

Минобрнауки России

федеральное государственное бюджетное образовательное учреждение высшего образования

«Санкт–Петербургский государственный технологический институт

(технический университет)»

КУРСОВОЙ ПРОЕКТ

**Направление подготовки**

09.03.03 Прикладная информатика в химии

Прикладная информатика в химии

Информационных технологий и управления

Систем автоматизированного проектирования и управления

Программирование

**Группа** 405

Шахов Михаил Андреевич

**Направленность**

**Факультет**

**Кафедра**

**Учебная дисциплина**

**Курс** 1

**Студент**

|  |  |
| --- | --- |
| **Тема:** | Разработка мобильного игрового приложения «Rolly Bolly» в жанре 2D платформер |

Заведующая кафедрой, проф. Т. Б. Чистякова

Лектор, доц. И. Г. Корниенко

Консультант, ст. преп. А. К. Федин

Исполнитель М. А. Шахов

Санкт–Петербург

2021

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 3](#_Toc85142437)

[**1 АНАЛИТИЧЕСКИЙ ОБЗОР** 4](#_Toc85142438)

[**1.1 Обзор и анализ сферы разработки мобильных игр. Сравнительная характеристика существующих мобильных игры в жанре кликер. Обоснование актуальности проекта** 4](#_Toc85142439)

[1.1.1 Сравнительная характеристика существующих мобильных игр в жанре 2D платформер 4](#_Toc85142440)

[1.1.2 Актуальности проекта 7](#_Toc85142441)

[**1.2** **Общая характеристика и особенности реализации мобильных игры в жанре 2D платформер** 7](#_Toc85142442)

[**1.3 Обзор и обоснование выбора инструментальных средств разработки мобильного игрового приложения «Rolly Bolly»** 8](#_Toc85142443)

[**2 ЦЕЛЬ И ЗАДАЧИ КУРСОВОГО ПРОЕКТА** 16](#_Toc85142444)

[**3 ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ** 17](#_Toc85142445)

[**3.1 Формализованное описание игрового процесса как объекта обработки информации.** 17](#_Toc85142446)

[**3.2 Постановка задачи обработки информации** 18](#_Toc85142447)

[**3.3 Разработка функциональной структуры мобильного игрового приложения «Rolly Bolly»**](#_Toc85142448)

[**3.4 Разработка блок-схем игровых механик** 19](#_Toc85142449)

[**3.5 Разработка структуры интерфейса для пользователя программного продукта** 21](#_Toc85142450)

[**3.6 Описание структуры программы (модули, основные функции, классы и т. д.)** 21](#_Toc85142451)

[**3.7 Тестирование игры** 23](#_Toc85142452)

[**ХАРАКТЕРИСТИКА ПРОГРАММНОГО И АППАРАТНОГО ОБЕСПЕЧЕНИЯ** 25](#_Toc85142454)

[**ВЫВОДЫ ПО КУРСОВОМУ ПРОЕКТУ** 27](#_Toc85142455)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 28](#_Toc85142456)

# **ВВЕДЕНИЕ**

Игры и, в частности, мобильные игры, продолжают стремительно расти в индустрии развлечений. Все больше людей играют в игры, чем когда-либо прежде, поскольку это является способом социального взаимодействия и возможностью провести свободное время, а Covid-19 только ускоряет процесс развития игровой индустрии.

По оценкам Newzoo, к концу 2020 года доход от мобильных игр достиг 76,7 миллиарда долларов, что на 12% больше, чем в 2019 году. Для сравнения: глобальный доход от кассовых сборов составил 42,5 миллиарда долларов. В 2019 году мировая музыкальная индустрия принесла оптовую выручку в размере 20,2 миллиарда долларов. Другими словами, к концу 2020 года одни только мобильные игры принесли больше годового дохода, чем музыкальная и киноиндустрия вместе взятые.

**Статистика индустрии мобильных игр**

* По данным GameAnalytics, в первом квартале 2020 года число мобильных геймеров выросло с 1,2 миллиарда человек до 1,75 миллиарда в месяц, что означает рост на 46%.
* Newzoo заявляет, что число мобильных геймеров в 2020 году достигло пика и составит 2,6 миллиарда, из которых 38% платят за игры.
* Потребители потратили более 19 миллиардов долларов на мобильные игры во втором квартале 2020 года, что представляет собой самый крупный квартал, по данным App Annie, 21% загруженных приложений для Android и 25% приложений для iOS — это игры. Выручка Google Play продолжает расти быстрее, чем выручка iOS App Store: с 19,6% до 13.2%
* Прогнозируется, что к 2022 году игры для Android будут приносить 56% всех доходов от мобильных игр [1].

**Игра «Rolly Bolly»**

Кроссплатформенное приложение «Rolly Bolly» — это приложение игровой направленности.

Данное приложение представляет собой видеоигру, задачей которой является прохождение уровней, путем преодоления разнообразных сложностей и препятствий на своем на пути. Область применения данного приложения находится в сфере развлечений. Проект является 2D платформером.

# **1 АНАЛИТИЧЕСКИЙ ОБЗОР**

# **1.1 Обзор и анализ сферы разработки мобильных игр. Сравнительная характеристика существующих мобильных игры в жанре кликер. Обоснование актуальности проекта**

1.1.1 Сравнительная характеристика существующих мобильных игр в жанре 2D платформер

Общие черты платформеров проявляются в прыгании по платформам, преодолении различных препятствий. Противники (называемые «врагами»), всегда многочисленны и разнородны, обладают примитивным искусственным интеллектом, либо не обладают им вовсе, перемещаясь по круговой дистанции или совершая повторяющиеся действия. Соприкосновение с противником обычно отнимает жизненные силы у героя или вовсе убивает его. Графическое исполнение игр подобного жанра характеризуются нереалистичностью, рисованной мультяшной графикой.

Разберем три представителя мобильных игр в жанре 2D платформер:

* Super Mario Run,
* Red Ball,
* Bounce.

Разберем ряд плюсов и минусов каждой игры, выделим особенности.

**Super Mario Run**

В Super Mario Run элементарно приятно играть. Всё строится на тактильных ощущениях и зрительном восприятии. Прыгнувшая на ваш счёт монетка рассеивается с каким-то сверхудовлетворяющим звуком, рассыпая вокруг кольца из золота. Выбитый из блока бонус делает вас в два раза выше, и эффект тут не столько функциональный, сколько зрительный. Nintendo, как никто другой, умеет делать из своих игр даже не шоу, а праздник тепла, добродушия и домашнего очарования.

Проблемы у игры тоже есть, даже помимо цены. Super Mario Run стоит 749 рублей, а бесплатно доступны только несколько вступительных уровней. На внутриигровые покупки можно с легкостью, незаметно для себя, потратить больше денег, чем стоит сама игра. Еще одна проблема Super Mario Run заключается в том, что она является довольно обычной игрой в своем жанре, в частности в поджанре «раннер».



