**АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ**

**ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**«БАЛТИЙСКИЙ ИНФОРМАЦИОННЫЙ ТЕХНИКУМ»**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**НА ТЕМУ: «РАЗРАБОТКА ИНФОРМАЦИОННО СПРАВОЧНОЙ СИСТЕМЫ ПО КУЛЬТУРНО-ОБРАЗОВАТЕЛЬНЫМ МЕРОПРИЯТИЯМ ТЕХНИКУМА»**

**Специальность 09.02.06 «ИНОФОРМАЦИОННЫЕ СИСТЕМЫ И ПРОГРАММИРОВАНИЕ»**

Автор выпускной

квалификационной работы \_\_\_\_\_\_\_\_\_\_\_\_ К. В. Липницкий, группа П-20

Руководитель ВКР \_\_\_\_\_\_\_\_ И. О. Янькова, преподаватель АНО ПО «БИТ»

Нормоконтроль\_\_\_\_\_\_\_\_\_\_\_В. В. Барсукова, преподаватель АНО ПО «БИТ»

**К защите допустить:**

Заместитель директора по УВР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Н.Г. Беляева

Выпускная квалификационная работа (дипломный проект)

Выполнена с оценкой:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

Секретарь ГЭК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И. О.) (подпись)

г. Калининград, 2024

**АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ**

**ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**«БАЛТИЙСКИЙ ИНФОРМАЦИОННЫЙ ТЕХНИКУМ»**

# Утверждаю

Заместитель директора по УВР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.Г. Беляева

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

**ЗАДАНИЕ**

**на выполнение выпускной квалификационной работы**

студенту Липницкому Кириллу Владимировичу, учебная группа П-20

1. Тема: «Разработка игрового приложения для обучения студентов правилам дорожного движения»

Утверждена приказом № 169/1 у от « » 202 г.

2. Срок сдачи студентом работы « » 202 г.

3. Дата представления работы к защите«\_\_\_\_» \_\_\_\_\_\_\_\_ 202\_\_ г

4. Перечень основной литературы, материалов практики (исходные данные) : “Джо Хокинг. Unity in Action: Multiplatform Game Development in C#, 2019”. “Джон Маннинг, Парис Батфилд-Аддисон, Тим Нугент. Unity Game Development Cookbook: Essentials for Every Game, 2018”. Дополнительная литература: “Трейси Фуллертон. Game Design Workshop, 2019”

5. Содержание работы

5.1 Теоретическая часть — Описание предметной области, постановка задач, сравнительный анализ средств для разработки программного обеспечения, разработка технического задания

5.2 Практическая часть — Выбор программного обеспечения, разработка интерфейса, имплементация программного кода.

5.3 Экономическая часть – Расчёт себестоимости продукта.

6. Перечень графического и (или) иллюстрационного материала (таблицы, графики, диаграммы, слайды, формы и т.д.): Интерфейс игры, спрайты персонажей, презентация проекта (вкл. доклад-защиту).

**Руководитель ВКР**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата) (фамилия, инициалы)

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г.

**Задание принял к исполнению**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента, дата) (фамилия, инициалы студента)

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г.

**АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ**

**ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**«БАЛТИЙСКИЙ ИНФОРМАЦИОННЫЙ ТЕХНИКУМ»**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**на выпускную квалификационную работу**

**Тема: «Разработка игрового приложения для обучения студентов правилам дорожного движения»**

Руководитель ВКР Работу выполнил

студент группы П-20

И. О. Янькова   К. В. Липницкий

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г. «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г.

# СОДЕРЖАНИЕ

|  |  |
| --- | --- |
| **ВВЕДЕНИЕ**……………………………………………………………….............................. | 5 |
| **ГЛАВА 1 Анализ предмета исследования, разработка технического задания** …...... | 7 |
| * 1. Анализ предприятия..……………..………………………………………….. | 7 |
| * 1. Необходимость разработки игр ……………………………………….…….. | 9 |
| * 1. Обзор существующей игры ………………...………………………………..   2. Техническое задание..........................................................................................   3. Функциональное назначение............................................................................. | 9  14  15 |
| **ГЛАВА 2 Практическая часть**……………………………….………………...………..... | 17 |
| 2.1 Настройка рабочего окружения…………..…….…………………………......... | 0 |
| 2.2 Разработка дизайна программного обеспечения……………………………… | 0 |
| **ГЛАВА 3 Экономическая часть**........................................................................................... | 0 |
| **ЗАКЛЮЧЕНИЕ**………………………………………………………………….…………. | 0 |
| **СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ** ………………………..................... | 0 |
| **ПЕРЕЧЕНЬ СОКРАЩЕНИЙ**…………………………………………………………….. | 0 |
| **ПРИЛОЖЕНИЯ**  **#A: Блок-схемы работы программного модуля «Вход в систему»** …………………... | 0 |
|  |  |

# ВВЕДЕНИЕ

Индустрия компьютерных игр появилась сравнительно недавно, около 30 лет назад, но уже сумела развиться в огромную сферу с большими заработками в несколько миллиардов долларов в год. Понять подобный внезапный рост известности виртуальных развлечений очень просто: все это благодаря широкому распространению компьютерных технологий, в том числе появлению сети Интернет. Благодаря этому, в отличие от других вариантов развлечений, компьютерные игры больше доступны для конечного пользователя. Для того что бы элементарно поиграть игроку необходимо иметь исключительно компьютер или игровую приставку и копию самой игры, а с широким распространением сети Интернет для получения копии игры не нужно выходить из дома. Более того, потребителю не требуется иметь специализированных познаний для того что бы выбрать подходящую для него игру, в то время как для большинства прочих видов развлечений нужно разбираться как минимум в необходимой экипировке.

Так же стоит принять во внимание, что компьютерные игры в последнее время перестали позиционироваться как программы исключительно для отдыха и развлечений. К примеру, сегодня, из-за применения игровых технологий, основываются особые комплексы по симуляции, которые служат для обучения специалистов в различных областях: от лесорубов до пилотов реактивных самолетов.

В России же, к сожалению, все несколько иначе. Игровая индустрия у нас, также как и в большинстве других стран постсоветского пространства, развита крайне слабо. Связано это с тем, что культура компьютерных развлечений пришла к нам слишком поздно и практически не развивалась. Из-за этого, даже при достаточно высоком спросе, мы имеем слишком малое количество компаний-разработчиков, которые могут составить конкуренцию зарубежным компаниям.

Поэтому развитие технологий в данном направлении можно считать одним из наиболее перспективных, особенно в нашей стране.

Актуальность выпускной квалификационной работы заключается в том, что игровая индустрия популярна и востребована, как на мобильных телефонах, так и на персональных компьютерах. Игры используются в развлекательных целях, в образовательных и тренировочных целях. Сфера разработки игр является актуальной сферой, которая продолжает развиваться и привлекать внимание как игроков, так и разработчиков. Она предлагает возможности для проявления творчества, инноваций и создания уникальных игровых продуктов, которые могут доставить удовольствие и развлечение игрокам.

Предмет исследования: технологии разработки компьютерной игры жанра 2D-платформер.

Объектом исследования является разработка компьютерных игр.

Цель выпускной квалификационной работы – разработка игрового приложения в Unity 3D.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. изучить особенности и состояние компьютерной индустрии России;

2. выбрать жанр, вид и платформу для компьютерной игры;

3. разработать сценарий, концепцию основных элементов;

4. выбрать и изучить средство реализации;

5. подготовить необходимые для игры материалы;

6. разработать прототип игры.

В первой главе рассмотрены основные этапы проектирования игр, средства для разработки программного обеспечения, исследование предметной области, описание технического задания.

Во второй главе выполнен выбор жанра и платформы, разработана концепция основных элементов, подготовлены необходимые материалы, разработан прототип игры.

В третьей главе выполнен расчёт экономической эффективности разрабатываемой игры, расчёт постоянных издержек, переменных издержек, общих издержек и эксплуатационные затраты.

**Глава 1** **Анализ предмета исследования, разработка технического задания**

**1.1 Анализ предприятия**

АНО (Автономная некоммерческая организация) "Балтийский информационный техникум" - это учебное заведение, которое специализируется на обучении студентов в области информационных технологий. Вот некоторая подробная информация о техникуме:

Сокращенное наименование: АНО ПО «БИТ»

Дата создания образовательной организации: 28 августа 1999 года

Место нахождения образовательной организации:

236016, Российская Федерация, г. Калининград, ул. Литовский вал, д.

38, литер А1, пом. YII, подъезд 6.

Учредитель:

Сергеев Вячеслав Владимирович

1. Организационная структура: Балтийский информационный техникум является самостоятельной организацией, не имеющей коммерческой деятельности. Возможно, у него есть правление или руководство, которое занимается управлением и принятием решений в учебном заведении.
2. Образовательные программы: Техникум предлагает различные образовательные программы по информационным технологиям. Это могут быть программы по программированию, системному администрированию, разработке веб-сайтов и другим аспектам IT-сферы. Программы могут быть как дневными, так и вечерними. Проверь нумерацию, снизу одна сверху другая
3. Курсы и сертификаты: Возможно, техникум предлагает различные курсы и сертификаты для учащихся и выпускников. Это может включать в себя курсы профессиональной переподготовки или повышения квалификации, а также получение сертификатов, подтверждающих определенные навыки и знания.

