Минобрнауки России

федеральное государственное бюджетное образовательное учреждение высшего образования

«Санкт-Петербургский государственный технологический институт

(технический университет)»

|  |  |  |  |
| --- | --- | --- | --- |
| **Направление подготовки** | | 09.03.01 Информатика и вычислительная техника | |
| **Направленность** | | Автоматизированные системы обработки информации и управления | |
| **Факультет** | | Информационных технологий и управления | |
| **Кафедра** | | Систем автоматизированного проектирования и управления | |
| **Курс** | 1 | **Группа** | 404 |

**КУРСОВОЙ ПРОЕКТ ПО ДИСЦИПЛИНЕ «ПРОГРАММИРОВАНИЕ»**

|  |  |
| --- | --- |
| **Тема:** | Разработка компьютерной игры-платформера |

|  |  |
| --- | --- |
| Выполнил обучающийся | К. А. Рухлова |
| Заведующий кафедрой, проф. | Т. Б. Чистякова |
| Руководитель, доц. | И. Г. Корниенко |
| Консультант, ст. преп. | А. К. Федин |

Санкт-Петербург

2021

Оглавление

[Введение 3](#_Toc74117800)

[1. Аналитический обзор 5](#_Toc74117801)

[1.1 Обзор и анализ компьютерных игр. Сравнительная характеристика существующих систем-аналогов. Обоснование актуальности проекта. 7](#_Toc74117802)

[1.2 Общая характеристика и особенности жанра платформера. 11](#_Toc74117803)

[1.3 Обзор и обоснование выбора инструментальных средств разработки компьютерной игры. 12](#_Toc74117804)

[2. Цели и задачи курсового проекта 14](#_Toc74117805)

[3. Технологическая часть 15](#_Toc74117806)

[3.1. Формализованное описание игрового процесса как объекта обработки информации 15](#_Toc74117807)

[3.2. Постановка задачи обработки информации 15](#_Toc74117808)

[3.3 Разработка функциональной структуры программного комплекса для реализации игрового процесса. 16](#_Toc74117809)

[3.4 Создание изображений для персонажей и объектов. 16](#_Toc74117810)

[3.5 Создание игровой физики и алгоритмов взаимодействия объектов между собой. 17](#_Toc74117811)

[3.6 Разработка структуры интерфейса для пользователя. 19](#_Toc74117812)

[3.7 Описание структур данных и алгоритмов (формат представления данных в памяти и на внешних носителя. 20](#_Toc74117813)

[3.8 Описание структуры программы (модули, основные функции, классы и т. д.). 20](#_Toc74117814)

[3.9 Тестирование приложения-игры путем пробного запуска. 20](#_Toc74117815)

[4. Характеристика программного и аппаратного обеспечения 22](#_Toc74117816)

[5. Вывод 25](#_Toc74117817)

[6. Список использованной литературы 26](#_Toc74117818)

# Введение

В наши дни существуют миллионы игр, отличающиеся жанром, графикой, количеством режимов, системными требованиями и т. д. Глядя на все это многообразие, сложно поверить, что первая компьютерная появилась не так давно. Одним из первых стал космический симулятор "Spacewar!", разработанный в 1961 году студентами MIT для компьютера PDP-1. По экрану монитора перемещались два космических корабля, задачей было с помощью джойстика уничтожить судно противника. Игра была рассчитана на двух человек. Уклоняться необходимо было не только от выстрелов соперника, но и от столкновений с его кораблем и звездами.

Именно из тех лет берет свое начало игровая индустрия. Поначалу компьютерные игры были чужды обществу, поэтому не были востребованными, однако с каждым годом их популярность росла. Причины этому очевидны: они становились все доступнее и доступнее, их постоянно совершенствовали, появлялись новые жанры, новая техника для игр и тому подобное. Так, например, в 90-е далеко не в каждом доме были компьютеры, поэтому люди ходили в компьютерные клубы, чтобы поиграть. Сейчас же существует возможность покупать игры и проходить их, не выходя их дома.

Существует ошибочное мнение, что компьютерные игры предназначены для детей. На самом деле, существует множество игр для каждой возрастной категории. Более того, по данным статистики средний возраст российского игрока — 33 года, и играют россияне в среднем около 10 часов в неделю.

Что касается меня, я играю не так часто, хотя и люблю это делать. Я пробовала много разных игр, но больше всего меня привлекают два жанра: стратегия (военная стратегия, если быть точным) и экшен. Жанр экшен объединяет в себе самых разных представителей: начиная от файтингов и простеньких шутеров, заканчивая всевозможными аркадами и платформерами.