Рисунок 1 — Скриншот игры «Super Mario Run»

**Red Ball**

Центральный персонаж — маленький мячик с забавной рожицей. Он обитает в сказочном мире, состоящем из дорог, дорожек, тропинок, коридоров, проспектов и прочих замысловатых конструкций, похожих на один гигантский лабиринт. Задача главного героя — собрать все звезды и золотые ключи, рассыпанные по его просторам. Но есть сила, мешающая малышу-мячику это сделать.

Red Ball отличается своими головоломками, что смещает жанр игры в сторону одноименного жанра «головоломка»

Если у кошки девять жизней, то у шарика, который постоянно тонет, падает в пропасть и попадает в лапы злобным кубикам, этих жизней должна быть хотя бы добрая сотня, но их очень мало, и они быстро заканчиваются. Жизни восполняются, но гораздо медленнее, чем шарик будет умирать. Из этого вытекает один из главных минусов игры. Игра как бы подталкивает игрока купить премиум аккаунт, который позволяет умирать и воскресать безлимитно.



Рисунок 2 **—** Скриншот игры «Red Ball»

**Bounce**

Игра является классическим примером мобильного приложения в жанре 2D платформер. Bounce имеет предельно простые правила, в своё время стала настоящим испытанием для нервной системы многих игроков. Казалось бы, что может быть сложного: катаешь шарик, проводишь его через все кольца и идёшь к финишу, но с каждым последующим уровнем Bounce становится всё сложнее и сложнее, и уже на 6-7 уровне она превращается в настоящее мучение. Нужно очень быстро реагировать, чтобы попасть в кольцо и не нарваться на вездесущие шипы.

Недостатки игры связаны прежде всего с ее возрастом: устаревшая графика, несовременный дизайн уровней, наипростейшая физика, негативно сказывающаяся на отзывчивости управления, что, в свою очередь, проявляется в ощущении «вязкости» игрового процесса.



Рисунок 3 — Игровой процесс игры «Bounce»

Сравнение трех популярных игр в жанре 2D платформер помогает в создании своего конечного продукта. Заимствование лучших аспектов этих игр, минимизация минусов **—** вот ключ к успеху. У «Rolly Bolly» интуитивное управление, приятная физика передвижения персонажа, легкая графика. Сложность игры усреднена, с уклоном в сторону «Bounce».

1.1.2 Актуальности проекта

Актуальность проекта заключается в том, что на данный момент цены на видеокарты существенно превысили ожидание покупателя. Из-за того, что сейчас многие люди стали покупать видеокарты для майнинга криптовалюты, купить видеокарту по заявленной производителем цене становится невозможным. Так как спрос превышает предложение, в ход событий включаются перекупщики, которые лишь усложняют жизнь обычным пользователям.

Таким образом все больше людей начинают использовать свои смартфоны в качестве игрового устройства. Рынок мобильных игр растёт, количество пользователей смартфонов увеличивается с каждом днём, и эта тенденция только набирает обороты. Поэтому создание мобильных игр является актуальной задачей.

# **Общая характеристика и особенности реализации мобильных игры в жанре 2D платформер**

Платформер — жанр компьютерных игр, в которых основу игрового процесса составляют прыжки по платформам, лазанье по лестницам, сбор предметов, необходимых для победы над врагами или завершения уровня.

Многие игры подобного жанра характеризуются нереалистичностью, рисованной мультяшной графикой. Персонажами таких игр часто бывают вымышленные существа (к примеру, драконы, гоблины) или антропоморфные животные.

Некоторые предметы, называемые пауэр-апами, наделяют управляемого игроком персонажа особой силой, которая обычно иссякает со временем (к примеру: силовое поле, ускорение, увеличение высоты прыжков). Коллекционные предметы, оружие и «пауэер-ап» собираются обычно простым прикосновением персонажа и для применения не требуют специальных действий со стороны игрока. Реже предметы собираются в «инвентарь» героя и применяются специальной командой (такое поведение более характерно для аркадных головоломок). Сходный жанр компьютерных игр сайд-скроллер.

Противники (называемые «врагами»), всегда многочисленные и разнородные, обладают примитивным искусственным интеллектом, стремясь максимально приблизиться к игроку, либо не обладают им вовсе, перемещаясь по круговой дистанции или совершая повторяющиеся действия. Соприкосновение с противником обычно отнимает жизненные силы у героя или вовсе убивает его. Иногда противник может быть нейтрализован либо прыжком ему на голову, либо из оружия, если им обладает герой. Смерть живых существ обычно изображается упрощённо или символически (существо исчезает или проваливается вниз за пределы экрана).

Уровни, как правило, изобилуют секретами (скрытые проходы в стенах, высокие или труднодоступные места), нахождение которых существенно облегчает прохождение и подогревает интерес игрока.

# **1.3 Обзор и обоснование выбора инструментальных средств разработки мобильного игрового приложения «Rolly Bolly»**

Для разработки необходимо выбрать движок. Рассмотрим несколько популярных представителей.

**Unity**

Unity — это один из самых популярных движков на данный момент. Составить ему конкуренцию может разве что Unreal Engine 4, но что у первого, что и у второго есть свои плюсы и свои минусы. Unity поддерживает почти 30 платформ, в их числе мобильные, виртуальная реальность, настольные компьютеры, консоли и т. д. Unity — это не просто хороший вариант для старта, это идеальный вариант для старта! Тут заложено использование гибкой модульной системы при создании сцен и персонажей в игре. Даже новичок способен создать хороший проект, используя готовые спрайты и конструктор движка.

Если же считаете, что готовы погрузиться с головой в «кирпичную систему» для продвинутых, то тут есть возможность настройки практически любого компонента. К примеру, разработка своего собственного шейдера или же переписать сетевой стек для потребностей своей игры.

