницам](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0%22%20%5Co%20%22%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0). .Плюсы: с использованием написанных на JavaScript плагинов и скриптов справится даже неспециалист, Полезные функциональные настройки, Взаимодействие с приложением может осуществляется даже через текстовые редакторы – Microsoft Office и Open Office.

Минусы: Множество мелких раздражающих ошибок на каждом этапе работы, большая часть из них легко исправляется, но их наличие позволяет считать этот язык менее профессиональным, сравнительно с такими языками, как C++, Java и т.п.

**C++:** был разработан Бьёрном Страуструп в начале 1980-х годах. Область его применения включает создание операционных систем, разнообразных прикладных программ, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также игр.

Плюсы: высокая скорость работы программ, универсальность, возможность работы программ на любом оборудовании. Минусы:Сложность изучения, большое количество обязательных строк.

* **PascalABC.NET:** В [2005](https://ru.wikipedia.org/wiki/2005)—[2006 годах](https://ru.wikipedia.org/wiki/2006_%D0%B3%D0%BE%D0%B4) система была полностью переработана: изменена её архитектура — на полноценный компилятор языка, близкого к [Delphi](https://ru.wikipedia.org/wiki/Delphi_%28%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%29%22%20%5Co%20%22Delphi%20%28%D1%8F%D0%B7%D1%8B%D0%BA%20%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%29), с расширениями, связанными с платформой [.NET](https://ru.wikipedia.org/wiki/Microsoft.NET). Новая система получила название PascalABC.NET. В основном используется для обучения.

Плюсы: простота изучения, может работать на современных версиях Windows.

Минусы: малый функционал, устарел.

Мне не подходит язык C++, т.к. его сложно изучить.

Я мог написать игру как на Java, так и на JavaScript, но я решил выбрать PascalABC.NET, т.к. его мне не нужно будет изучать с нуля. Не смотря на то, что он устаревший, мне хватит его функционала для создания игры. К тому же я хочу познакомиться с новым функционалом «Паскаля», так как в процессе обучения практически не используется направление работы с графикой в среде программирования.

## 3. Способы графического изображения в PascalABC.NET

**Модуль GraphABC**

Модуль GraphABC представляет собой простую графическую библиотеку и предназначен для создания несобытийных графических и анимационных программ в процедурном и частично в объектном стиле, т.е. мы можем как «рисовать» объекты вручную, написав программу, так и просто вставив картинку.

Например, «Рисуем» оранжевый квадрат с помощью прямого написания программного кода:

**uses** GraphABC;

**begin**

line(100,100,1000,100);

line(100, 100, 100, 1000);

line(1000,1000,1000,100);

line(1000,1000,100,1000);

floodfill(150,150,clorange)

**end**.



Рис.1. Использование простейших операторов в модуле GraphABC.

Пример вставки картинки:

**uses** graphABC;

**var** pic:picture;

**begin**

pic:=picture.Create('Pictures\Mario.png');

Pic.draw(500,300)

**end**.



Рис.2. Использование вставки картинки в модуле GraphABC

**Модуль ABCObjects**

Модуль ABCObjects реализует векторные графические объекты с возможностью масштабирования, наложения друг на друга, создания составных графических объектов и многократного их вложения друг в друга, т.е. этот модуль больше подходит для рисования объекта вручную, а потом копируя его.

Пример рисования квадрата:

**uses** ABCObjects;

**begin**

SquareABC.Create(10, 10, 1000)

**end**.



Рис. 3. Использование простейшего оператора в модуле ABCObjects.

Я решил выбрать модуль GraphABC, потому что он больше подходит для создания игры. Мы можем просто вставить нужную картинку, а не рисовать уровень и героя «вручную».

# III. Описание создания игры «The Maze»

 Создание 2D игры в жанре лабиринт:

1. Скачать и установить бесплатную программу PascalABC.NET
2. Изучить модуль GraphABС
3. Создать папку, в которой будем хранить графические изображения объектов: уровень и герой.
4. Нарисовать уровень и главного героя в программе.