Я думаю, что каждый геймер мечтал создать свою игру, в которой он может реализовать все свои идеи, самому отрисовать персонажа, написать сюжет, а возможно и придумать свой жанр, создать уникальный проект. Я только учусь программировать, поэтому мне показалось оптимальным вариантом выбрать популярный и несложный в реализации жанр, который в то же время нравится мне. Вот почему я решила создать игру жанра платформер.

# 1. Аналитический обзор

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

Классифицировать их можно по-разному. Сначала я рассмотрю игры по их жанрам. Итак, самыми популярными являются:

1. Экшен – в играх данного типа игрок, как правило, действуя в одиночку, должен уничтожать врагов при помощи холодного и огнестрельного оружия, для достижения определённых целей на данном уровне, обычно, после достижения заданных целей игрок, переходит на следующий уровень;

2. Симулятор – при помощи компьютера, как можно более полно, имитируется физическое поведение и управление каким-либо сложным объектом технической системы (например: боевым истребителем, автомобилем и т. д.);

3. Стратегия – игра, требующая выработки стратегии, например для победы в военной операции. Игрок управляет не одним персонажем, а целым подразделением, предприятием или даже вселенной;

4. Ролевая игра – игрок управляет одним или несколькими персонажами, каждый из которых описан набором численных характеристик, списком способностей и умений; примерами таких характеристик могут быть очки здоровья, показатели силы, ловкости, интеллекта, защиты, уклонения, уровень развития того или иного навыка и т. п.;

5. Приключения (квесты) – важнейшими элементами игры в жанре квеста являются собственно повествование и исследование мира, а ключевую роль в игровом процессе играет решение головоломок и задач, требующих от игрока умственных усилий. Такие характерные для других жанров компьютерных игр элементы, как бои, экономическое планирование и задачи, требующие от игрока скорости реакции и быстрых ответных действий, в квестах сведены к минимуму или вовсе отсутствуют;

6. Головоломка – целью является решение логических задач, требующих от игрока задействования логики, стратегии и интуиции. То есть для каждого найдется своя игра по вкусу.

Также можно рассмотреть их по количеству игроков:

1. Однопользовательская – игра либо одиночная, либо в качестве соперника/ напарника выступает компьютер;

2. Совместная игра на одном устройстве – игроки совершают свои ходы по очереди, используя один и тот же компьютер или консоль;

3. Многопользовательская – тип игры, в которой могут принимать участие сразу несколько игроков по локальной сети.

По визуальному представлению компьютерные игры можно разделить

на следующие виды:

1. текстовые – минимальное графическое представление, общение с

игроком проходит с помощью текста;

1. 2D – все элементы отрисованы в виде двумерной графики (спрайтов);
2. 3D – все элементы отрисованы в виде трехмерной графики (3D

модели).

По платформам:

1. Персональный компьютер (ПК, PC, ноутбук, нетбук);

2. Игровая консоль или приставка (PS, Xbox, Nintendo);

3. Мобильное устройство: телефон, планшет, карманный компьютер (КПК, PDA);

4. Игровой автомат;

5. Браузерная или флеш-игра.

Существует также классификация по расположению игровой камеры, по цели игры, по бюджету разработки, по издательскому формату и т. д.

Мой проект – это игра жанра платформер (поджанр экшена).

# 1.1 Обзор и анализ компьютерных игр. Сравнительная характеристика существующих систем-аналогов. Обоснование актуальности проекта.

Рассмотрим 3 игры жанра платформер.

**1. Super Mario World**

Сюжет:

Как и в предыдущих играх серии главный герой — водопроводчик Марио. Злодей король Купа похищает принцессу Тоадстул и заточает динозавра Йоши в яйцо. Марио освобождает Йоши и верхом на нём отправляется к замку короля Купы спасать принцессу.

Геймплей:

Игрок управляет персонажем (Марио или Луиджи) от третьего лица. Прохождение игры осуществляется с помощью навигации на двух экранах: карты мира и самого игрового уровня. На карте мира представлены примерные параметры текущей игровой локации и имеются несколько путей её прохождения, ведущих от начала карты к конечной цели — замку. Маршруты карты, по которым перемещается главный герой, соединены между собой подсвеченными значками: крепостей, т. н. домов призраков и т. д. Они позволяют игроку выбирать различные пути прохождения. Перемещаясь на значок той или иной локации, персонаж получает доступ на игровой уровень. Большая часть игры происходит на уровнях, которые наполнены препятствиями и врагами, а игрок волен проходить их различными способами. В конце каждого мира происходит битва с боссом — одним из Купалингов. В финале игры игрок должен победить главного антагониста — Боузера. Врагами на пути являются черепахи, броненосцы, динозавры и др.