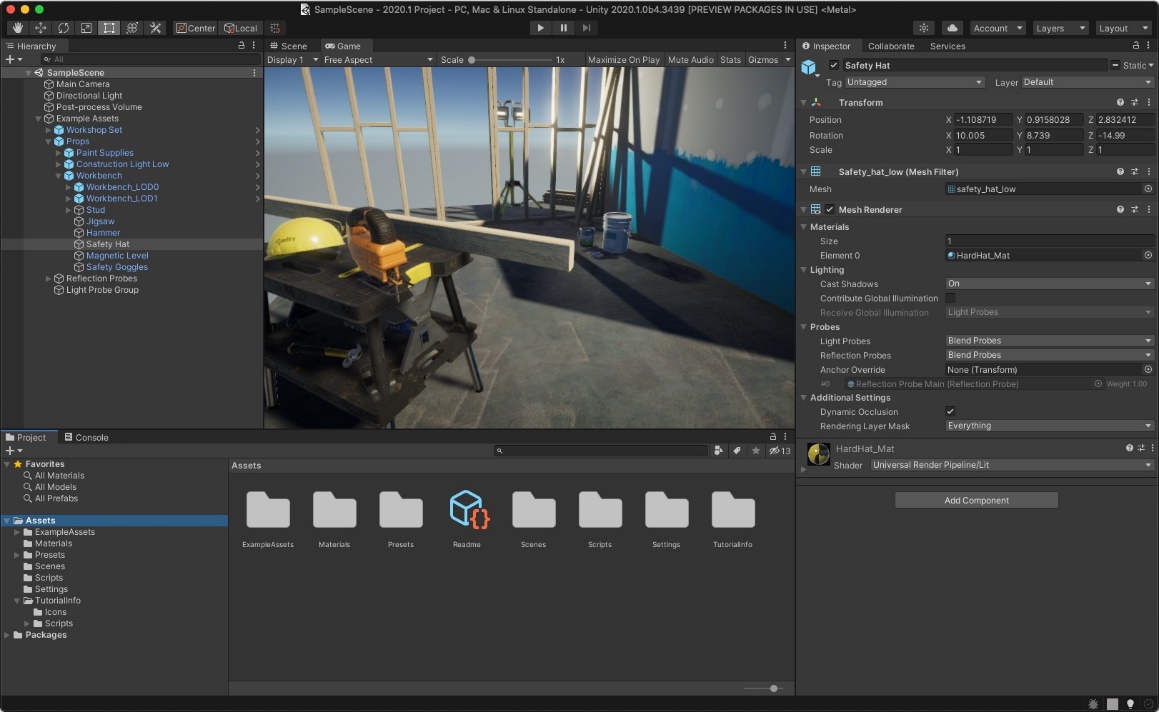


Рисунок 4 — Скриншот интерфейса Unity

**Unreal Engine**

Unreal Engine 4 — это набор инструментов для разработки игр, имеющий широкие возможности: от создания двухмерных игр на мобильные до AAA-проектов для консолей. Этот движок использовался при разработке таких игр, как ARK: Survival Evolved, Tekken и Kingdom Hearts III.

Разработка в Unreal Engine 4 очень проста для начинающих. С помощью системы визуального создания скриптов Blueprints Visual Scripting можно создавать готовые игры, не написав ни строчки кода! В сочетании с удобным интерфейсом это позволяет быстро изготавливать рабочие прототипы.



Рисунок 5 — Скриншот интерфейса Unreal Engine

**CryEngine**

CryEngine — игровой движок, созданный немецкой компанией Crytek в 2002 году и первоначально используемый в шутере от первого лица Far Cry. На CryEngine разных версий сделано много отличных игр от разных игровых студий, которые лицензировали движок: Far Cry, Crysis, Entropia Universe, Blue Mars, Warface, Homefront: The Revolution, Sniper: Ghost Warrior, Armored Warfare, Evolve и многие другие. В марте 2016 года компания Crytek анонсировала выход своего нового движка CryEngine V и вскоре опубликовала исходный код на Github. Игровой движок написан на языке C++ и сразу привлёк внимание как сообщества open-source разработчиков.

Изображение выглядит как текст, снимок экрана, электроника, компьютер

Автоматически созданное описание

Рисунок 6 — Скриншот интерфейса CryEngine

Таблица 1 — Сравнения движков для разработки игр

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Движки | Русификация | Уроки на русском языке | Кроссплатформенность (iOS, Windows, android) | Наличие разнообразных плагинов, конструкторов | Простота разработки 2D игры |
| Unity | + | Много | + | много | просто |
| Unreal Engine | + | Мало | - | мало | средне |
| CryEngine | + | Мало | - | мало | сложно |

Для реализации данного проекта мною был выбран игровой движок Unity версии 5.5.,

Unity имеет намного больше (в 3.5-4 раза) уроков на русском языке, нежели Unreal Engine или CryEngine. В связи с его популярностью он имеет существенно объемное количество разнообразных плагинов, что облегчает разработку 2D игр.

Достоинства данного движка:

* Удобный интерфейс, содержащий все необходимые элементы;
* Огромное количество поддерживаемых платформ, что позволит создавать игры не только для компьютера, но и для смартфона, игровой консоли и ряда других устройств;
* Широкие возможности работы со скриптами. Unity взаимодействует с двумя популярными языками программирования (C#, JavaScript), что обеспечивает высокую скорость компиляции скриптов;
* Ряд инструментов для создания качественной графики. Различные режимы освещения, шейдеры, эффекты и другие технологии обеспечат достойное визуальное оформление.
* Превосходный физический движок.
* Высокая производительность.
* Бесплатная версия программы, позволяющая опробовать любому человеку её возможности.

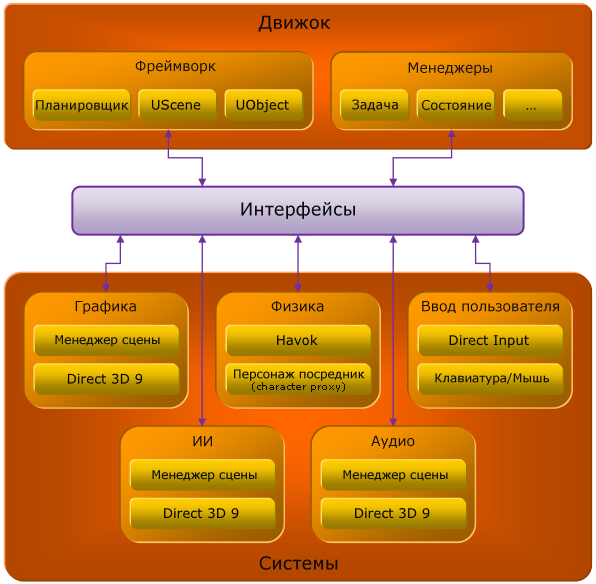


Рисунок 7 **—** Архитектура игрового движка Unity

Для разработки необходимо выбрать язык программирования. Рассмотрим несколько популярных представителей.

Таблица 2 — Сравнения языков программирования для разработки игр

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Язык программирования | Уроки на русском языке | Интеграция с Unity | Наличие разнообразных плагинов, для разработки на Unity | Желание изучить данный язык |
| C# | много | + | много | + |
| JavaScript | много | + | много | - |
| Rust | мало | + | мало | - |

Как на C#, так и на JavaScript существует множество различных уроков на русском языке, большое количество разнообразных плагинов для Unity, в отличии от Rust

Для создания данного проекта используется язык C#. Выбор пал на данный язык из-за простоты его изучения и знакомого синтаксиса.

При написании кода важным фактором является редактор кода. Рассмотрим несколько популярных редакторов кода C#.