4.Обучающий персонал: Техникум, вероятно, имеет квалифицированный обучающий персонал, состоящий из преподавателей, специалистов в области информационных технологий. Они играют важную роль в обучении студентов и передаче им необходимых знаний и навыков.

5.Инфраструктура: Техникум обладает соответствующей инфраструктурой для обучения студентов. Это может включать в себя компьютерные классы с необходимым программным обеспечением, лаборатории для проведения практических занятий и другие учебно-материальные ресурсы.

6.Студенческая жизнь: Как и в большинстве учебных заведений, есть вероятность наличия студенческой жизни в техникуме. Это может включать в себя различные клубы и общественные организации, спортивные мероприятия и другие активности, созданные для социального и духовного развития студентов

7.Сотрудничество с другими организациями: Возможно, техникум имеет партнерские отношения или сотрудничает с другими организациями и предприятиями в сфере информационных технологий. Это может обеспечивать студентам возможности стажировки, практики или трудоустройства после окончания обучения.

Балтийский информационный техникум был основан в 1994 году и имеет статус государственного учебного заведения. Учебное заведение предоставляет высокое качество образования в области информационных технологий, формирует профессиональные навыки студентов и готовит их к успешной карьере в сфере IT.

На данный момент БИТ предлагает обучение по следующим специальностям:

" Обеспечение информационной безопасности автоматизированных систем "

" Информационные системы и программирование "

" Сетевое и системное администрирование "

Преподаватели техникума - высококвалифицированные специалисты в области информационных технологий, которые имеют опыт работы в индустрии и постоянно совершенствуют свои навыки. Учебная программа комбинирует теоретические знания и практическую работу, чтобы студенты могли сразу применить свои навыки на практике.

Студенты БИТ имеют доступ к современному оборудованию и программным продуктам, что позволяет им получить реальный опыт работы в IT-отрасли. Техникум также сотрудничает с ведущими компаниями и предлагает студентам практику и стажировку. После окончания образовательной программы студенты получают государственный диплом и имеют хорошие перспективы для дальнейшей карьеры в IT-сфере.

Балтийский информационный техникум - это учебное заведение, которое активно развивается и следит за новейшими тенденциями в IT-отрасли. Он предлагает студентам широкие возможности для развития и приобретения ценных навыков, которые будут полезны в будущей карьере

**1.2 Необходимость разработки игр**

В современности компьютерные технологии активно интегрируются в различных сферах нашей жизни. Программирование разнообразных простых и сложных приложений для разных областей продолжает развиваться, и среди них особое место занимают компьютерные игры.Игры представляют собой эффективный метод обучения для людей всех возрастов, а также для высокоорганизованных машин с искусственным интеллектом, которые могут эффективно обучаться, наблюдая или участвуя в играх. С использованием IT-технологий игры становятся более доступными и разнообразными. Игровая активность способствует когнитивному развитию, представляя практическую форму мышления. Благодаря особенностям игры, люди овладевают абстрактным мышлением. Компьютерные игры, в свою очередь, представляют новую форму развивающего обучения.

Использование компьютерных технологий открывает новые перспективы в применении педагогических методов:

* Позволяет предложить индивидуально подходящий материал различной сложности в соответствии с возможностями и учебными целями каждого человека.
* Помогает выявить труднодоступные проблемы в развитии учащихся, которые могут быть не обнаружимы в традиционном обучении.
* Позволяет ученикам лучше понять процесс развития своих собственных навыков.
* Сложное программное обеспечение становится простым в использовании.
* Переносит успешно применяемые методы на компьютерную платформу, обогащая знания и навыки преподавателей и давая им возможность переосмыслить методики.
* Создает комфортные условия для учеников, способствующие успешному выполнению упражнений.
* Обеспечивает увлекательные занятия, используя экспериментирование, моделирование и сравнение.
* Помогает людям правильно высказываться и исправлять ошибки с помощью самоконтроля и привлекательной графики.
* Позволяет учащимся видеть результат своих усилий на экране.

Таким образом, использование компьютерных программ стимулирует мотивацию не только к стратегии игры, на которой основана программа, но и к получению одобрения и похвалы. Однако следует использовать компьютерные игры разумно, выборочно и творчески. Большинство игр направлено на развитие психических процессов, таких как память, внимание, воображение, восприятие, мышление, а также координацию глаз и рук.

Жанры компьютерных игр:

1. 3D Shooter (3D-шутеры, "бродилки")

Этот жанр компьютерных игр основан на создании виртуального трехмерного пространства, где игрок может взаимодействовать с окружающим миром. Основной поджанр - "шутер от первого лица", где игровое изображение имитирует вид из глаз игрока.

2. Arcade (аркады)

Аркады представляют собой быстрые игры, требующие от игрока быстрых рефлексов и реакции. В них часто присутствует система бонусов, начисление очков и постепенное открытие новых элементов игры. Изначально аркады были простыми, чтобы привлечь больше игроков, и впоследствии стали основным жанром игровых приставок.

3. Fighting (Драки)

Жанр драк представляет игры, в которых два персонажа сражаются на арене, используя различные удары, броски и комбинации. Хотя этот жанр не так популярен на компьютере из-за ориентации на мультиплеерную игру, на игровых приставках он развит и даже проводятся мировые чемпионаты по некоторым играм этого жанра.

4. Platformer (Платформеры)

Платформеры - один из самых популярных жанров игр, требующий от игрока концентрации внимания, реакции и усидчивости. В этих играх игрок должен преодолевать препятствия, прыгая с платформы на платформу и избегая врагов. Жанр получил свое название от абстрактно размещенных в воздухе "палочек" - платформ, на которые приходится прыгать.

5. Scrollers (Скроллеры)

В скроллерах экран непрерывно движется в одном направлении, а игроку предлагается уничтожать появляющихся врагов и собирать бонусы. Этот жанр был популярен в 90-х годах, но сейчас его выпуск почти прекратился.

6. Simulation (симуляторы)

Симуляторы - это игры, которые пытаются максимально реалистично имитировать управление сложной технической системой, такой как боевой истребитель или автомобиль.

7. Strategy (стратегии)

Жанр стратегий требует от игрока разработки стратегии, особенно в военных операциях. Игрок управляет целым подразделением, предприятием или даже вселенной, взвешивая различные аспекты и принимая тактические решения.

8. Sport (спортивные)

Спортивные игры имитируют различные виды спорта, например футбол или баскетбол. Они предлагают игроку участие в виртуальных спортивных состязаниях.

9. Puzzle (головоломки, логические)

Жанр головоломок и логических игр предлагает игроку различные головоломки и задачи, которые требуют логического мышления и размышлений. Они часто не зависят от реакции игрока и могут включать в себя счет времени, потраченного на решение.

10. Traditional (традиционные) и board (настольные)

Этот жанр включает компьютерные версии настольных игр, таких как шахматы или монополия, которые предлагают игроку возможность играть в классические настольные игры на компьютере.

Таким образом, компьютерные игры охватывают широкий спектр жанров, позволяя игрокам выбирать игры в соответствии с их предпочтениями и интересами.

С ростом компьютерных технологий все больше людей, известных как "компьютерные фанаты" или "геймеры", становятся увлеченными игрой на компьютере. У них ограниченный круг социальных контактов, а их основной фокус - игра на компьютере. Это формирует класс фанатов компьютерных игр, и взаимодействие с ними показывает, что для многих из них увлечение компьютером может иметь негативные последствия, а некоторым может потребоваться психологическая помощь. Большинство из них имеют определенные психологические проблемы, такие как неудовлетворенность личной жизнью, недостаток самооценки и потеря смысла жизни и нормальных ценностей. Для них компьютер и все, что связано с ним, становятся единственной ценностью. Это формирует психологическую зависимость.

Компьютерные игры могут быть разделены на ролевые и не ролевые, и это разделение имеет важное значение, поскольку механизмы психологической зависимости от них отличаются. Наибольшее влияние на психику игрока оказывают ролевые игры, особенно те, где игрок видит игровой мир глазами своего персонажа. Этот тип игр создает сильное вовлечение и идентификацию с виртуальным героем, а игрок может серьезно относиться к виртуальному миру и его действиям. Таким образом, ролевые компьютерные игры существенно влияют на личность, и проблемы, решаемые виртуально, могут переноситься в реальную жизнь.

Развитие компьютерной зависимости проходит через несколько стадий. На стадии легкой увлеченности игра еще не является значимой ценностью, но игрок получает удовлетворение и положительные эмоции от игры. На стадии увлеченности игра становится систематической. Затем наступает стадия зависимости, где игра становится основной потребностью и происходят серьезные изменения в ценностно-смысловой сфере личности. Зависимость может принимать две формы: социализированную и индивидуализированную.

Социализированная форма игровой зависимости подразумевает поддержание социальных контактов внутри игрового сообщества, и мотивация игры в основном имеет соревновательный характер. Эта форма зависимости имеет меньшее негативное влияние на психику человека, чем индивидуализированная форма.

Индивидуализированная форма зависимости является крайней формой, при которой нарушается не только нормальное восприятие мира, но и взаимодействие с реальным окружающим миром. Психика перестает отражать объективную реальность, и виртуальный мир становится основой восприятия.