Рисунок 1 и 2 – Экранные копии игры Super Mario World

**2. Disney's Hercules**

Сюжет:

Игра следует сюжету одноимённого мультфильма: Геркулес — сын Зевса, должен доказать, что он истинный герой, чтобы получить бессмертие и присоединиться к своей семье на горе Олимп. Для этого он должен совершить несколько подвигов и победить злодеев, в конце лицом к лицу встретившись со своим дядей, повелителем подземного мира, богом Аидом, ответственным за то, что после рождения Геркулес потерял своё бессмертие.

Геймплей:

Большая часть игры проходит в двух измерениях, хотя в некоторых миссиях игрок может подниматься по лестницам и наклонным плоскостям. Игра состоит из 10 уровней, имеет опции выбора сложности: начинающий, средний, «Геркулесовый», то есть сложный.

Последние два уровня работают только при втором и третьем вариантах. Кроме того, в игре присутствует три уровня, где у игрока нет возможности остановиться или нанести удар, при этом герой должен избегать столкновения с другими персонажами и объектами. Количество жизненной энергии показано в верхнем левом углу (на «скоростных уровнях» — в нижнем левом). Объём возможной энергии можно пополнять, подбирая фигурки Геркулеса, сама энергия пополняется за счёт найденных кружек с энергетическим напитком.

На большинстве уровней спрятаны послания с буквами: если собрать их все, то получится надпись «HERCULES» — благодаря ей игрок сможет начать игру со следующего уровня, а не с начала всей игры. Также в игре спрятаны 4 вазы — они дают игроку секретный пароль к данному уровню, чтобы начать игру с места, где была найдена ваза.

Рисунок 3 и 4 – Экранные копии игры Hercules

**3. Neon Abyss**

Сюжет:

Нет.

Геймплей:

Neon Abyss – это бодрый shoot 'em up. Игрок исследует случайно генерируемое подземелье, переходя из одной комнаты в другую. Убивает всех врагов в помещении – движется дальше. Есть тут и комнаты с сокровищами, и тайные комнаты, и даже комнаты с мини-играми. Не во все из них так просто попасть: где-то нужно применить гранату, где-то потратить кристаллы, где-то – монеты, ну и, конечно, ключи.

Сначала бездна состоит из пяти этажей, но потом, после убийства босса на самом глубоком уровне, их количество увеличивается. После того как вы одержите победу над главным противником, придётся начинать забег сначала и по новой убивать предыдущих богов-менеджеров, прежде чем увидеть нового. Помимо этого, на каждом этаже есть и собственный босс – пока его не одолеешь, вниз не пустят.

За убийство боссов дают самоцветы, которые нужно тратить на развитие бездны. В Neon Abyss есть целое многоуровневое древо улучшений, в котором можно покупать новые правила (например, если разрушишь облитые краской камни, то получишь награду), добавлять в игру особые предметы и целые комнаты и открывать новых персонажей.

Рисунок 5 и 6 – Экранные копии игры Neon Abyss

Таблица 1 – Сравнение существующих систем-аналогов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование критерия сравнения** | **Super Mario World** | **Disney's Hercules** | **Neon Abyss** | **Мой проект** |
| Год создания | 1990 | 1997 | 2020 | 2021 |
| Платформы | SNES, GBA, N3DS | PS, PC, GB | NSW, PC, PS4, X-one | PC, web |
| Продолжение таблицы 1 | | | | |
| **Наименование критерия сравнения** | **Super Mario World** | **Disney's Hercules** | **Neon Abyss** | **Мой проект** |
| Поддержка многопользовательской игры | + | - | - | - |
| Поддержка нескольких языков | + | + | + | - |
| Сюжет | + | + | - | + |
| Наличие музыкального сопровождения | + | + | + | + |
| Наличие врагов | + | + | + | - |
| Наличие тайных комнат/ уровней | + | + | + | - |

На данный момент мой проект пока не обладает никакими преимуществами над аналогами, поэтому сейчас он не конкурентоспособен. Однако, это моя первая игра, а потому для меня важнее сам процесс разработки и полученный опыт, нежели итоговый результат.

# 1.2 Общая характеристика и особенности жанра платформера.

Платформер — жанр компьютерных игр, где главным элементом является перемещение по платформам, обычно посредством прыжков. Платформы могут быть совершенно разными: от простой дорожки до выступов скал. От игрока требуется ловкость в управлении прыжками и перемещением персонажа. Целью может быть добраться до конца уровня, собрать все предметы, найти секрет или убежать от опасности невредимым.