**Visual Studio**

Microsoft Visual Studio — линейка продуктов компании Майкрософт, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Microsoft Silverlight

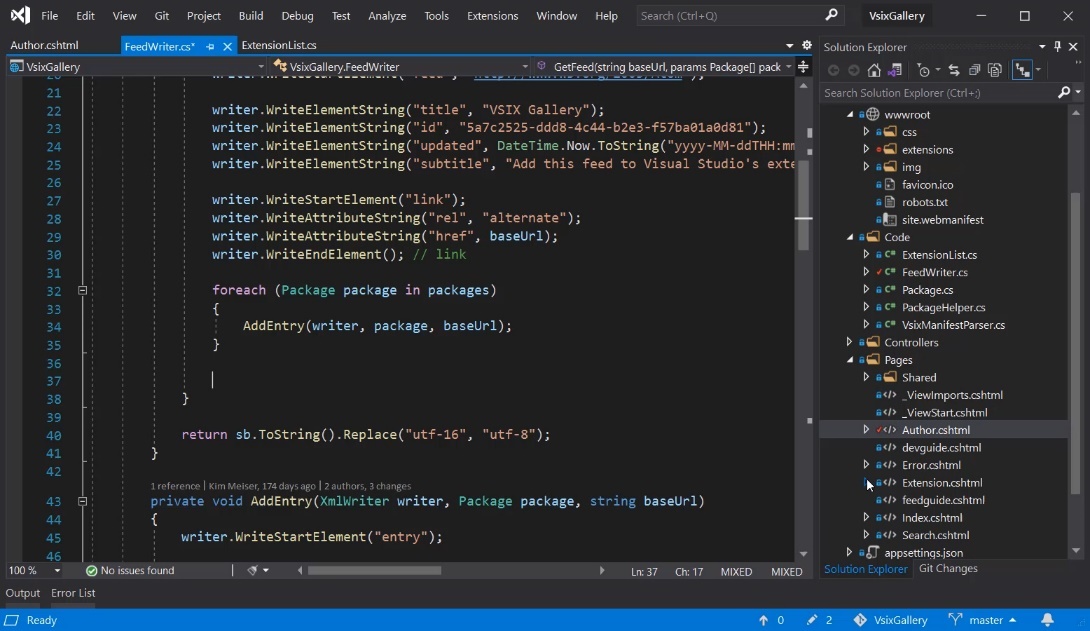


Рисунок 8 — Скриншот IDE Visual Studio

**Project Rider**

Project Rider— это кроссплатформенная IDE для .NET-разработчиков, основанная на платформе IntelliJ и ReSharper.

Rider поддерживает .NET Framework, новую платформу .NET Core и проекты на основе Mono. IDE позволяет разрабатывать десктопные приложения, .NET-сервисы и библиотеки, игры на движке Unity, мобильные приложения Xamarin, веб-приложения ASP.NET и ASP.NET Core.

Помимо умения запускать и отлаживать разные приложения в разных операционных системах, Rider сам по себе тоже поддерживает кроссплатформенность и работает на Windows, macOS и Linux.

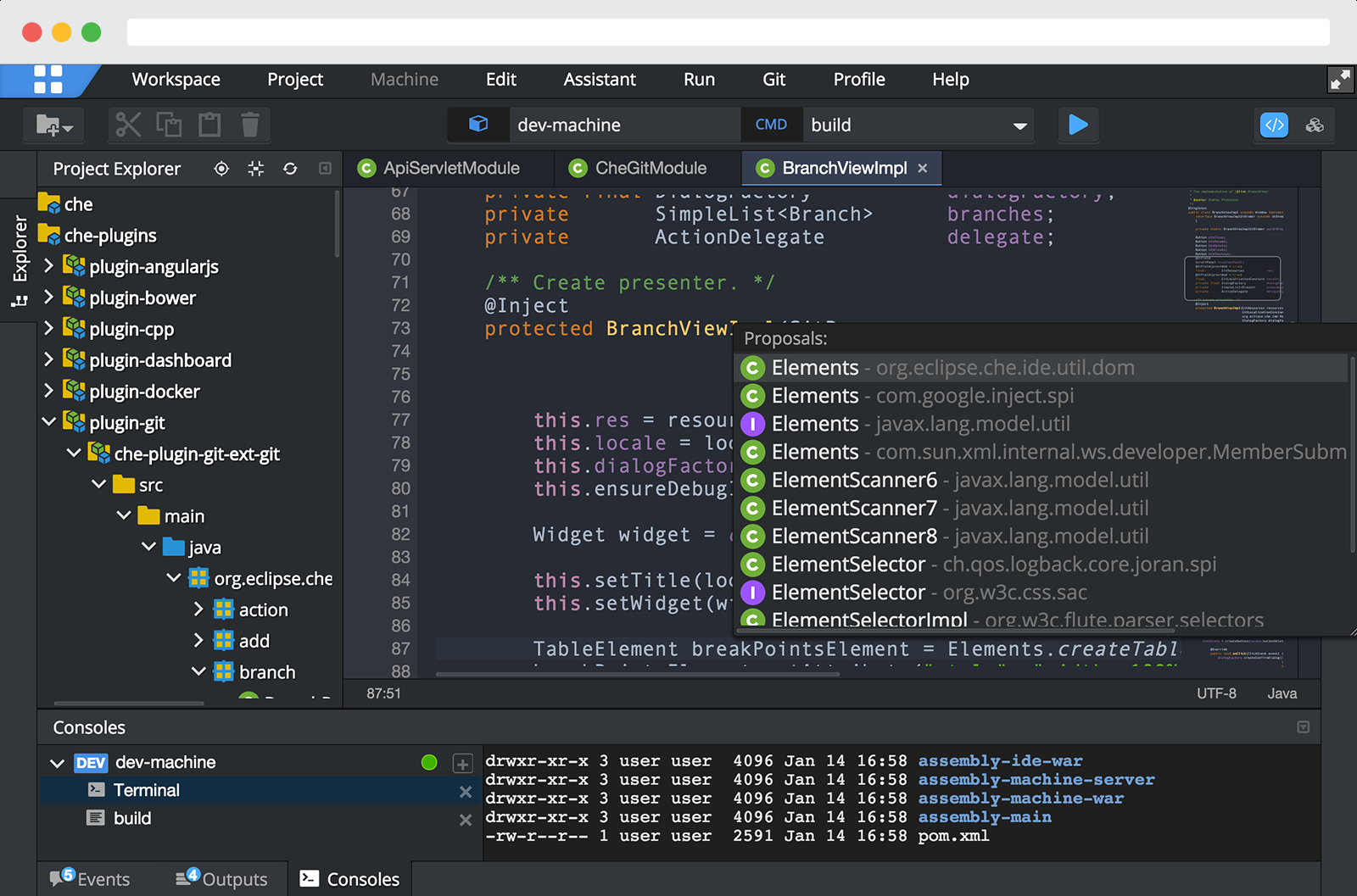


Рисунок 9 — Скриншот IDE Project Rider

**Eclipse**

Eclipse — открытая интегрированная среда разработки на языке программирования Java, которая распространяется и поддерживается Eclipse Foundation. Изначально Eclipse создавалась компанией IDE как преемник среды разработки IBM VisualAge. Несмотря на то, что разработка стоила компании 40 миллионов долларов, исходный код стал открытым, и IBM передала инструмент на дальнейшее развитие независимому сообществу.