В модуле GraphABC делаем так: переменной location присваиваем значение картинки уровня, а переменной player значение картинки героя, и потом, с помощью оператора «.Draw» вывести изображение этой картинки на экран.

location := Picture.Create('data\f1.jpg');

player := Picture.Create('data\pers.png');

location.Draw(0, 0);

**if** left = 1 **then** player.Draw(x, y);

 **if** left = 0 **then** player.Draw(x, y);

где data – папка, где находятся картинки, f1 – уровень, а pers – главный герой.

5. “Научить” героя двигаться.

Уровень находится в плоскости с двумя координатами: x и у.

Я создал процедуру, которая перемещает героя после нажатие клавиш(w,a,s,d).

**procedure** KeyDown(Key: integer);

**begin**

 **if** Key = vk\_w **then** up := 1;

 **if** Key = vk\_a **then** left := 1;

 **if** Key = vk\_s **then** down := 1;

 **if** Key = vk\_d **then** right := 1;

**end**;

Этот фрагмент реализует перемещение героя.

До этого фрагмента вызывается процедура KeyDown, которая обеспечивает присваивание переменным right, left, down и up значение 1 при нажатии определённой кнопки

**if** right = 1 **then** x := x + v;

**if** left = 1 **then** x := x - v;

**if** up = 1 **then** y := y - v;

**if** down = 1 **then** y := y + v ;

V – это «скорость» героя. Я её поставил 4, т.е. за каждое нажатие одной из клавиш движения герой смещается на 4 пикселя в одну из сторон.

6. Сделать всем стенам «осязаемость», т. е. сделать так, чтобы герой не мог проходить через стены. Прописать: Если герой находится на координатах стены, то его перемещает туда, откуда он сделал этот шаг. Вот фрагмент из программы:

**if** (y=390) **and** (x>=200) **and** (x<=652) **and** (up=1) **then** y:=y+v;

**if** (y=382) **and** (x>=772) **and** (x<=1360) **and** (up=1) **then** y:=y+v;

**if** (y=634) **and** (x>=880) **and** (x<=1368) **and** (up=1) **then** y:=y+v;

**if** (y=622) **and** (x>=200) **and** (x<=524) **and** (up=1) **then** y:=y+v;

**if** (y=714) **and** (x>=528) **and** (x<=664) **and** (up=1) **then** y:=y+v;

**if** (y=714) **and** (x>=744) **and** (x<=880) **and** (up=1) **then** y:=y+v;

7. Создаём финиш.

Когда герой встаёт в центр лабиринта, ему показывается надпись «Победа!!!!»

**if** (x=700) **and** (y=430) **then** lvl1:=1;

После того, как переменная lvl1 становится равной 1, заканчивается цикл уровня, после которого прописано:

w.Draw(0,0)



Рис.6.Финиш

12. Создать “пасхалку”.

Когда я тестировал уровень, я обнаружил в своей игре баг (ошибка в программе): если подойти вплотную к двум стенам снаружи лабиринта, и одновременно идти в них, то герой пройдёт сквозь них. Я решил сделать так, что когда он проходит стену, то игра перемещает его в секретное место. «Пасхалку» в своей игре я решил сделать такую же, как и первую «пасхалку» в играх, т. е. имя автора этой игры.

“Пасхалка” (Пасхальное яйцо, Easter egg) - секрет в [компьютерной игре](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B8%D0%B3%D1%80%D0%B0), заложенный создателями. Отличие пасхального яйца в игре от обычного игрового секрета состоит в том, что его содержание, как правило, не вписывается в общую концепцию, выглядит в контексте неправдоподобно, нелепо, и зачастую является внешней ссылкой. Пасхальные яйца играют роль своеобразных бонусов для внимательных игроков или зрителей. Согласно [Уоррену Робинетту](https://en.wikipedia.org/wiki/Warren_Robinett) первое «пасхальное яйцо» было спрятано в [компьютерной игре](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B8%D0%B3%D1%80%D0%B0) «[Adventure](https://ru.wikipedia.org/wiki/Adventure_%281979%29%22%20%5Co%20%22Adventure%20%281979%29)». Игра выпускалась в [1979 году](https://ru.wikipedia.org/wiki/1979_%D0%B3%D0%BE%D0%B4_%D0%B2_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D1%85_%D0%B8%D0%B3%D1%80%D0%B0%D1%85) компанией [Atari](https://ru.wikipedia.org/wiki/Atari), и, поскольку в то время в Atari не было принято указывать в программах имена авторов, программист Уоррен Робинетт решил упоминание о себе спрятать внутри игры. Чтобы попасть в комнату с именем разработчика, следовало отыскать невидимую точку в одной из частей лабиринта и перенести её в другой конец уровня.