Многие игры подобного жанра характеризуются нереалистичностью, рисованной мультяшной графикой. Персонажами таких игр часто бывают вымышленные существа или антропоморфные животные.

До распространения трёхмерной графики платформеры были одним из основных жанров и до сих пор остаются среди самых используемых в независимых видеоиграх, хотя их популярность среди ААА-игр существенно упала.

# 1.3 Обзор и обоснование выбора инструментальных средств разработки компьютерной игры.

Написать игру можно по-разному: например, написать ее в среде разработки на чистом языке (C++, Java). Но такие игры не будут многофункциональными, а написание даже простой игры занимает много усилий.

Также для создания игр можно использовать специальные игровые движки, которые значительно упрощают работу программиста. Наиболее популярные среди них являются: *Unity, Unreal Engine 4, GameMaker: Studio, CryEngine 3*. Я взяла для разработки не самую известную программу – *Cocos Creator*, однако, среда разработки интуитивно понятная и простая, что для меня, как для новичка, очень важно. Доступны два языка для написания скриптов: либо Javascript, либо Typescript. Рассмотрим преимущества каждого из языков.

JavaScript – это в основном язык сценариев, который следует правилам программирования на стороне клиента. Он работает в веб-браузере без необходимости веб-сервера или чего-то еще. Это позволяет нашему коду взаимодействовать с браузером и даже может изменять или обновлять как HTML, так и CSS. Хотя вы можете использовать JavaScript с REST API, XML и т.д. Он не предназначен для разработки приложений, апплетов или больших сложных систем.

Преимущества JavaScript:

• JavaScript – это гибкий и мощный язык с открытым исходным кодом, поддерживаемый такими легендами, как Microsoft и Mozilla.

• Он может быть использован как на стороне клиента, так и на стороне сервера.

• Он поддерживает классы, интерфейсы и модули.

• Он специально разработан для небольших сценариев.

• Это кроссплатформенный язык.

• Он обеспечивает надежный рабочий процесс тестирования и поддержку зависимостей.

Typescript – это своего рода обновленная версия языка Javascript. Он может работать на Node.js или любом веб-браузере, который поддерживает ECMAScript 3 или выше. TypeScript – это статически скомпилированный язык, который обеспечивает необязательную статическую типизацию, классы и интерфейс.

Преимущества TypeScript:

• TypeScript обеспечивает простую навигацию по коду, предотвращение ошибок и поддержку кода.

• Он поддерживает статическую типизацию.

• Он также поддерживает интерфейсы, подчиненные интерфейсы, классы, подклассы и ECMAScript 6 или ES6.

• Он поставляется с возможностями объектно-ориентированного программирования с наследованием закрытых членов и интерфейсов.

• Много доступных IDE.

Два языка очень похожи, но остановилась я на Java Script. Так как JS проще в обучении, и он намного известнее TS, а следовательно, и общество разработчиков у него больше. Тем более, это классический язык, который, на мой взгляд, хорош в знакомстве со скриптами.

# 2. Цели и задачи курсового проекта

Целью курсового проекта является разработка собственной компьютерной игры.

Основные задачи курсового проекта:

1) Разработка структуры интерфейса для пользователя;

2) Разработка функциональной структуры программного комплекса для реализации игрового процесса;

3) Создание игровой физики и алгоритмов взаимодействия объектов между собой;

4) Описание структуры программы (модули, основные функции, классы и т. д.);

5) Тестирование приложения-игры путем пробного запуска;

6) Оформление документации (пояснительной записки, презентации) по проекту.

# 3. Технологическая часть

# 3.1. Формализованное описание игрового процесса как объекта обработки информации



X{K} Y{M}

F{Q}

Рисунок 7 – Формализованное описание игрового процесса как объекта обработки информации

Где:

X – входные данные;

K – клавиша, клик мышки;

F – управляемые параметры;

Q – обработка нажатия;

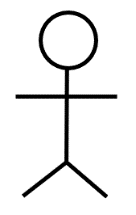
Y – выходные данные;

M – движение персонажа.

# 3.2. Постановка задачи обработки информации

Необходимо разработать компьютерную игру, играя в которую, человек может развлечься и приятно провести время. Сюжет игры должен быть обучающим.

# 3.3 Разработка функциональной структуры программного комплекса для реализации игрового процесса.

Для более простого представления программного комплекса, была разработана его функциональная структура, в которой отражены основные взаимодействия с программой.