При использовании оборотов “самый популярный инструмент” и “бесплатно” многие думают, что функционал IDE урезан и поддерживает стандартный набор функций. Это не так. Eclipse Java IDE — полноценный инструмент, принятый в качестве корпоративного стандарта во многих больших компаниях

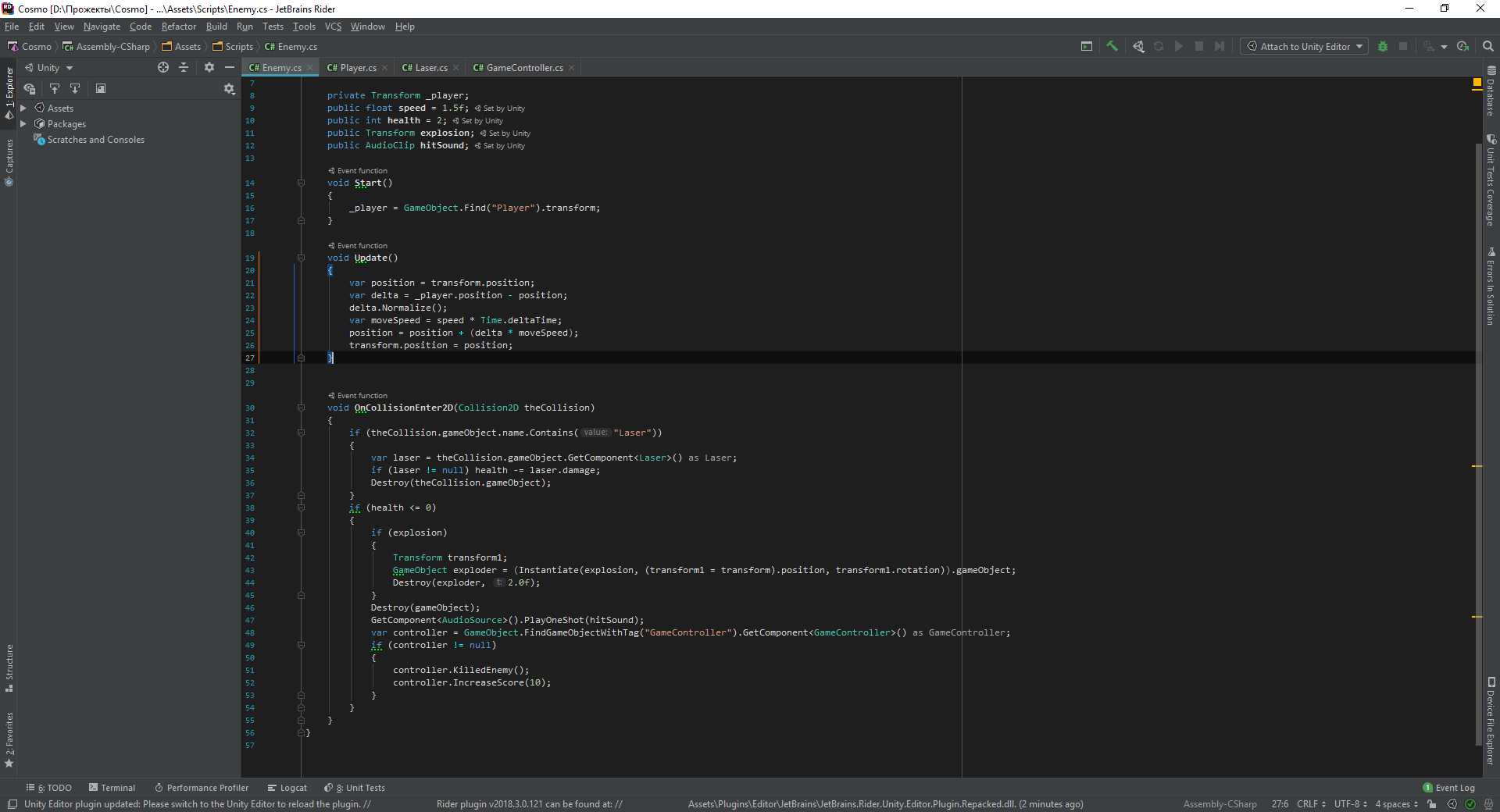


Рисунок 10 — Скриншот IDE Eclipse

Таблица 3 — Сравнения редакторов кода для разработки игр

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Редакторы кода | Бесплатно | Русификация | Количество плагинов | Кастомизация | Сложность в изучении |
| Visual Studio | + | + | много | + | средне |
| Project Rider | - | + | мало | + | легко |
| Eclipse | + | + | много | + | сложно |

Для кода программы был выбран редактор Visual Studio, так как ранее этот редактор был изучен.

# **2 ЦЕЛЬ И ЗАДАЧИ КУРСОВОГО ПРОЕКТА**

Цель работы — разработать мобильное игровое приложение «Rolly Bolly» в жанре 2D платформер, с использованием базовой физики объектов.

В соответствии с поставленной целью были сформулированы следующие 8 задач:

1. формализованное описание игрового процесса как объекта обработки информации,
2. постановка задачи обработки информации,
3. разработка функциональной структуры мобильного игрового приложения «Rolly Bolly».
4. создание алгоритмов и разработка основных механик игрового процесса,
5. разработка структуры интерфейса для игрока,
6. описание структуры программы (модули, основные функции, классы и т. д.),
7. тестирование программного комплекса,
8. оформление документации (пояснительной записки, презентации) по проекту.

# **3 ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ**

# **3.1 Формализованное описание игрового процесса как объекта обработки информации.**

Для разработки данного проекта было необходимо иметь представление о моделируемом объекте, а именно о том, как представить игровой процесс в виде алгоритма, представляющего собой последовательность процессов, подпроцессов и условий.

При моделировании игрового процесса было разработано формализованное описание процесса, представленное на рисунке 11.

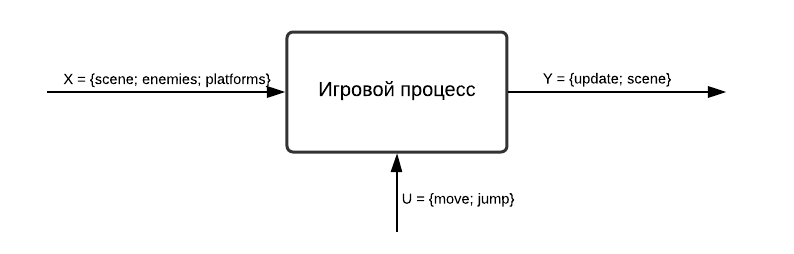


Рисунок 11 **—** Формализованное описание игрового процесса

На рисунке 11 обозначены:

X = {scene; enemies; platforms} — вектор входных данных, где {scene} — игровой уровень, {enemies} — враги, расположенные на сцене, {platforms} — платформы, расположенные на сцене,

Y = {update; scene} — вектор выходных данных, где {update} — обновление игрового процесса и уровня, {scene} — смена уровня

U = {move; jump} — вектор управляющего воздействия, где {move} — команды от пользователя по перемещению персонажа, {jump} — команды от пользователя по прыжку персонажа.