Фрагмент, программирующий «пасхалку»:

**if** (x=532) **and** (y=254) **then** EE:=1;

**if** EE=1 **then**

Easter\_egg.draw(0,0);

13. Протестировать всё.

Протестировав программу, я обнаружил некоторые проблемы, поэтому пришлось исправлять ошибки – осязаемость некоторых стен. Но, после того как я отладил это, тестирование игры прошло успешно.

Текст готовой программы:

**program** TheMaze;

**uses** GraphABC;

**var**

player, location, loc1,loc2,loc3,loc4,loc5,loc6,loc7, Easter\_egg, EE1, w, w2: picture;

 x, y, left, right, up, down,

 v, lvl1: integer;

 game: boolean;

**procedure** KeyDown(Key: integer);

**begin**

 **if** Key = vk\_w **then** up := 1;

 **if** Key = vk\_a **then** left := 1;

 **if** Key = vk\_s **then** down := 1;

 **if** Key = vk\_d **then** right := 1;

**end**;

**procedure** KeyUp(Key: integer);

**begin**

 **if** Key = vk\_w **then** up := 0;

 **if** Key = vk\_a **then** left := 0;

 **if** Key = vk\_s **then** down := 0;

 **if** Key = vk\_d **then** right := 0;

 **end**;

**var** f1 ,EE,r :integer;

**begin**

window.title := 'TheMaze';

window.Left := 100;

window.Top := 100;

SetWindowSize(1600, 872);

OnkeyDown := KeyDown;

OnkeyUp := KeyUp;

v := 4;

player := Picture.Create('data\pers.png');

location := Picture.Create('data\f1.jpg');

Easter\_egg:=Picture.Create('data\EE.jpg');

w:=picture.Create('data\Win.jpg');

w2:=picture.Create('data\win2.jpg');

game := true;

lvl1:=0;

x := 20; y := 10;

LockDrawing;

f1:=1;

**while** (game = true) **do**

**begin**

**while** lvl1=0 **do begin**

textout(1700, 900, x);

textout(1700,1000, y);

**if** EE=1 **then**

**begin**

Easter\_egg.draw(0,0);

**end**;

**if** f1=1 **then begin**

location.Draw(0, 0);