Наконец, стадия привязанности характеризуется снижением активности игры и возвращением психологического содержания личности к норме. Человек сохраняет некоторую эмоциональную привязанность к компьютерным играм, но может удерживать некую дистанцию. Однако полное освобождение от психологической привязанности к компьютерным играм может быть длительным процессом, зависящим от скорости снижения привязанности.

Таким образом, процесс развития компьютерной зависимости проходит через несколько стадий, начиная с увлечения и заканчивая привязанностью. Важно осознавать потенциальные риски и относиться к компьютерным играм с умеренностью, чтобы избежать негативных последствий для психического и социального благополучия.

Компьютерные игры, также известные как видеоигры, представляют собой программы, созданные для организации игрового процесса. Они могут использоваться для взаимодействия с партнером по игре или самостоятельно выступать в качестве партнера. Компьютерные игры могут быть основаны на сторонних источниках, таких как фильмы или книги, и могут включать дополнительные материалы, расширяющие игровую вселенную. Некоторые игры также могут использоваться в образовательных или научно-исследовательских целях.

Киберспорт, организованные соревнования по компьютерным играм, стали популярным явлением. Компьютерные игры оказывают значительное влияние на общество, и в последнее время наблюдается тенденция к геймификации в других сферах, где компьютерные игры используются для неигрового программного обеспечения, включая обучение и научные исследования.

Хотя существует разнообразие жанров и подкатегорий компьютерных игр, классификация не является полной и может быть дополнена. В разработке компьютерных игр обычно применяется общий алгоритм, который включает проектирование, разработку и издание игры.

На этапе проектирования определяются цель игры, ее идея, жанр и сеттинг. Выбор средств разработки, включая программный код и игровой движок, также является важной частью процесса. Выбор платформы, для которой разрабатывается игра, влияет на выбор языка программирования.

Компьютерные игры стали неотъемлемой частью нашей жизни и их применение расширилось за последние 10 лет, включая развлечение, обучение и научные исследования.

К сожалению, данная классификация не является исчерпывающей и может быть дополнена. Например, в список жанров можно включить музыкальные игры, где основное внимание уделяется музыке и звукам, а многопользовательские игры можно разделить на подкатегории. Важно отметить, что хотя данная классификация достаточна для определения большинства существующих игр, в игровой индустрии до сих пор нет единой и полной классификации. Это связано с тем, что многие игры не подходят под одну категорию. Некоторые игры могут сочетать несколько жанров, быть доступными на разных платформах и предлагать как одиночный, так и многопользовательский режимы игры.

Это объясняется тем, что игровая индустрия сравнительно молода по сравнению с другими развлекательными секторами. Однако для определения большинства существующих компьютерных игр на данный момент достаточно данной классификации.

Исходя из этой классификации, было принято решение разработать двухмерный однопользовательский платформер для персональных компьютеров. Такой выбор обусловлен простой и понятной механикой платформера, что делает его доступным для начинающих разработчиков. Кроме того, разработка двухмерных игр является более простой по сравнению с трехмерными.

Общий алгоритм разработки компьютерной игры схож с алгоритмом разработки другого программного продукта и состоит из трех основных этапов: проектирование, разработка и издание с последующей поддержкой.

На этапе проектирования определяется цель игры и выбираются инструменты для её разработки. Цель игры определяет идею, жанр и сеттинг. Например, идея RPG может заключаться в возможности игрока прожить свою роль по своему усмотрению, а идея шутера - в участии игрока в реальных или вымышленных боевых действиях. Жанр выбирается на основе основных идей игры.

После выбора жанра и идеи игры следующим шагом является выбор сеттинга - окружения, в котором происходит основное действие игры. Заранее исходя из предпочтений целевой аудитории лучше выбирать сеттинг, поскольку это значительно облегчает разработку сценария игры.

К программным средствам разработки в первую очередь относятся программный код и игровой движок. От правильного выбора этих компонентов зависит скорость разработки и функциональность конечного продукта. Выбор программного кода в основном зависит от платформы, на которой будет выпущена компьютерная игра. Например, для игр, предназначенных для браузеров, разумно использовать языки программирования Java или Flash, в то время как для игр на персональных компьютерах хорошим выбором может быть язык программирования C#.

Игровой движок выполняет функции низкоуровневого описания физики объектов, правил рендеринга графики и других задач. При выборе игрового движка важно учитывать его доступность и язык программирования, который используется. Например, Unity - игровой движок, который позволяет разрабатывать игры на языке C# и бесплатен для компаний с доходом, не превышающим 100 000 долларов в год.

После определения целей игры и выбора средств разработки, наступает этап разработки. Разработка - это самый большой и длительный этап, включающий множество шагов, необходимых для создания работоспособного продукта.

Вначале необходимо придумать сюжет и выбрать игровую механику. Игровая механика определяет объекты и правила, по которым игрок будет взаимодействовать с игрой. Вместе с разработкой игровой механики часто проводится написание сюжета игры. Сюжет играет важную роль, влияя на интересность игры для игрока. Сюжет может быть представлен в форме литературного или режиссерского сценариев. Литературный сценарий описывает основные события и персонажей игры, а режиссерский - детально описывает уровни и события в игре.

На этом этапе начинается также предварительная разработка графики и дизайна игры. На основе сюжета и обсужденного дизайна создаются концепт-арты, которые затем используются для разработки основного вида игры и персонажей.

После разработки сюжета и игровой механики начинается самая важная часть - разработка самой игры. Исходя из сюжета и концепт-артов, создаются персонажи и объекты игры, одновременно с этим разрабатываются игровые уровни. При разработке уровней сначала создается упрощенный план, на котором схематически изображен сам уровень и предметы, с которыми будет взаимодействовать игрок.

Затем создается первая версия уровня, которая обычно представляет собой почти пустую локацию с минимальным количеством предметов, необходимых для прохождения. Эта версия уровня используется для тестирования проходимости. После тестирования уровень постепенно заполняется другими объектами.

Сразу после создания первых уровней происходит сборка первого прототипа игры, который называется альфа-версией. Альфа-версия необходима для тестирования основной механики игры и проверки ее соответствия заявленным требованиям. Часто в альфа-версии игры объекты либо не имеют текстур, либо представлены в виде абстрактных объектов.

Если альфа-версия игры успешно проходит тестирование, наступает следующий этап - доработка механики и объектов игры. На этом этапе происходит улучшение уровней и игровой механики, а также добавление сюжетных элементов, таких как видеоролики, сюжетные диалоги и кат-сцены. Также исправляются ошибки и неисправности в игровом коде, которые были выявлены в альфа-версии игры.

Затем начинается создание второго прототипа игры, известного как бета-версия. Бета-версия представляет собой практически готовую игру и используется для тестирования наличия неисправностей. В бета-версии могут отсутствовать незначительные элементы, которые не влияют на игровой процесс. Часто для тестирования бета-версии игры приглашают обычных игроков, особенно если игра мультиплеерная. Это позволяет ускорить процесс тестирования и снять нагрузку с разработчиков.

Если игра успешно проходит бета-тестирование, она отправляется на окончательную доработку и исправление критических ошибок, после чего создается финальная версия игры и происходит ее релиз.

После выпуска игры необходимо обеспечить ее последующую поддержку. Это включает выпуск патчей, которые исправляют ошибки в игре, а также выпуск дополнительного контента в виде DLC (Downloadable Content), который добавляет новые предметы и возможности для игроков.

Таким образом, этапы разработки компьютерных игр в значительной степени аналогичны этапам разработки других программных продуктов. Основными инструментами разработки игр являются игровые движки, которые отвечают за физическую модель игры, графический рендеринг и другие аспекты. Также используются графические редакторы для создания игровой графики.  
Для разработки компьютерных игр основным инструментом является игровой движок - программное обеспечение для создания игр. Игровые движки включают в себя необходимые алгоритмы для разработки и функционирования игры. Существует множество игровых движков с различными особенностями, поддерживаемыми языками программирования и стоимостью лицензии. Одни из самых популярных игровых движков, это Unreal Engine, Unity и Game Maker.

Game Maker: Studio, разработанный и поддерживаемый компанией YoYo Games, является популярным игровым движком для создания двухмерных игр на различных современных платформах. Он написан на Delphi (с будущей версией на C#) и использует свой собственный простой язык программирования GML (Game Maker Language) для разработки.

Игровой движок имеет несколько версий для распространения:

* Trial (пробная)
* Desktop (для настольных компьютеров)
* Web (для веб-платформ)
* UWP (для универсальной платформы Windows)
* Mobile (для мобильных платформ)

Каждая из этих версий, за исключением Trial, предназначена для разработки игр на определенных платформах. Пробная версия является бесплатной и предоставляется тем, кто хочет

ознакомиться с игровым движком, но она имеет ограничения на количество объектов, используемых в игре, и позволяет компилировать проект только для тестирования на операционной системе Windows.

Unreal Engine – это игровой движок, который разрабатывается и поддерживается компанией Epic Games. Первая игра на этом движке, названная Unreal, была выпущена в 1998 году. С тех пор различные версии Unreal Engine были использованы в более чем ста играх и других проектах.