ГПИ

Клик мышью

Отображение объекта

Нажатие на клавишу

Пользователь

K

Обработка нажатия

M

Q

Движение персонажа

Рисунок 8 – Функциональная структура программного комплекса для реализации игрового процесса

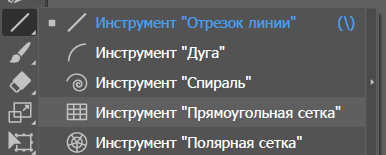
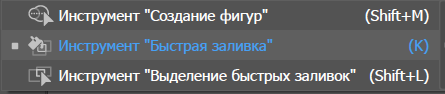
# 3.4 Создание изображений для персонажей и объектов.

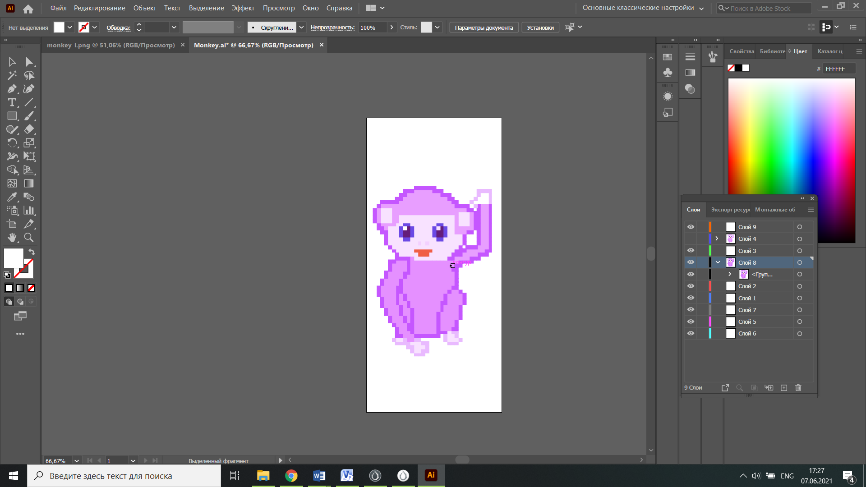
Для создания спрайтов было принято решение использовать программу Adobe Illustrator. Adobe Illustrator — векторный графический редактор, разработанный и распространяемый компанией Adobe Systems.

Я выбрала ее, так как она очень многофункциональная и удобная для создания всяческих иллюстраций и картинок. Картинки в Illustrator состоят из отдельных цветных фигур. Можно редактировать контур этих фигур, добавлять разные заливки или обводки. Чтобы получилось полноценное изображение с несколькими цветами, приходится собирать несколько фигур вместе. В Illustrator они накладываются одна на другую, как в аппликации, и можно менять их порядок по ходу работы.

Помимо однотонных заливок, в программе доступны градиенты, разные уровни прозрачности, возможность добавлять эффекты. Также в Illustrator очень удобно работать с текстом: легко перемещать надписи, редактировать параметры шрифта, а при желании буквы можно перевести в кривые и работать с ними как с обычными векторными фигурами.

Также здесь удобно делать пиксельные изображения. Для этого необходимо выбрать инструмент «Прямоугольная сетка» и задать ее параметры. После с помощью инструмента «Быстрая заливка» можно по пикселям нарисовать изображение.



Рисунки 9 - 10 – Инструменты для рисования пикселей

Рисунки 11– Экранная копия приложения Adobe Illustrator

# 3.5 Создание игровой физики и алгоритмов взаимодействия объектов между собой.

Библиотека класса CC от Cocos Creator очень обширна – в ней все есть, что надо, чтобы написать игру.

Так, например, для задания движения персонажа мы воспользуемся функцией класса Rigid Body (это класс, который составляет игровую физику) – *applyForceToCenter* (приложить силу к центру масс), аргументом которой является приложенная сила. Сила вычисляется как произведение направления (оно может быть 1, если движение влево, и -1, если движение вправо) и скорости (она задается как константа). А перемещается объект с помощью конструктора класса *Vec2 – v2,* аргументами которой являются две координаты: по x, по y. В данном случае координата по x – это и есть сила, а по y – 0.

Точно такие же операции проводятся для создания прыжка объекта. Разница в определении силы и перемещении объекта. Сила равна константе силы прыжка. А перемещение осуществляется по y, поэтому по x – 0, а по y – сила.

Для вычисления столкновения с объектом я использую класс Collider, его функцию *onBeginContact*, аргументами на вход которой являются контакт, свойство пересечения объекта данного, свойство пересечения объекта другого. Далее, можно задать определенный тег для объекта и по тегу определять пересечения с другими объектами.