# **Постановка задачи обработки информации**

Постановка задачи обработки информации состоит в воздействии на игровой процесс, с помощью элементов управления пользователя, которое приводит к обновлению информации на экране.

Входными данными в данном приложении являются кнопки управления:

* W, A, S, D **—** передвижение персонажа
* Spacebar **—** прыжок персонажа
* R **—** начать уровень сначала
* Escape **—** выход в главное меню

Выходными данными являются: вывод на экран игрового управления, а также музыкальное сопровождение.

# **Разработка функциональной структуры мобильного игрового приложения «Rolly Bolly»**

Программный комплекс состоит из 5 модулей:

1. подпрограмма, которая обеспечивают работу с меню
2. подпрограмма, отвечающая за показ экрана смерти,
3. подпрограмма, которая отвечает за смену игрового уровня,
4. подпрограмма, отвечающая за обработку игрового уровня и его процессов,
5. подпрограмма, осуществляющая взаимодействие всех программных модулей игры.

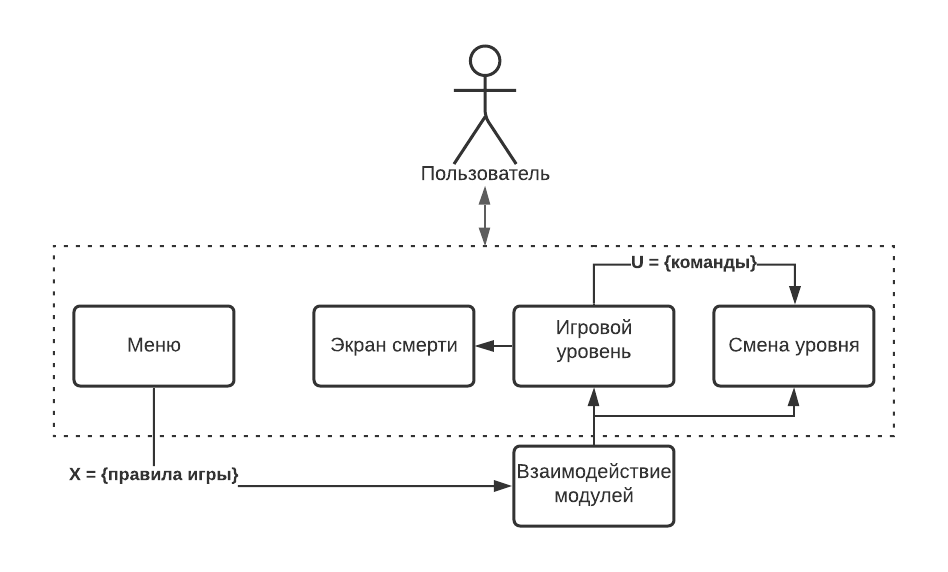
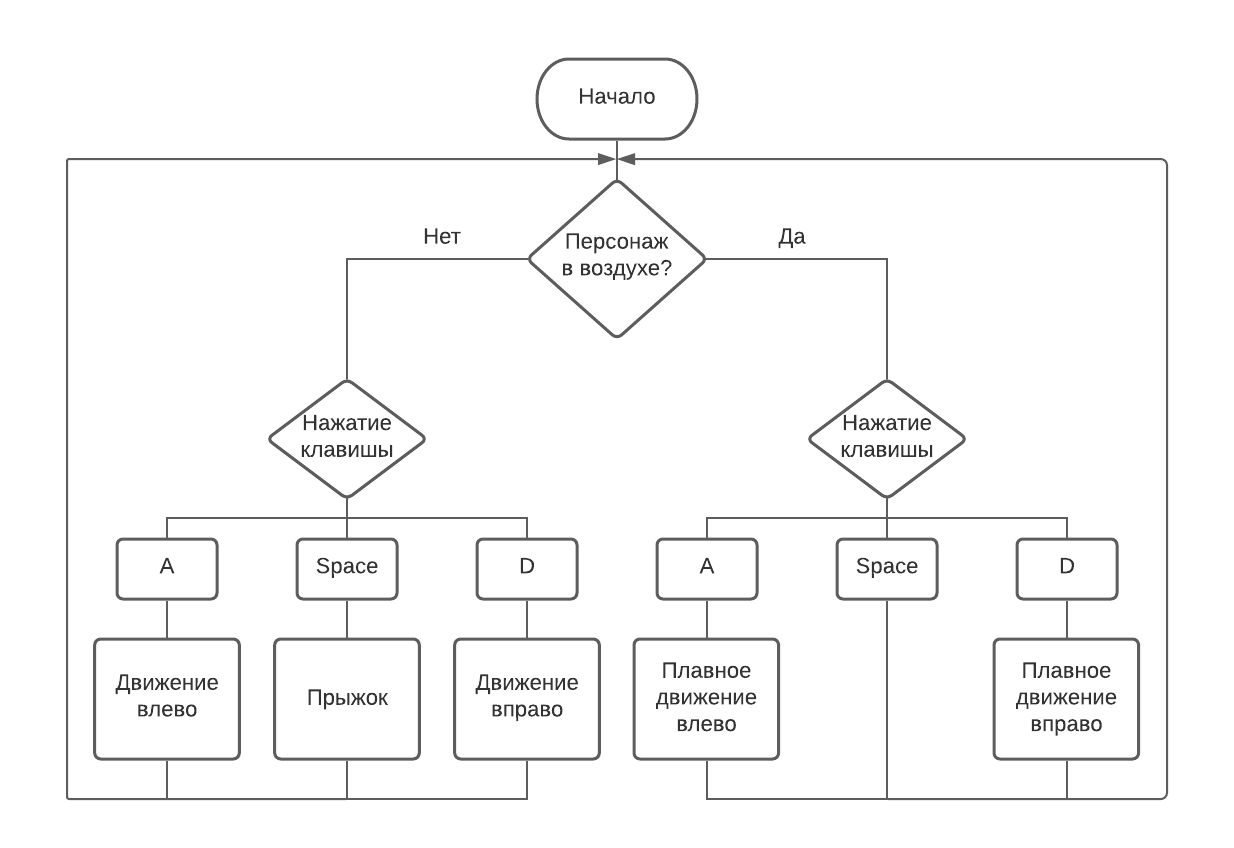


Рисунок 12 — Функциональная структура программного комплекса для мобильного игрового приложения в жанре 2D платформер

# **Разработка блок-схем игровых механик**

При создании программного комплекса необходимо было разработать алгоритмы игровых механик. Алгоритм перемещения персонажа показан на рисунке 13. Алгоритм взаимодействия персонажа с различными объектами показан на рисунке 14.