Движок написан на языке C++ и позволяет создавать игры для широкого спектра операционных систем и платформ, включая Microsoft Windows, Linux, Mac OS и Mac OS X, а также консоли Xbox, Xbox 360, Xbox One, PlayStation 2, PlayStation 3, PlayStation 4, PSP, PS Vita, Wii, Dreamcast, GameCube и другие устройства, работающие под управлением iOS и Android.

Для облегчения портирования Unreal Engine использует модульную систему зависимых компонентов. Он поддерживает различные системы рендеринга, такие как Direct3D, OpenGL, Pixomatic, а также предоставляет функциональность воспроизведения звука (EAX, OpenAL, DirectSound3D), голосового воспроизведения текста, распознавания речи, модули для работы с сетью и поддержки различных устройств ввода.

С 2 марта 2015 года Unreal Engine 4 стал доступным бесплатно. Однако разработчики игр до сих пор обязаны отчислять 5% прибыли от своих игр компании Epic Games, если доходы превышают $3000 за квартал.

Unity является одним из самых популярных инструментов разработки двух- и трехмерных игр в настоящее время. Он позволяет создавать приложения, которые работают на большинстве современных операционных систем (Windows, OS X, Windows Phone, Android, Apple iOS, Linux), а также на игровых приставках Wii, PlayStation 3, PlayStation 4, Xbox 360, Xbox One и дисплеях MotionParallax3D (таких как Nettlebox), используемых для воспроизведения виртуальных голограмм. Unity также поддерживает создание приложений, которые можно запускать в веб-браузерах с помощью специального плагина Unity (Unity Web Player), а также с использованием технологии WebGL. Последние версии Unity позволяют создавать приложения для виртуальной реальности.

Редактор Unity обладает простым интерфейсом Drag&Drop, который легко настраивается, и состоит из различных окон, позволяя производить отладку игры прямо в редакторе. Движок поддерживает два сценарных языка: C# и JavaScript (в модифицированной форме). Физические вычисления осуществляются с помощью физического движка PhysX от NVIDIA.

Проект в Unity состоит из сцен (уровней) - отдельных файлов, содержащих игровые миры с их собственным набором объектов, сценариев и настроек. Сцены могут содержать как моделированные объекты, так и пустые игровые объекты без модели. Объекты в Unity содержат компоненты, с которыми взаимодействуют скрипты.

Unity распространяется бесплатно, но существуют также четыре платные версии: стандартная Unity, Unity iOS Pro, Android Pro и командная лицензия. Они отличаются стоимостью и функциональностью.

Бесплатная версия Unity имеет некоторые ограничения, но позволяет распространять игры, если годовой доход от игры не превышает 100 000 долларов. С выпуском Unity 5 многие ограничения бесплатной версии были убраны, но ограничение на годовой доход все еще остается - он не должен превышать 100 000 долларов.

Все это делает Unity одним из наиболее предпочтительных движков для начинающих игровых разработчиков.

**1.4 Техническое задание**

1. Общие сведения.

1.1. Название продукта разработки.

Разработать игровое приложение с названием «ПДД» в жанре 2D-платформера на Unity.

1.2. Назначение продукта

Игровое приложение.

1.3. Плановые сроки начала и окончания работ.

В соответствии с ВКР (30.09.2022 - 15.04.2023)

2. Характеристика области применения продукта.

2.1. Процессы и структуры, в которых предполагается использование продукта разработки.

Игровое приложение для продажи на сайте.

Выпускная квалификационная работа должна быть выполнена несколько этапов:

* Постановка задачи.
* Разработка технического задания.
* Создать необходимые материалы для игры.
* Разработка игры.
* Тестирование приложения.
* Оформление пояснительной записки.
* Защита выпускной квалификационной работы.

2.2. Характеристика персонала (количество, квалификация, степень готовности)

Целевая аудитория – любые пользователи от 6 лет, желающие весело провести время.

3. Требования к продукту разработки.

3.1. Требования к продукту в целом.

Входными данными в игре является непосредственное управление во время игрового процесса с помощью клавиатуры. Проект относится к играм в реальном времени, где действия игрока незамедлительно оказывают влияние на игровой процесс.

Требуется разработать: игровое меню с выбором настроек, выхода и запуском игры. Меню выбора уровней. Игровое пространство, состоящее из разных платформ. Возможность перенаправлять персонажа в любой момент времени. Шкала здоровья. Добавить врагов, мешающих прохождению. Дополнительные предметы, лечащие игрока. Уровень должен быть проходимым.

3.2. Требования к надёжности.

В игре должна присутствовать проверка входных данных на соответствие типов, принадлежность диапазону допустимых значений. В случае возникновения ошибок предусмотреть возможность вывода информативных диагностических сообщений. Требуется система логов. Проект должен запускаться без каких-либо ошибок.

3.3. Аппаратные требования.

Для создания и использования программы необходимы следующие минимальные аппаратные требования:

* ОС (операционная система): Windows
* Процессор с 2 ядрами
* Оперативная память – 2 Gb
* Жёсткий диск – 5 Gb
* Видеокарта – 512 Mb
* Клавиатура, мышь

3.4. Указание системного программного обеспечения (операционные системы, браузеры, программные платформы и т.п.)

Для создания и использования программы необходимо следующее программное обеспечение: Windows 7/Windows 8/Windows 10/Windows 11

3.5. Указание программного обеспечения, используемого для реализации.

Программная среда разработки Unity 3D.

4. Требования к пользовательскому интерфейсу.

4.1. Общая характеристика пользовательского интерфейса.

# 

# ГЛАВА 2 Разработка игрового приложения

## **2.1 Этап разработки графического оформления**

Было решено дать игре имя «ПДД». «ПДД» это игра-платформер о правила дорожного движения, который обучают студентов знания правил. На рисунке 1 показана схема функционала игры.

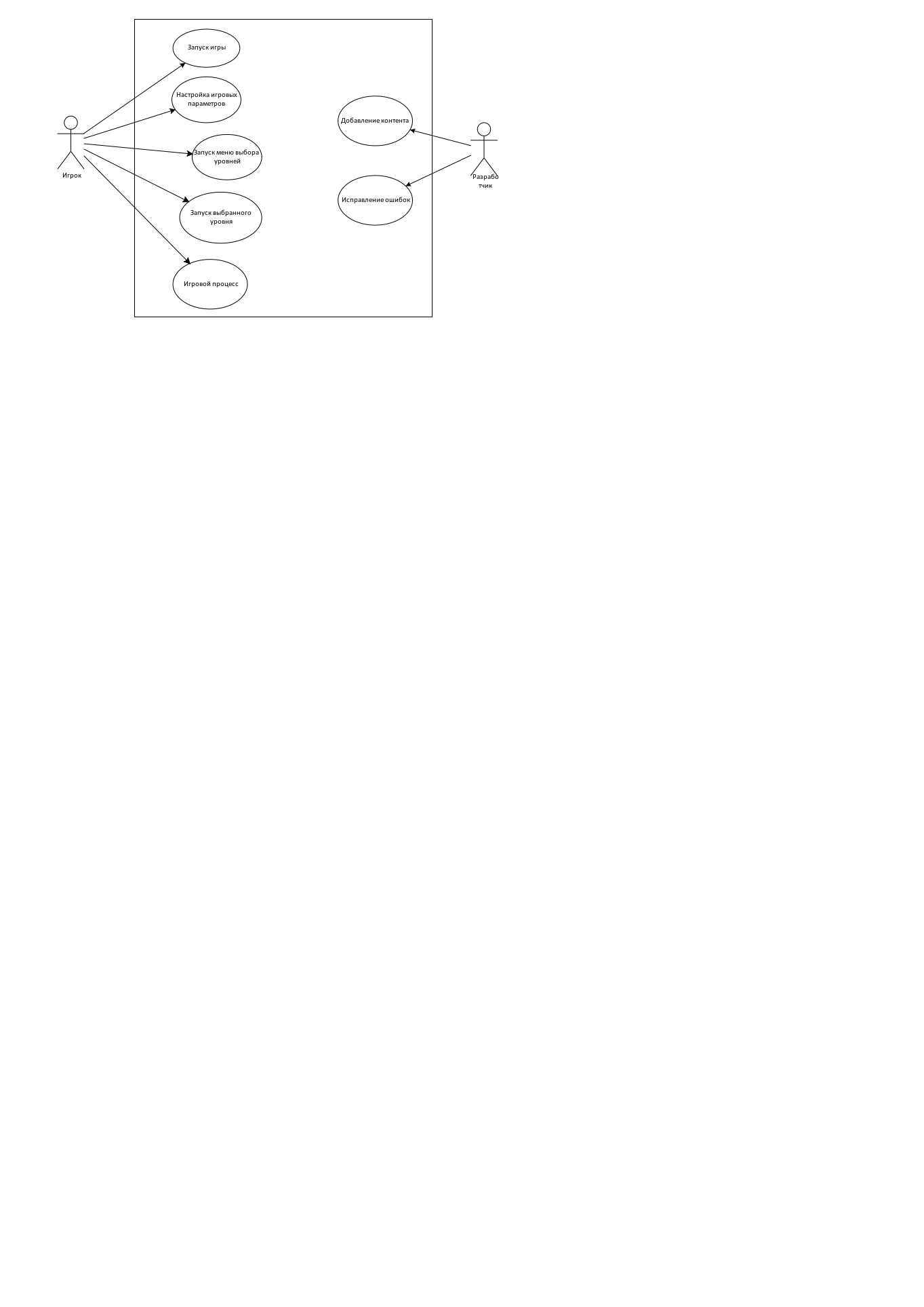


Рис.1. Схема функционала игры.