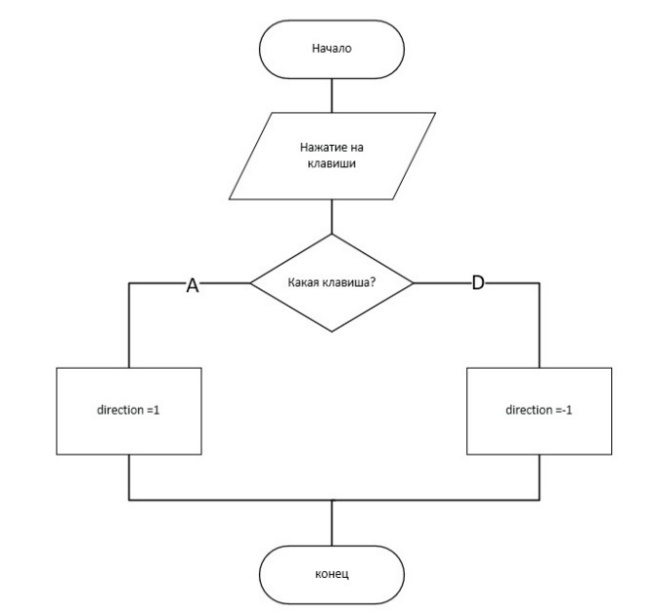


Рисунок 12 – Блок-схема движения обезьянки

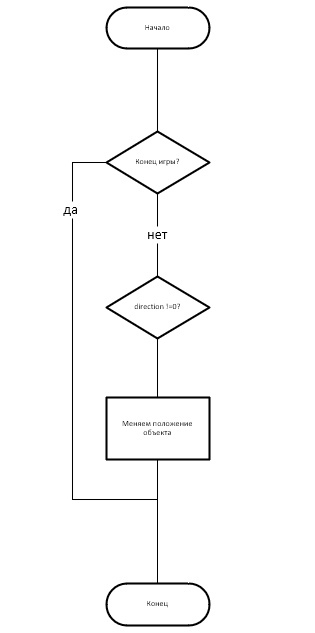


Рисунок 13 – Блок-схема функции update()

# 3.6 Разработка структуры интерфейса для пользователя.

При входе в игру пользователь видит сюжет игры в картинках. После истории, пользователь переходит в меню, где есть три кнопки: начало игры, настройки и выход.

Нажав на кнопку «Start the game» (Начать игру), открывается дополнительное окно, где предлагается загрузить игру, либо же создать новую. После создания, в окне загрузки игр появится имя, которое пользователь только что ввел. При нажатии на кнопку имени, запускается игра.

Нажав на кнопку «Settings» (Настройки), открывается дополнительное окно, в котором пользователь может остановить песню.

Нажав на кнопку «Exit» (Выход), открывается дополнительное окно, в котором пользователя переспрашивают, хочет ли он выйти из игры. При нажатии на кнопку «Yes», закрывается окно программы. При нажатии на кнопку «No», пользователь возвращается обратно в меню.

Во всех дополнительных окнах есть кнопка крестик, позволяющая закрыть это окно.

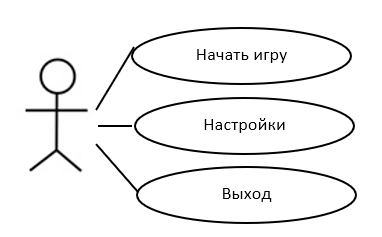


Рисунок 14 - UML-диаграмма вариантов использования пользовательского интерфейса в меню игры

# 3.7 Описание структур данных и алгоритмов (формат представления данных в памяти и на внешних носителя.

Для того, чтобы запустить программу необходимо перейти по ссылке в браузере (открыть ярлык программы).