 **if** (x=0) **then** left:=0;

 **if** (x+84=1600) **then** right:=0;

 **if** (y=0) **then** up:=0;

 **if** ((y+100)=872) **then** down:=0;

 **if** left = 1 **then** player.Draw(x, y);

 **if** left = 0 **then** player.Draw(x, y);

 **if** right = 1 **then** x := x + v;

 **if** left = 1 **then** x := x - v;

 **if** up = 1 **then** y := y - v;

 **if** down = 1 **then** y := y + v ;

 **if** (x=12) **and** (left=1) **then** x:=x+v;

 **if** (y=6) **and** (up=1) **then** y:=y+v;

 **if** (x=1488) **and** (right=1) **then** x:=x-v;

**if** (y=762) **and** (down=1) **then** y:=y-v;

**if** (y=390) **and** (x>=200) **and** (x<=652) **and** (up=1) **then** y:=y+v;

**if** (y=382) **and** (x>=772) **and** (x<=1360) **and** (up=1) **then** y:=y+v;

**if** (y=634) **and** (x>=880) **and** (x<=1368) **and** (up=1) **then** y:=y+v;

**if** (y=622) **and** (x>=200) **and** (x<=524) **and** (up=1) **then** y:=y+v;

**if** (y=714) **and** (x>=528) **and** (x<=664) **and** (up=1) **then** y:=y+v;

**if** (y=714) **and** (x>=744) **and** (x<=880) **and** (up=1) **then** y:=y+v;

**if** (y=254) **and** (x>=200) **and** (x<=528) **and** (down=1) **then** y:=y-v;

**if** (y=486) **and** (x>=200) **and** (x<=656) **and** (down=1) **then** y:=y-v;

**if** (y=494) **and** (x>=760) **and** (x<=1372) **and** (down=1) **then** y:=y-v;

**if** (y=250) **and** (x>=904) **and** (x<=1360) **and** (down=1) **then** y:=y-v;

**if** (y=54) **and** (x>=552) **and** (x<=664) **and** (down=1) **then** y:=y-v;

**if** (y=54) **and** (x>=764) **and** (x<=900) **and** (down=1) **then** y:=y-v;

**if** (x=200) **and** (y>=254) **and** (y<=390) **and** (right=1) **then** x:=x-v;

**if** (x=200) **and** (y>=486) **and** (y<=622) **and** (right=1) **then** x:=x-v;

**if** (x=528) **and** (y>=622) **and** (y<=714) **and** (right=1) **then** x:=x-v;

**if** (x=744) **and** (y>=498) **and** (y<=714) **and** (right=1) **then** x:=x-v;

**if** (x=768) **and** (y>=50) **and** (y<=378) **and** (right=1) **then** x:=x-v;

**if** (x=532) **and** (y>=50) **and** (y<=250) **and** (right=1) **then** x:=x-v;

**if** (x=880) **and** (y>=638) **and** (y<=714) **and** (left=1) **then** x:=x+v;

**if** (x=664) **and** (y>=490) **and** (y<=714) **and** (left=1) **then** x:=x+v;

**if** (x=664) **and** (y>=54) **and** (y<=390) **and** (left=1) **then** x:=x+v;

**if** (x=900) **and** (y>=54) **and** (y<=250) **and** (left=1) **then** x:=x+v;

**if** (x=1360) **and** (y>=250) **and** (y<=382) **and** (left=1) **then** x:=x+v;

**if** (x=1368) **and** (y>=502) **and** (y<=634) **and** (left=1) **then** x:=x+v;

**if** (x=524) **and** (y>=324) **and** (y<=482) **and** (right=1) **then** x:=x-v;

**if** (y=250) **and** (x>=668) **and** (x<=764) **and** (down=1) **then** y:=y+v;

**if** (x=908) **and** (y>=386) **and** (y<=490) **and** (left=1) **then** x:=x+v;

**if** (x=532) **and** (y=254) **then** EE:=1;

**if** (x=700) **and** (y=430) **then** lvl1:=1;

**end**;

 redraw;

 **end**;

 sleep(100);

 r:=random(2);

**if** r=1 **then**

w.Draw(0,0)

**else**

w2.Draw(0,0);

 redraw;

 **end**

**end**.

# Заключение

Без сомнений, процесс создания своей собственной компьютерной игры – очень увлекательный и творческий процесс. Но, с другой стороны, очень сложный и долгий.

Я считаю, что я достиг цели, поставленной в начале работы. Я сумел сделать игру. Да, конечно она получилась не идеальной, но я считаю, что игра для первого раза получилась неплохой.

PascalABC.Net не очень подходит для создания игр. У этого языка очень малый функционал. Поэтому, для создания хороших игр, в будущем мне понадобиться изучить такой язык, как С++. Рассмотрев в своей работе языки программирования, я смог спрогнозировать своё развитие в изучении программирования. Следующим языком, который я изучу, будет Java, т.к. он более лёгок для изучения. После того, как я изучу Java, я смогу изучить C++ и создавать действительно хорошие игры.

Работать над созданием игр показалось мне очень интересным. Трудности, которые встречались мне, меня не останавливали. Надеюсь, в будущем я смогу создать игру, не уступающую другим по качеству.

# Источники информации

1. Научные опыты, доказывающие положительное влияние игр. URL: https:// dtf.ru/gamedev/5940-poleznye-igry-mirovye-issledovaniya-interaktivnyh-razvlecheniy
2. Классификация компьютерных игр. URL: https://ru.wikipedia.org/wiki/ Классификация\_компьютерных\_игр
3. Модуль GraphABC. URL: https://it.wikireading.ru/44679
4. Модуль ABCObjects. URL: https://wm-help.net/lib/b/book/1298783189/214
5. Первая компьютерная игра. URL: https://ru.wikipedia.org/wiki/ История\_компьютерных\_игр