Важно отметить, что основные этапы проекта, такие как разработка графического оформления и создание игры, проходили одновременно. Это было возможно благодаря тому, что для написания программного кода не требовались готовые анимации или другие элементы, а для проверки работоспособности кода использовались стандартные примитивы Unity. Такой подход значительно ускорял работу. Прежде всего, следует уточнить основные термины, такие как "тайл" и "спрайт". Тайл - это маленький повторяющийся фрагмент, который используется для создания изображений большего размера (тайловая графика) и часто применяется при создании уровней для двухмерных игр. Спрайт - это графический объект, представляющий собой растровое изображение, используемое в компьютерной графике как основная единица для анимации двухмерных объектов. Для создания графики использовался графический редактор PaintTool SAI.

Важно отметить, что основные этапы проекта, такие как разработка графического оформления и создание игры, проходили одновременно. Это было возможно благодаря тому, что для написания программного кода не требовались готовые анимации или другие элементы, а для проверки работоспособности кода использовались стандартные примитивы Unity. Такой подход значительно ускорял работу. Прежде всего, следует уточнить основные термины, такие как "тайл" и "спрайт". Тайл - это маленький повторяющийся фрагмент, который используется для создания изображений большего размера (тайловая графика) и часто применяется при создании уровней для двухмерных игр. Спрайт - это графический объект, представляющий собой растровое изображение, используемое в компьютерной графике как основная единица для анимации двухмерных объектов. Для создания графики использовался графический редактор PaintTool SAI.

## **2.2 Этап разработки игры шрифт**

Управление автомобилем выполняется с использованием руля, педалей газа и тормоза.

Педаль "A" - поворот влево;

Педаль "S" - торможение;

Педаль "D" - поворот вправо;

Педаль "W" - ускорение.

Для разработки вождения были выбраны автомобиль и система управления. Водитель открывает дверь, садится в машину и вставляет ключ в замок зажигания. После запуска двигателя он может использовать руль для изменения направления движения, педаль газа для увеличения скорости и педаль тормоза для замедления или остановки. Система включает в себя множество датчиков и контроллеров, чтобы обеспечивать безопасное и эффективное управление автомобилем.

Unity предоставляет возможность сохранять элементы с добавленными характеристиками и скриптами для их последующего использования. Эти элементы называются префабами и могут быть перемещены на сцену без необходимости создания новых объектов. Для создания префаба необходимо переместить созданный элемент из окна иерархии в окно проекта. Рекомендуется организовать отдельную папку для префабов, чтобы они не потерялись среди других элементов проекта.

Для создания платформы, по которой будут перемещаться персонаж и враги, необходимо добавить спрайт платформы на сцену. Это можно сделать, перетащив спрайт из папки в окно проекта. Затем платформу следует добавить в текущую сцену. Это можно сделать двумя способами: либо перетаскиванием из окна проекта в окно сцены, либо созданием пустого объекта в окне иерархии и применением к нему спрайта платформы. К объекту платформы также следует добавить коллайдер, чтобы определить его границы и взаимодействие с персонажем. В Unity существуют коллайдеры для трехмерных и двумерных объектов, и в данной игре требуется использовать двумерные коллайдеры.

Для создания героя игры также потребуется добавить двумерный объект и применить к нему спрайт героя. Для обеспечения функциональности героя необходим коллайдер и компонент Rigidbody 2D, который управляет физической моделью объектов. Для этого создается скрипт на языке C#, который можно создать в окне проекта или напрямую на объекте в окне инспектора. Когда скрипт создан, Notepad++ откроется, и в коде уже будут подключены основные библиотеки, а также создан стандартный метод Update. В этом скрипте можно написать код для управления движением персонажа. Затем скрипт следует добавить к герою.

В дополнение к коду для движения, необходимо создать анимацию для персонажа. Для этого добавляются атласы с соответствующими анимациями в проект. При выборе файла в Unity, он рассматривается как единый файл, поэтому важно указать, что файл содержит несколько различных объектов, изменяя значение параметра "Sprite mode" на "Multiple" в инспекторе. Затем можно использовать инструмент Slice в Sprite Editor для разделения кадров анимации. Размеры кадров и их центры могут быть настроены, чтобы минимизировать "плавание кадров".

Приведенный ниже код реализует движение игрока в Unity с использованием физического движка Rigidbody2D, анимацию и изменение направления взгляда персонажа в зависимости от направления движения. Пояснение к его работе представлено в приложении А.

Rigidbody2D Rigidbody - переменная для хранения ссылки на компонент Rigidbody2D игрока.

* Animator Animator - переменная для хранения ссылки на компонент Animator игрока.
* SpriteRenderer sprite - переменная для хранения ссылки на компонент SpriteRenderer игрока.
* float directionX = 0f` - переменная для хранения значения горизонтального направления движения игрока.

2. Объявление публичных сериализуемых переменных, которые легко можно будет

настраивать в Unity:

* float moveSpeed = 10f` - переменная для определения скорости движения игрока.
* float jumpForce = 8f` - переменная для определения силы прыжка игрока.

3. Метод Start():

Получение ссылок на компоненты Rigidbody2D, Animator и SpriteRenderer, присваивая их к соответствующим приватным переменным. Установка параметра "running" в состояние false в аниматоре игрока.

4. Метод Update():

Обновление горизонтального направления движения игрока на основе ввода с клавиатуры (Input.GetAxisRaw("Horizontal")). Установка скорости Rigidbody2D игрока с использованием полученного горизонтального направления и скорости движения. Проверка, была ли нажата кнопка "Jump" (пробел) (Input.GetButtonDown("Jump")). Если да, то происходит установка вертикальной скорости Rigidbody2D игрока на значение jumpForce для выполнения прыжка.

5. Метод UpdateAnimationUpdate():

Обновление анимации игрока в зависимости от значения переменной directionX.

* Если directionX больше 0, то устанавливается параметр "running" в состояние true в аниматоре игрока, а также отключается отражение спрайта по горизонтали (sprite.flipX = false).
* Если directionX меньше 0, то устанавливается параметр "running" в состояние true в аниматоре игрока, а также включается отражение спрайта по горизонтали.

Таким образом, персонаж может поворачиваться в сторону своего движения.

Затем следует создать Animator Controller и добавить его к персонажу. Этот контроллер определяет правила и связи для анимаций объекта. Для создания анимаций необходимо создать новый клип анимации в окне Animation, выбрав соответствующий объект, а затем добавить все кадры необходимой анимации в окно в правильном порядке. Здесь можно также настроить скорость анимации. После создания всех клипов анимаций нужно перейти в окно Animator. Здесь следует создать связи и настроить условия переходов между различными анимациями, которые были созданы ранее.

Далее мы создаем второй скрипт, который добавляем к камере, и в нем определяем условия, по которым камера будет следовать за героем . Если мы не сделаем это, то не сможем отслеживать действия персонажа.

Второй код представляет собой скрипт управления камерой в Unity. Код представлен в приложении Б. Вот пошаговое описание его работы:

Подключение необходимых пространств имен:

* System.Collections для работы с коллекциями.
* System.Collections.Generic для использования обобщенных коллекций.
* UnityEngine для доступа к классам и функциональности Unity.

Объявление класса CameraControl, который наследуется от MonoBehaviour, базового класса для скриптов Unity.

Объявление публичной сериализуемой переменной:

Transform player - переменная для хранения ссылки на компонент Transform игрока.

Метод Update():

Обновление позиции камеры, чтобы она следовала за позицией игрока.

Установка позиции камеры (transform.position) равной новому вектору (new Vector3) с координатами player.position.x, player.position.y и текущей позицией по оси Z (transform.position.z).

Таким образом, позиция камеры обновляется, чтобы оставаться на той же высоте и горизонтальном расположении, что и игрок, но с сохранением своей текущей глубины (по оси Z).

После запуска уровня игрок сможет бегать, прыгать и выполнять другие доступные действия. Теперь следует создать сам уровень, разместив на нем различные объекты. Сначала нужно разработать общий план уровня и расположение объектов. Затем можно приступить к реализации самого уровня.

После завершения создания первого уровня разрабатываемого проекта можно проверить его работоспособность прямо в окне Unity. Для этого нужно нажать кнопку запуска проекта в верхней части экрана. Рядом с ней находятся кнопки паузы и окончания теста игры, которые имеют внешний вид кнопок управления в аудио- или видеоплеере.

Технический процесс добавления таких врагов начинается с создания спрайтов или моделей врагов с различными степенями прозрачности. Эти спрайты или модели затем импортируются в Unity, чтобы стать частью игрового мира.

Скрипт управления машиной определяет логику изменения прозрачности в зависимости от определенных условий или событий в игре.

Важно также учитывать производительность игры при использовании изменяемой прозрачности. Если в игре будет большое количество машин,необходимо оптимизировать код и использовать соответствующие методы для уменьшения нагрузки на процессор и графическую подсистему.

Далее, на игровом уровне создаются объекты., которым присваиваются соответствующие компоненты и свойства. К примеру, они могут включать в себя компоненты анимации, коллайдеры и скрипты, определяющие их поведение и движение.