# 3.8 Описание структуры программы (модули, основные функции, классы и т. д.).

# 3.9 Тестирование приложения-игры путем пробного запуска.

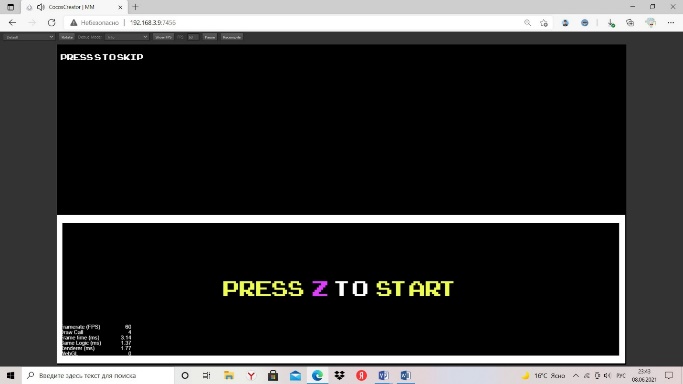


Рисунок 15 - Запуск

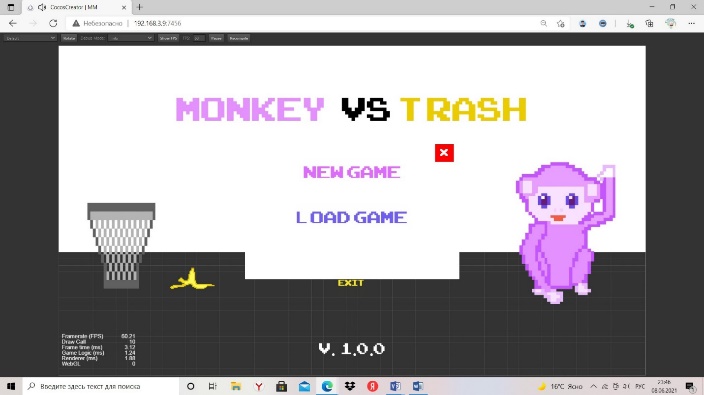


Рисунок 16 – Меню при нажатии на “Start”

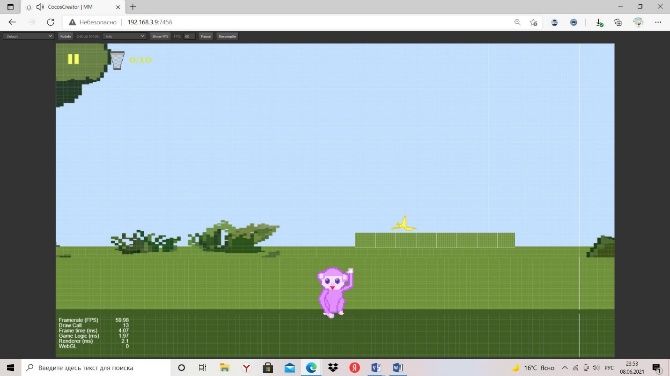


Рисунок 17 – Процесс игры

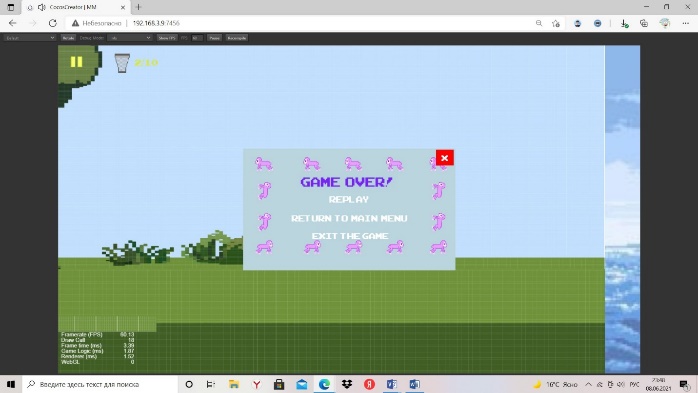
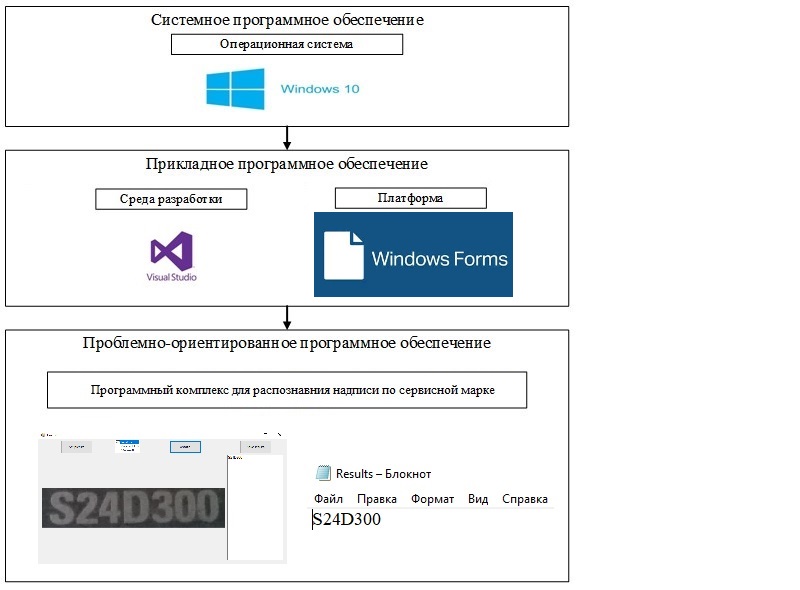


Рисунок 18 – Конец игры

# 4. Характеристика программного и аппаратного обеспечения

На рисунке 12 представлена структура программного обеспечения. Компьютерная игра разработана под управлением операционной системы Windows 10 на платформе .NET Framework. Средой разработки является Cocos Creator 2.4.4.