Рисунок 13 — Блок-схема алгоритма перемещения персонажа

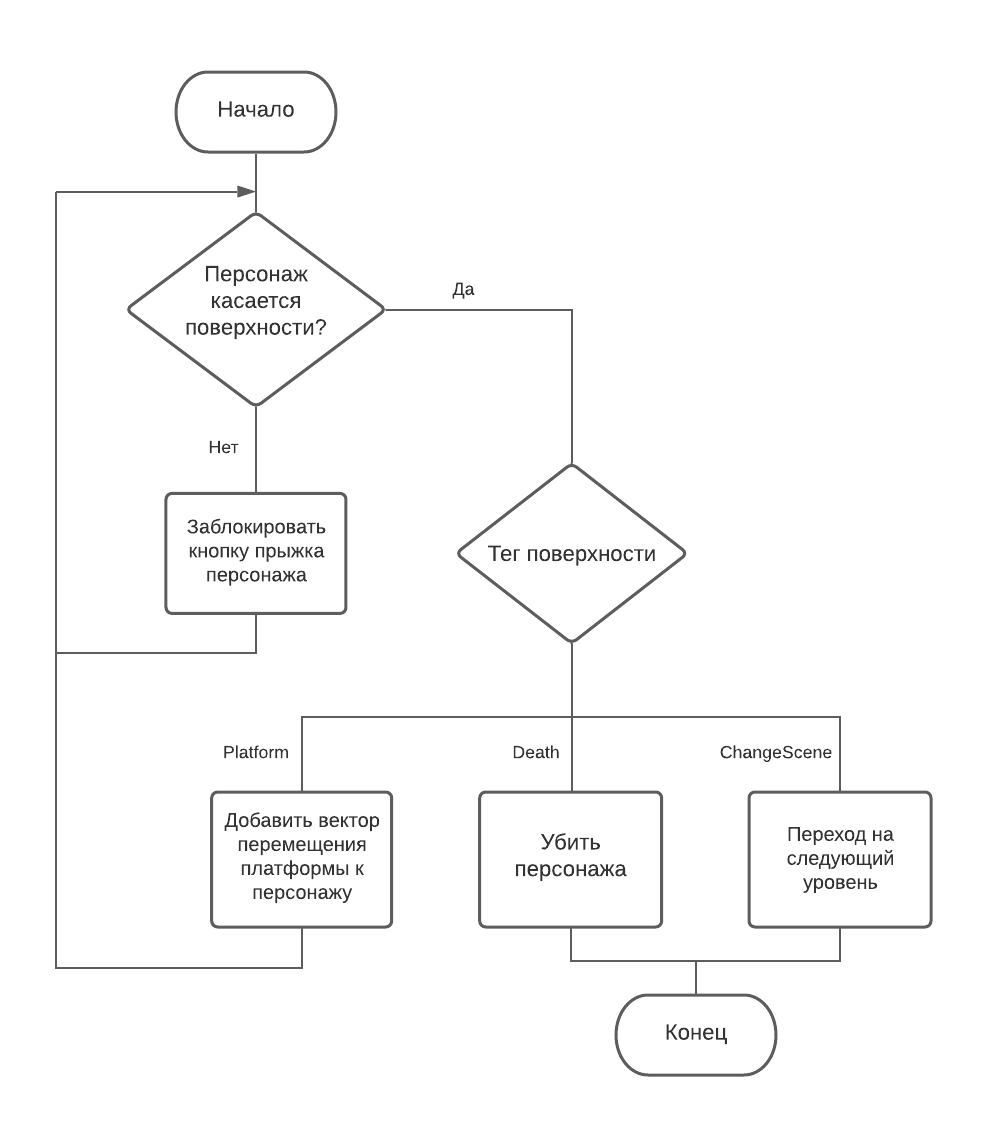


Рисунок 14 — Блок-схема алгоритма взаимодействия игрового персонажа с различными поверхностями

# **3.5 Разработка структуры интерфейса для пользователя программного продукта**

Данный программный комплекс представляет один интерфейс: интерфейс пользователя. На рисунке 8 представлена UML-диаграмма прецедентов использования для пользователя.

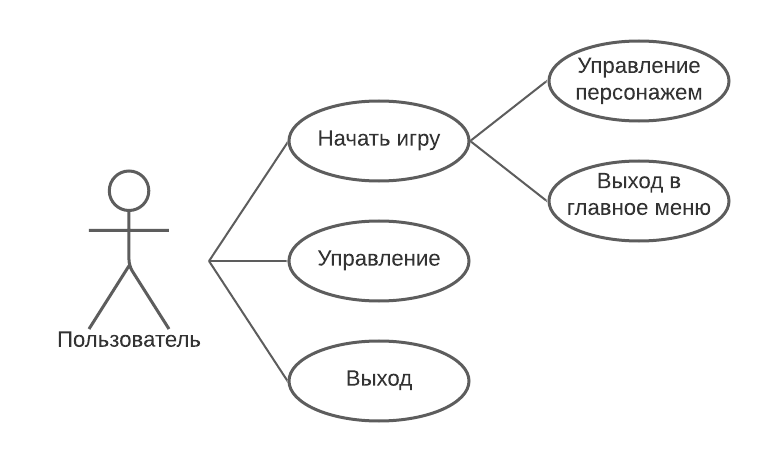


Рисунок 15 — UML-диаграмма действий пользователя

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

Таблица 5 **—** Основные классы и функции

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя класса** | **Описание класса** | **Основные Методы** | **Описание** |
| MonoBehaivor | Основной класс, от которого наследуются все скрипты | Start() | Функция, вызываемая при запуске скрипта |
| Update() | Функция, вызываемая при каждом обновлении кадра |

Продолжение таблицы 5

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя класса** | **Описание класса** | **Основные Методы** | **Описание** |
|  |  | FixedUpdate() | То же, что и Update(), но отличается более частым обновлением. Используется для обработки физики |
| LateUpdate() | Вызывается сразу же после Update() |
| Wait() | Используется для задержки платформ |

Скрипты:

* CameraController.cs — камера;
* ChangeLevel.cs — смена сцены;
* Fading.cs — затемнение экрана при смене сцены;
* Falling\_Obstacle.cs — скрипт вражеского персонажа;
* GroundCheck.cs — проверка персонажа в пространстве;
* ImageSize.cs — адаптация изображения под разрешение экрана;
* Moving.cs — скрипт платформ;
* PlayerController.cs — скрипт персонажа;
* PlayerControllerMenu.cs — скрипт бота главного героя;
* PlayerHolder.cs — удержание персонажа на платформе;
* Respawn.cs — скрипт смерти персонажа;
* SmphBtns — сенсорное управление персонажем.

# **3.7 Тестирование игры**

**Объекты испытаний**

Объект испытаний **—** видеоигра «Rolly Bolly»

Данный проект даёт возможность пользователю развить его логическое мышление внимание, реакцию, так как в игре предстоит логически думать, быстро реагировать на опасности и правильно рассчитывать момент нажатия клавиш.

**Цель испытаний**

Цель проведения испытания **—** проверка функционала программы и устранение возможных ошибок, вызванных тестированием программного продукта.