Для создания более интересного игрового опыта можно варьировать характеристики и поведение разных типов быстрых врагов. Некоторые из них могут иметь увеличенную скорость передвижения, другие - способность преследовать игрока, а еще некоторые могут использовать хитрые тактики для избежания столкновений с препятствиями.

При разработке платформера на Unity важным аспектом является создание главного меню, которое позволяет игроку настраивать игровой опыт и выбирать персонажей и уровни.

Вначале необходимо создать сцену для главного меню. Для этого можно использовать пустой объект, который будет служить контейнером для всех элементов меню. Затем в этой сцене следует разместить интерфейсные элементы, такие как кнопки, ползунки, текстовые поля и т.д.

Одним из важных элементов главного меню является настройка игры. Она включает в себя настройку уровня сложности, настройку звука, или других параметров, влияющих на игровой процесс. Для этого используются ползунки или переключатели, которые позволяют игроку выбрать определенные значения или варианты.

Другим важным компонентом главного меню является выбор персонажей. Игроку предоставляется возможность выбрать своего персонажа из предложенного списка. Это было выполнено в виде кнопок и изображений, представляющих каждого персонажа. При выборе персонажа, отображается соответствующая информация о нем, такая как его способности, особенности или характеристики.

Также в главном меню следует предусмотреть выбор уровней. Это было представлено списком уровней с изображениями, представляющими каждый уровень. Игроку предоставляется возможность выбрать уровень, на котором он хочет играть, и приступить к прохождению.

Главное меню игры является важным элементом, который создает комфортную и интуитивно понятную среду для игрока. Оно предоставляет возможность настройки игры под индивидуальные предпочтения, а также выбора персонажей и уровней, что добавляет разнообразие и интерес к игровому процессу.

Добавление таймера в уровень платформера на Unity позволяет создать динамичную и захватывающую игровую механику, где игрок должен завершить уровень в определенное время. Это добавляет элемент соревновательности и стимулирует игрока к быстрому и эффективному прохождению уровня.

Для реализации таймера необходимо создать соответствующий интерфейсный элемент на игровом экране, который будет отображать оставшееся время.

Затем, в скрипте уровня, следует добавить логику таймера. Это сделано путем использования переменных для отслеживания текущего времени и установки времени, отведенного для прохождения уровня. На каждом обновлении игрового цикла необходимо уменьшать значение таймера и обновлять его отображение на экране.

Когда таймер достигает нулевого значения, вызывается функция, которая обрабатывает проигрыш уровня. Дополнительно нужно предусмотреть возможность остановки и возобновления таймера, например, при паузе игры или достижении определенного события.

Добавление таймера в уровень платформера на Unity придает игре дополнительную динамику и вызовет у игрока чувство срочности. Таймер стимулирует игрока к более эффективному и быстрому прохождению уровня, что добавляет дополнительные вызовы и удовлетворение от игрового процесса.

Добавление системы очков в платформер на Unity позволяет создать механику награды, прогрессии и оценки успеха игрока.

Для реализации системы очков необходимо создать переменную, которая будет отслеживать текущее количество очков игрока. Эта переменная установлена в ноль в начале уровня или игры.

Затем, в игровом скрипте, следует добавить логику назначения очков игроку при выполнении определенных действий или достижении целей. Игрок будет получать очки за сбор сердец, уничтожение врагов, прохождение уровней и достижение определенных временных рамок.

При каждом событии, которое должно привести к начислению очков, значение переменной увеличивается и обновляется отображение количества очков на игровом экране.

Разработка функционала предметов молний, способных ускорять игрока на определенное время. Технический процесс добавления предметов молний начинается с создания спрайта молнии. Затем этот спрайт молнии импортируется в проект Unity, чтобы стать доступным для использования в игре.

Далее, на сцене создается пустой игровой объект, к которому применяется спрайт молнии, чтобы обеспечить его отображение на игровом уровне. Для обеспечения взаимодействия с игроком, к объекту молнии добавляется коллайдер. Коллайдер определяет границы объекта и позволяет игроку взаимодействовать с молниями.

При соприкосновении игрока с молнией, активируется соответствующий скрипт, который отвечает за обработку этого взаимодействия. Этот скрипт реализует логику временного ускорения игрока после сбора молнии. Например, он может вызывать специальную функцию, которая увеличивает скорость игрока на заданное значение.

Важным аспектом разработки является реализация логики временного ускорения. Это может быть достигнуто путем вызова функции, которая изменяет скорость игрока на определенное значение в течение определенного времени.

Технический процесс начинается с создания спрайта для потолочных шипов. Затем шипы размещаются на потолке сцены и применяется спрайт к каждому из них.

# ГЛАВА 3 Расчёт экономической эффективности

**3.1 Понятие экономической эффективности (Нумерация подзаголовков так)**

Экономическая эффективность представляет собой показатель успеха, который определяется отношением полученных результатов в сфере человеческой деятельности, производства товаров или предоставления услуг к затратам труда и ресурсов на эти процессы.

Суть экономической эффективности заключается в достижении наилучших результатов при использовании доступных ресурсов компании. Компании активно ищут выгодные материалы и поставщиков, а также применяют различные методы для снижения производственных потерь.

Экономическая эффективность представляет собой теорию без границ, позволяя разрабатывать различные стратегии, направленные на увеличение доходов и сокращение издержек.

Расчет экономической эффективности выполняется с целью понимания того, как правильно используются ресурсы, и для определения перспектив развития предприятия. Существует два типа экономической эффективности проекта: абсолютная и относительная. Абсолютная эффективность используется для общего анализа результатов на различных уровнях - от общенационального до уровня отдельных предприятий и регионов. Относительная эффективность применяется для обоснования организационных, производственных, технических и хозяйственных решений.

Цель каждой компании заключается в обеспечении максимальной прибыли, что достигается путем оптимизации производственных затрат и адаптации к потребностям рынка. Конкуренция играет ключевую роль в распределении продуктов или услуг в соответствии с требованиями рынка, что определяет уровень цен. Для оценки экономической эффективности предприятия применяются различные методы анализа, включая оценку производственной эффективности, финансовую отчетность и анализ рыночных тенденций.

Оценка экономической эффективности предприятия изучает фундаментальные критерии деятельности организаций, способствуя привлечению инвестиций и формированию авторитета и имиджа управленцев. Источники экономической эффективности включают инновации и технологический прогресс, эффективное использование ресурсов, конкуренцию, эффективное управление, инфраструктуру, образование и квалификацию рабочей силы. Затраты на создание системы могут быть определены с помощью метода калькуляций.

Затраты могут быть классифицированы на три основные категории:

1. Постоянные затраты: это расходы, которые возникают в пределах одного производственного цикла и могут быть точно рассчитаны для каждого цикла производства. Продолжительность цикла определяется предприятием.

2. Переменные затраты: это полные расходы, связанные с производством конечного продукта и изменяющиеся в зависимости от объема производства.

3. Общие затраты: это расходы, связанные с определенной стадией производства и нельзя прямо отнести ни к постоянным, ни к переменным затратам.

Для расчета амортизации проекта необходимо умножить стоимость основных средств на норму амортизации и разделить на 100%.

Прибыль составляет от 0 до 41% от полной себестоимости единицы продукта. Полная себестоимость включает в себя как производственные, так и непроизводственные затраты.

Рыночная цена определяется как сумма полной себестоимости, налога на добавленную стоимость и прибыли.

Рентабельность продаж вычисляется как отношение прибыли к полной себестоимости, а рентабельность основных фондов - как отношение прибыли к сумме основных и оборотных фондов.

Основные показатели рентабельности включают в себя рентабельность авансированного капитала и рентабельность собственного капитала. Эти показатели позволяют оценить, сколько рублей прибыли приходится на каждый рубль авансированного или собственного капитала.

Приведенные затраты представляют собой экономическую категорию, отражающую в стоимостном выражении все затраты на производство продукции. Численно приведенные затраты включают в себя текущие затраты, амортизацию и капитальные вложения, соответствующие их нормативу. Приведенные затраты используются для планирования капитальных вложений и принятия решений о размещении производства.

* 1. **Способы расчета экономической эффективности**

**1.2.1 Расчет постоянных издержек**

Срок полезного использования (СПИ) представляет собой временной период, в течение которого компьютерная система может успешно выполнять свои функции или приносить пользу. Часто СПИ используется для определения длительности, в течение которой компьютер ожидается использоваться на производстве. СПИ определяет примерный срок службы до тех пор, пока компьютер не станет устаревшим или не потребуется замена. Например, для настольных компьютеров данный срок составляет обычно 4 года, а для ноутбуков - 1,5 года.

Норма амортизации за месяц - это параметр, определяющий количество средств, которое оценивается как износ или уменьшение стоимости компьютерной системы за один месяц её эксплуатации. Норма амортизации за месяц используется для расчета суммы амортизации, которая должна быть учтена и списана со стоимости актива за определенный период времени. Для этого параметр рассчитывается путем деления первоначальной стоимости компьютерной системы на её СПИ.

Амортизация за период разработки системы считается так

Ап = (Агод/365) \*Т (1)

Так формулы 1 пробел до формулы 1 после

Где, Ап – сумма амортизационных отчислений за период создания системы, руб.;

Т – время эксплуатации компьютера при создании системы.