Компьютерная игра

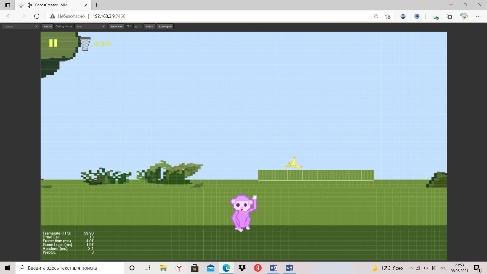


Таблица 2 – Характеристика программного обеспечения

|  |  |
| --- | --- |
| Показатель | Значение |
| Среда разработки | Cocos Creator 2.4.4 |
| Технология программирования | Объектно-ориентированное программирование |
| Язык программирования | Javascript |
| Количество входных переменных |  |
| Количество внутренних переменных |  |
| Количество выходных переменных |  |
| Количество классов, структур |  |
| Количество функций |  |
| Размер исполняемого файла, МБ |  |
| Время обработки данных и визуализация результатов, мс |  |

Требования к ЭВМ, необходимой для нормального функционирования дистанционной системы представлены в таблице 3.

Таблица 3 – Минимальные системные требования

|  |  |
| --- | --- |
| Показатель | Значение |
| Тип ЭВМ | Персональный компьютер |
| Тактовая частота процессора, ГГц | 3.2 |
| Объём оперативной памяти, ГБ | 8 |
| Объём внешней памяти, ГБ | 1000 |
| Продолжение таблицы 3 – Минимальные системные требования | |
| Состав и характеристика периферийных устройств ЭВМ | |
| Показатель | Значение |
| Состав и характеристика системного оборудования | Сетевой адаптер |
| Операционная система | Windows 10 |
| Прикладное программное обеспечение, необходимое для функционирования программного комплекса | .NET Framework 4.6.1 |

# 5. Вывод

# 6. Список использованной литературы

1. Флэнаган, Д. JavaScript. Подробное руководство/ Пер. с англ. яз. под общей ред. А. Галунов – Санкт-Петербург: Символ-Плюс, 2008. – 992 с. - ISBN 978-5-8459-2036-2

2. Фаулер, М. Рефакторинг кода на JavaScript/ Пер. с англ. яз. под общей ред. А. Галунов – Санкт-Петербург: Символ-Плюс, 2019. - 432 с. - ISBN 978-5-907144-59-0

3. Michael McMillan Data Structures and Algorithms with JavaScript – США: O’Reilly Media, 2017. - 246 с. - ISBN 978-1-4493-6493-9

4. Минник, К. JavaScript для чайников/ Пер. с англ. яз. под общей ред. С. Н. Тригуб – Москва: И. Д. Вильямс, 2013. - 320 с. - ISBN 978-5-8459-2036-2

5. Шелл, Д. Геймдизайн. Как создать игру, в которую будут играть все/ Пер. с англ. яз. под общей ред. С. Турко – Альпина Паблишер, 2019. - 782 с.

- ISBN 978-5-9614-1209-36.

6. Крокфорд, Д. JavaScript. Сильные стороны/ Пер. с англ. яз. под общей ред. Ю. Сергиенко – Москва: И. Д. Вильямс, 2019. - 304 с.

- ISBN 978-5-4461-1260-9

7.Хавербеке, М. [​](https://telegra.ph/file/f3c36bebeb973042741f8.jpg)Выразительный JavaScript. Современное веб программирование. 3-е издание/ Пер. с англ. яз. под общей ред. Е. Сандицкая –Спб: Питер, 2020. - 480 с.

- ISBN 978-5-4461-1226-5

8. Херман, Д. Сила JavaScript. 68 способов эффективного использования JS. — СПб.: Питер, 2013. — 288 с. - ISBN 978-5-496-00524-1

9.Миковски, С., Пауэлл, К. Разработка одностраничных веб-приложений/ Пер. с англ. яз. под общей ред. Н. Слинкин — Москва, ДМК Пресс, 2018. — 512 с. - ISBN 978-5-97060-580-6

10. Бибо, Б., Марас, И., Резиг, Д. Секреты JavaScript ниндзя –Москва: И. Д. Вильямс, 2018. - 544 с. - ISBN 978-5-9908911-8-0