**Требования к программе**

* Удобный интерфейс;
* Простота усвоения пользователем;
* Игровой мир, в котором будет находиться главный герой;
* Осуществить взаимодействие некоторых предметов с персонажем игрока;
* Осуществить управление персонажем: движение влево, вправо, прыжок, ограничение ускорения персонажа, усложненная управляемость персонажа в воздухе;
* Организовать и осуществить главное меню игры, в котором игрок может начать игру, выйти из игры, посмотреть управление;
* Организовать смерть персонажа и его возрождение;
* Организовать переход между уровнями;
* Организовать выход из игры.

**Методы испытаний**

1. Проверка перемещения персонажей: проверялось путем отклика персонажа (действие) на соответствующую клавишу;
2. Проверка взаимодействия персонажа с платформой: проверялось путем перемещения персонажа на платформу;
3. Проверка перехода между уровнями;
4. Проверка смерти персонажа и его возрождения: проверялось путем попадания персонажа на триггер, вызывающий смерть персонажа;
5. Полностью функционирующее главное меню.

**Тестовый пример**

 Изображение выглядит как текст

Автоматически созданное описание

Рисунок 16 — Проверка перемещения персонажа

Изображение выглядит как текст, знак

Автоматически созданное описание Изображение выглядит как текст

Автоматически созданное описание

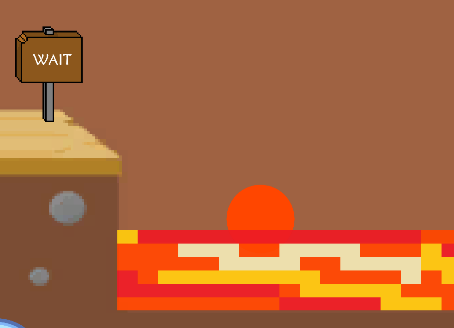
Рисунок 17 — Проверка перемещения персонажа на платформе

Изображение выглядит как текст, игрушка, векторная графика

Автоматически созданное описание Изображение выглядит как текст

Автоматически созданное описание

Рисунок 18 — Проверка перехода между уровнями

 Изображение выглядит как текст

Автоматически созданное описание Изображение выглядит как текст

Автоматически созданное описание

Рисунок 19 — Смерть персонажа и его возрождение

**Изображение выглядит как текст, игрушка, векторная графика

Автоматически созданное описание**

Рисунок 20 — Полностью функционирующее главное меню

# **ХАРАКТЕРИСТИКА ПРОГРАММНОГО И АППАРАТНОГО ОБЕСПЕЧЕНИЯ**

На рисунке 21 изображена трёхуровневая структура ПО.



Рисунок 21 — Трёхуровневая структура программного обеспечения

Таблица 6 и 7 представляют собой характеристику разработанного ПО и минимальные системные требования.

Таблица 6 — Характеристика разработанного ПО

|  |  |
| --- | --- |
| Показатель | Значение |
| Среда разработки | Visual Studio |
| Технология программирования | Объектно-ориентированное программирование |
| Язык программирования | C# |
| Количество переменных | 78 |
| Количество методов | 38 |
| Количество модулей | 12 |
| Размер исполняемого файла, Мб | 38.8 |

Таблица 7 **—** Минимальные системные требования

|  |  |
| --- | --- |
| Показатель | Значение |
| Тип ЭВМ | Смартфон |
| Тактовая частота процессора, ГГц | 0.7 и выше |
| Объём оперативной памяти, Гб | 1 и выше |
| Объём внешней памяти, Мб | 240 |
| Операционная система | Android 4.4 |
| Прикладное программное обеспечение, необходимое для функционирования программного комплекса | нет |

# **ВЫВОДЫ ПО КУРСОВОМУ ПРОЕКТУ**

В ходе работы над курсовым проектом была поставлена цель и определены задачи, которые было необходимо решить в процессе создания мобильной игры в жанре кликер. Итоги работы над задачами:

а) было сформировано формализованное описание игрового процесса как объекта обработки информации, также выделены и обозначены векторы входных данных, вектор управляющего взаимодействия и вектор выходных данных;

б) на основе описания процесса была поставлена задача обработки информации, разработаны варианты взаимодействия пользователя с программным комплексом;

в) программный комплекс состоит из нескольких модулей, разработанных в ходе работы над проектом; они были отражены на функциональной структуре программного комплекса для игрового процесса;

г) разработаны основные алгоритмы игровых механик;

д) создана структура пользовательского интерфейса, отображенная на UML-диаграмме;

е) сформулировано описание структур данных, алгоритмов и функций, используемых в программном комплексе;

ж) был проведен ряд тестовых пусков, продемонстрировавших различные варианты работы программы.

Подводя итог, можно отметить, что цель была достигнута, а все задачи выполнены.

В дальнейших исследованиях данной темы планируется добавить:

а) новые уровни;

б) дополнительные пауэр-апы;

в) новые различные виды противников;

г) еще больше секретов и пасхальных яиц.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. VC.ru **—** крупнейшая в рунете платформа для предпринимателей и высококвалифицированных специалистов малых, средних и крупных компаний [Электронный ресурс]. **—** URL: https://vc.ru/flood/273356-trendy-mobilnyh-igr-v-2021-godu-obzor-industrii-mobilnyh-igr — (дата обращения: 04.08.2021).
2. Прайс, М. C# 7 и .NET Core. Кросс-платформенная разработка для профессионалов / М. Прайс. **—** 3-е изд. **—** Санкт-Петербург : Питер, 2018. **—** 640 с. **—** ISBN 978-5-4461-0516-8.
3. Варламова, Л. Н. Управление документацией : англо-русский словарь

стандартизированной терминологии / Л. Н. Варламова, Л. С. Баюн, К. А. Бастрикова. **—** Москва : Спутник+, 2017. **—** 398 с. **—** ISBN 978-5-9973-4489-4

# Виссер, Д. Разработка обслуживаемых программ на языке С# / Д. Виссер. **—** Москва : ДМК Пресс, 2016. **—** 194 с. **—** ISBN 978-5-97060-446-5

# Шелл, Д. Геймдизайн. Как создать игру, в которую будут играть все / Д. Шелл. **—** Москва : Альпина Паблишер, 2020. **—** 640 с. **—** ISBN 978-5-9614-1209-3

# Васильев, А. Н. Программирование на C# для начинающих. Особенности языка / А. Н. Васильев. **—** Москва : Бомбор, 2018. **—** 528 с. **—** ISBN 978-5-04-092520-9

1. Википедия **—** свободная энциклопедия [Электронный ресурс]. **—** URL: http://wikipedia.org — (дата обращения: 07.08.2021).
2. Голощапов, А. Microsoft Visual Studio 2010 / А. Голощапов. - М.: БХВ-Петербург,

2020. **—** 420 c. **—** ISBN 978-5-9775-0617-5

1. Прохоренок, Н.А. Python3 и PyQt5. Разработка приложений/ Н.А. Прохоренок, В.А.Дронов.-Санкт-Петербург:БХВ-Петербург,2012**—** 704 с. **—** ISBN 978-5-9775-0797-4.