Затраты на текущий ремонт. Затраты на текущий и профилактический ремонт (Пр) принимаются равными 5% от стоимости ЭВМ. Следовательно, затраты на текущий ремонт за время эксплуатации вычисляются по формуле:

Зтр = (Сбал / 365) \* Пр \* Т

Где Сбал – балансовая стоимость компьютера

Т - количество рабочих дней.

Пр – 5% от стоимости ЭВМ

**1.2.2 Расчет переменных издержек**

Расчет переменных издержек - это процесс определения затрат, которые изменяются в зависимости от изменения объема производства или объема предоставляемых услуг. Эти затраты непосредственно связаны с производством товаров или оказанием услуг и изменяются в соответствии с изменением объема работы.

Расчет переменных издержек является важным инструментом для принятия управленческих решений и оптимизации производственных процессов. Он позволяет оценить, какие затраты будут изменяться в зависимости от изменения объема производства и вычислить их величину.

Примером переменных издержек может служить затраты на сырье и материалы для производства товаров, затраты на рабочую силу, затраты на энергию и транспортные услуги. Все эти затраты могут изменяться в зависимости от объема производства и предоставляемых услуг.

Расчет переменных издержек позволяет более точно оценить стоимость производства и учесть эти издержки при принятии управленческих решений. Например, при принятии решения о расширении производства или введении новой услуги, необходимо учесть изменение переменных издержек и оценить их влияние на общую стоимость производства

Кроме того, расчет переменных издержек позволяет определить точку безубыточности, то есть объем производства или предоставления услуг, при котором компания покрывает все свои переменные издержки и начинает получать прибыль.

Таким образом, расчет переменных издержек является важным инструментом для планирования и управления производственными процессами и помогает компаниям принимать обоснованные управленческие решения.

В данную категорию издержек входят:

1. Затраты на подписку на программы;
2. Затраты на приобретение материалов;
3. Затраты на потребляемую электроэнергию, мобильную связь и интернет;
4. Затраты на оплату труда разработчика;

Стоимость электроэнергии, потребляемой за год, определяется по формуле:

Зэн = Рэвм \* Тэвм \* Сэл

Где, Рэвм – суммарная мощность ЭВМ, кВт

Тэвм – время работы компьютера, часов

Сэл - стоимость одного кВт/ч электроэнергии, руб. э

Расчет основной заработной платы выполняется на основе трудоемкости выполнения каждого этапа, вычисляется по формуле:

Ззп = О \* Тр

Где, О – почасовой оклад работника, руб.

Тр – количество часов, затраченных сотрудником на создание системы

* 1. **Совокупная стоимость решения**

Полная себестоимость разработки игры для АНО ПО «БИТ» определяется суммированием постоянных и переменных издержек и вычисляется по формуле:

Зоб = Зпос + Зпер

Где, Зоб - себестоимость системы АНО ПО «БИТ»

Зпос - постоянные издержки

Зпер - переменные издержки

Эксплуатационные расходы – это сумма текущих затрат на эксплуатацию оборудования, амортизационных отчислений (на реновацию и капитальный ремонт).

Они повторяются в каждом цикле производства и рассчитываются за год. Они осуществляются синхронно с производством и составляют себестоимость продукции или услуг. В состав эксплуатационных расходов на информационную систему входят следующие затраты:

С = Сзп+Сао+Сто+Слс+Сни+Спроч,

Где, Сзп – зарплата управленческого персонала, работающего с использованием программного продукта;

Сао – амортизационные отчисления;

Сто – затраты на техническое обслуживание готового ПО. Обслуживание проводится 1 раз в месяц.

Слс – затраты на интернет;

Сни – затраты на носители информации

Спр – прочие затраты.

**Глава 2 Расчет экономической эффективности**

**2.1 Определение этапов работ**.

Основные этапы работы приведены в таблице 1.

Таблица 1 - Таблица-Определение этапов работы. Так каждую таблицу

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Этап разработки | Время часов | Используемое оборудование | Используемое программное обеспечение | Используемые услуги |
| 1 | 2 | 3 | 4 | 5 |
| Постановка задачи, разработка технического задания, определение подходящих инструментов для разработки игры | 10 | Компьютер | ПО компьютера  LibreOffice Writer,  Mozilla Firefox | Интернет, электроэнергия, мобильная связь |
| Настройка рабочего окружения и системы для разработки игры | 12 | Компьютер | ПО компьютера  Visual Studio Code, Unity Hub, Mozilla Firefox | Интернет, электроэнергия, мобильная связь |

Продолжение таблицы 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Создание прототипа игры | 32 | Компьютер | | ПО компьютера  Visual Studio Code, Unity Hub, Mozilla Firefox | Интернет, электроэнергия, мобильная связь | |
| Написание кода для игры и разработка визуального окружения | 85 | Компьютер | | ПО компьютера  Visual Studio Code, Unity Hub, Mozilla Firefox | Интернет, электроэнергия, мобильная связь | |
| Тестирование | 8 | Компьютер | ПО компьютера  Visual Studio Code, Unity Hub, Mozilla Firefox | | | Интернет, электроэнергия, мобильная связь |
| Редактирование неполадок, выявленных при тестировании | 8 | Компьютер | ПО компьютера  Visual Studio Code, Unity Hub, Mozilla Firefox | | | Интернет, электроэнергия, мобильная связь |

Продолжение таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Выпуск готового продукта | 5 | Компьютер | ПО компьютера  ПО компьютера  Visual Studio Code, Unity Hub, Mozilla Firefox | Интернет, электроэнергия, мобильная связь |
| 160 | | | | |

Все работы выполняет один программист, разработка системы заняла 160 часов или 20 рабочих дня по 8 часа в день.

**2.2 Расчет постоянных издержек**

Стоимость оборудования приведена в таблице 2.

Таблица 2.

Расчет стоимости оборудования.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование | | Цена за единицу, руб. | Стоимость, руб. | |
| 1 | | 2 | 3 | |
| Комплектующая ПК - Процессор Intel Core i5-11400K | | 9 899 | 9 899 | |
| Комплектующая ПК - Материнская плата MSI H520M-A | | 7 299 | 7 299 | |
| Корпус DEXP DC-202M | | 3199 | 3199 | |
| Комплектующая ПК - Видеокарта  [KFA2 GeForce RTX 3060](https://www.dns-shop.ru/product/49b77a8077562eb0/videokarta-kfa2-geforce-rtx-3060-core-lhr-36nol7md1vok/)12 гб | | 31499 | 31499 | |
| Комплектующая ПК - Кулер для процессора ID-COOLING SE-903-SD | | 1 299 | 1 299 | |
| Комплектующая ПК - Оперативная память  [G.SKIll AEGIS](https://www.dns-shop.ru/product/2117ec86c6aded20/operativnaa-pamat-gskill-aegis-f4-3200c16d-16gis-16-gb/) [ГБx2 шт](https://www.dns-shop.ru/product/2117ec86c6aded20/operativnaa-pamat-gskill-aegis-f4-3200c16d-16gis-16-gb/) 16Гб | | 3890 | 3890 | |
| 7 | Комплектующая ПК - 500 ГБ 2.5" SATA накопитель AGI AI238 [AGI500GIMAI238] | 2 399 | | 2 399 |
| 8 | Комплектующая ПК - Блок питания  [Блок питания CHIEFTEC VOITE VPS-700 Вт, 80+ Bronze](https://www.dns-shop.ru/product/098856bb9620ed20/blok-pitania-chieftec-volta-vps-700s-vps-700s/) | 4690 | | 4690 |
| 9 | Комплектующая ПК - Вентилятор DEEPCOOL XFAN 80 [DP-FDC-XF80] | 150 | | 150 |
| 10 | 23.6" Монитор MSI PRO MP241CA черный | 7 999 | | 7 999 |
| 11 | Мышка RAZER ESSENTIAL | 1999 | | 1999 |
| 12 | ATech FBK11- калавиатура | 3100 | | 3100 |
|  |  | Итого | | 77 422 |

Срок полезного использования 36 месяцев. Ликвидационная (остаточная стоимость) – 16 274 руб.

Норма амортизации за месяц = (77422 - 16 274) / 36 = 1 528,7 руб.

Амортизация за период создания системы = (1 528,7\* 12 / 365) \* 20 = 1005руб.

Затраты на текущий ремонт = (77422 / 365) \* 5% \* 20 = 212.28 руб.

Состав постоянных издержек приведён в таблице 3.

Таблица 3.

Состав постоянных издержек.

|  |  |
| --- | --- |
| Вид постоянных издержек | Денежная оценка, руб. |
| 1 | 2 |
| Амортизационные отчисления | 1005 |
| Текущий ремонт техники | 212.28 |
| Итого | 1217,1 |

**2.3 Расчет переменных издержек**

**2.3.1 Расчет затрат на подписки**

Расчет затрат на подписки приведён в таблице 4.

Таблица 4.

Расчет затрат на подписки.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Наименование | Стоимость лицензии, подписки, руб./месяц | Кол-во дней использования | Сумма, руб. |
| 1 | 2 | 3 | 4 | 5 |
| 1 | LibreOffice Writer | Бесплатно | 6 | 0 |
| 2 | LibreOffice Draw | Бесплатно | 8 | 0 |
| 3 | Mozilla Firefox | Бесплатно | 14 | 0 |
| 4 | Unity Hub | Бесплатно | 23 | 0 |
| 5 | Visual Studio Code | Бесплатно | 10 | 0 |
| Итого | | | | 0 |

**2.3.2. Расчет затрат на потребление электроэнергии, услуги мобильной связи и интернета.**

Стоимость электроэнергии за период работы компьютера во время создания системы = 0,250 \* 160 \* 4,62 = 184,8 руб.

Расчет на затраты услуги связи и интернета представлены в таблице 5.

Таблица 5.

Расчет затрат на услуги связи и интернета.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № п/п | Наименование | Тариф | Кол-во дней использования | Стоимость одного дня, руб. | Стоимость, руб. |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | Интернет | 400 руб./месяц | 20 | 13 | 260 |
| 2 | Услуги связи | 350 руб./месяц | 20 | 12,7 | 240 |
| Итого | | | | | 500 |

**2.3.3 Расчет заработной платы разработчика**

Расчет заработной платы сотрудника представлен в таблице 6.

Таблица 6.

Расчет заработной платы сотрудника.

|  |  |  |  |
| --- | --- | --- | --- |
| Должность | Часовая тарифная ставка, руб. | Количество часов, затраченных на создание системы | Сумма, руб. |
| 1 | 2 | 4 | 5 |
| Разработчик | 800 | 160 | 112800 |
|  |  |  |  |

**2.3.4 Общая сумма переменных издержек**

Расчет переменных издержек представлен в таблице 7.

Таблица 7.

Расчет переменных издержек при разработке игры для АНО ПО «БИТ».

|  |  |
| --- | --- |
| Вид переменных издержек | Величина, руб. |
| 1 | 2 |
| Затраты на подписку на программы | 0 |
| Затраты на мобильную связь | 240 |
| Затраты на Интернет | 260 |
| Затраты на потребляемую электроэнергию | 184,8 |
| Затраты на оплату труда разработчику | 112800 |
| Итого | 113484,8 |

**2.4 Совокупная стоимость решения**

Структура полных издержек, представленная в таблице 8.

Таблица 8.

Структура полных издержек при разработке системы.

|  |  |
| --- | --- |
| Вид издержки | Величина, руб. |
| 1 | 2 |
| Постоянные | 1005 |
| Переменные | 113484,8 |
| Итого | 114489,8 |

Эксплуатационные затраты приведены в таблице 9.

Таблица 9.

Расчет эксплуатационных расходов.

|  |  |  |
| --- | --- | --- |
| Вид затрат | Состав затрат | Планируемая сумма  (срок 1 год), руб. |
| 1 | 2 | 3 |
| Техническое обслуживание | Техобслуживание (1 000 руб./мес.) | 12 000 |
| Затраты на интернет | Стоимость услуги за доступ в Интернет | 4 200 |
| Прочие затраты | 2% | 634,92 |
|  | Итого | 16834,92 |

Совокупная стоимость решения приведена в таблице 10.

Таблица 10.

Расчет совокупной стоимости решения.

|  |  |
| --- | --- |
| Вид затрат | Сумма, руб. |
| 1 | 2 |
| Полные издержки | 114489,8 |
| Эксплуатационные затраты | 16834,92 |
| Итого | 131324,72 |

**2.5. Расчет экономической эффективности**

Экономическая эффективность решения приведена в таблице 11.

Таблица 11.

Экономическая эффективность решения.

|  |  |  |  |
| --- | --- | --- | --- |
| Факторы | До внедрения системы | После внедрения системы | Финансовый эффект |
| 1 | 2 | 3 | 4 |
| Время на передачу информации | 2.5 часа/неделю | 0.5 часов/неделю | 2 часа/неделю |
| Траты на бумажные носители | 15 000 руб./месяц | 2500 руб./месяц | 12500 руб. |

**•** Улучшение коммуникации - Внедрение нового приложения может улучшить коммуникацию между преподавателями, упростить обмен информацией и облегчить совместную работу. Это может снизить время, затрачиваемое на поиск и получение необходимой информации, и ускорить принятие решений. Это может привести к более эффективной работе и улучшению учебного процесса.

• Оптимизация рабочих процессов - Новое приложение может автоматизировать и оптимизировать определенные рабочие процессы, что может сократить время, затрачиваемое на выполнение задач и улучшить эффективность работы студентом. Например, новое приложение может автоматически генерировать отчеты или формы, что позволит студентам избежать ручного ввода данных и сократить количество ошибок.

• Увеличение доступности информации - Новое приложение может обеспечить более широкий и удобный доступ к информации для студентов, что может повысить их эффективность и способствовать более быстрому принятию решений. Например, студенты смогут получить доступ к необходимым данным или документам из любого места и в любое время

• Сокращение административных задач - Новое приложение может автоматизировать и упростить выполнение административных задач, таких как управление проектами, планирование встреч или отслеживание рабочего времени. Это может освободить время студентов и позволить им сконцентрироваться на более важных задачах, что может повысить их производительность и качество работы.

• Сокращение затрат на обучение - Новое приложение может быть легким в освоении и использовании, что может сократить необходимость в длительном и сложном обучении студентов. Это может сэкономить время и деньги, затрачиваемые на обучение, и позволить студентом быстрее начать использовать приложение в своей работе.

Учет этих факторов при оценке экономической эффективности внедрения нового приложения позволяет более точно оценить его влияние на операционные расходы, производительность и эффективность работы организации.

# ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы был выполнен проект разработки игрового приложения в среде Unity 3D. Целью данного проекта было создание увлекательной и качественной игры «Пдд», которая предлагает игрокам интересный игровой опыт и вызывает положительные эмоции.

В процессе разработки были достигнуты следующие цели:

1. Изучены особенности и состояние компьютерной индустрии России;
2. Выбран жанр, вид и платформу для компьютерной игры;
3. Разработаны концепции основных элементов;
4. Выбрано и изучено средство реализации;
5. Подготовлены необходимые для игры материалы;
6. Реализация основной механики движения персонажа.
7. Разработка системы управления и пользовательского интерфейса.
8. Разработан прототип игры.

В результате выполненной работы был создан полноценный 2D платформер с увлекательным геймплеем, качественной графикой и звуковым сопровождением. Проект соответствует поставленным целям.

В ходе разработки были получены ценные навыки и опыт работы с Unity, включая программирование, создание анимаций, управление ресурсами. Эти навыки и опыт будут полезны как в сфере разработки приложений, так и в сфере разработки игр

Итоговая разработка 2D платформера в Unity является успешной и положительной, и может служить примером качественной работы в области создания игровых проектов.

В процессе выполнения выпускной квалификационной работы был проведен расчёт стоимости разработки игры «Пдд». Также рассчитана экономическая эффективность разработки игры. Полностью игра окупится примерно через 6 месяцев.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

*Федеральные законы и нормативные документы:*

1. НК РФ Статья 258. Амортизационные группы (подгруппы). Особенности включения амортизируемого имущества в состав амортизационных групп (подгрупп).

2. Приказ МИНФИНА РОССИИ от 17.09.2020 № 204н «Об утверждении Федеральных стандартов бухгалтерского учета ФСБУ 6/2020 "Основные средства" и ФСБУ 26/2020 "Капитальные вложения".

3. Федеральный закон от 07.02.1992 N 2300 – 1 (ред. от 03.07.2016) «О защите прав потребителей» // Собрание законодательства РФ, 15.01.1996, N 3, 140 с.

*Основная литература:*

4. Джереми Гибсон. Introduction to Game Design, Prototyping, and Development. Addison-Wesley Professional, 2022 г. – 1296 с.

5. Джон Маннинг, Парис Батфилд-Аддисон, Тим Нугент. Unity Game Development Cookbook: Essentials for Every Game. O’Reilly Media, 2018 г. – 408 с.

6. Джо Хокинг. Unity in Action: Multiplatform Game Development in C#. Manning, 2019 г. - 416 с.

7. Патрик Фелисия. Unity from Zero to Proficiency (Foundations). Independently published, 2019 г. – 240 с.

*Дополнительная литература:*

10. Липсиц И.В. Экономика. Омег, 2019 г. – 182 с.

11. Марк Рид. C# 3 Books in 1. Independently published, 2022 г. – 344 с.

12. Трейси Фуллертон. Game Design Workshop. AK Peters, 2019 г. – 386 с.

8. Джесси Шелл. Геймдизайн. Альпина Паблишер, 2019 г. – 657 с.

9. Крейнина М.Н. Финансовый анализ. Москва: Дело и сервис, 2020 г. – 158 с.

*Интернет- ресурсы:*

13. Производственный календарь на 2023 год [Электронный ресурс] URL: [https://www.consultant.ru/law/ref/calendar/proizvodstvennye/2023](https://www.consultant.ru/law/ref/calendar/proizvodstvennye/2023/) (Дата обращения 11.03.2023)

14. How to Build a Complete 2D Platformer in Unity [Электронный ресурс]. URL: https://gamedevacademy.org/how-to-build-a-complete-2d-platformer-in-unity (Дата обращения 17.03.2023)

15. Unity Documentation [Электронный ресурс]. URL: <https://docs.unity3d.com/Manual/index.html> (Дата обращения 15.03.2023)