TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа: Инженерная школа информационных технологий и робототехники
Направление подготовки: 09.03.04 «Программная инженерия»
Отделение школы (НОЦ): Отделение информационных технологий

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Разработка компьютерной игры в жанре "Шутер в виртуальной реальности" на базе игрового движка Unity

УДК: 004.925.84:004.946

Студент

Группа	ФИО	Подпись	Дата
8К82	Крымов Андрей Алексеевич		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Чердынцев Евгений Сергеевич	к.т.н.		

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор	Гасанов Магеррам Али оглы	д.э.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Мезенцева Ирина Леонидовна			

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Чердынцев Евгений Сергеевич	к.т.н.		

Томск – 2022г.

Планируемые результаты освоения ООП по направлению 09.03.04 «Программная инженерия»

Код компетенции	Наименование компетенции
Универсальные компетенции	
УК(У)-1	Способность осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач
УК(У)-2	Способность определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений
УК(У)-3	Способность осуществлять социальное взаимодействие и реализовывать свою роль в команде
УК(У)-4	Способность осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(-ых) языке(-ах)
УК(У)-5	Способность воспринимать межкультурное разнообразие общества в социально-историческом, этическом и философском контекстах
УК(У)-6	Способность управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни
УК(У)-7	Способность поддерживать должный уровень физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности
УК(У)-8	Способность создавать и поддерживать безопасные условия жизнедеятельности, в том числе при возникновении чрезвычайных ситуаций
УК(У)-9	Способность проявлять предприимчивость в профессиональной деятельности, в т.ч. в рамках разработки коммерчески-перспективного продукта на основе научно-технической идеи
Общепрофессиональные компетенции	
ОПК(У)-1	Способность использовать базовые знания естественных наук, математики и информатики, основные факты, концепции, принципы теорий, связанных с прикладной математикой и информатикой
ОПК(У)-2	Способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию образовательного контента, прикладных баз данных
ОПК(У)-3	Способность приобретать новые научные и профессиональные знания, используя современные образовательные и информационные технологии
ОПК(У)-4	Способность понимать сущность и значение информации в развитии современного общества, осознавать опасность и угрозу, возникающие в этом процессе, соблюдать основные требования информационной безопасности
ОПК(У)-5	Способность использовать основные методы, способы и средства получения, хранения, переработки информации и навыки работы с компьютером как со средством управления информацией
ОПК(У)-6	Способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической

	культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности
ОПК(У)-7	Способность использовать в своей профессиональной деятельности знание иностранного языка
ОПК(У)-8	Способность критически переосмысливать накопленный опыт, изменять при необходимости направление своей деятельности
ОПК(У)-9	Способность получить организационно-управленческие навыки при работе в научных группах и других малых коллективах исполнителей
Профессиональные компетенции	
ПК(У)-1	Способность работать в составе научно-исследовательского и производственного коллектива и решать задачи профессиональной деятельности
ПК(У)-2	Способность к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения
ПК(У)-3	Способность осуществлять целенаправленный поиск информации о новейших научных и технологических достижениях в информационно-телекоммуникационной сети "Интернет" и в других источниках
ПК(У)-4	Способность собирать, обрабатывать и интерпретировать данные современных научных исследований, необходимые для формирования выводов по соответствующим научным исследованиям
ПК(У)-5	Способность понимать, совершенствовать и применять современный математический аппарат
ПК(У)-6	Способность приобретать и использовать организационно-управленческие навыки в профессиональной и социальной деятельности
ПК(У)-7	Способность составлять и контролировать план выполняемой работы, планировать необходимые для выполнения работы ресурсы, оценивать результаты собственной работы
ПК(У)-8	Способность к реализации решений, направленных на поддержку социально-значимых проектов, на повышение информационной грамотности населения, обеспечения общедоступности информационных услуг
ПК(У)-9	Способность к организации педагогической деятельности в конкретной предметной области (прикладная математика и информатика)

TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа: Инженерная школа информационных технологий и робототехники
Направление подготовки (специальность): 09.03.04 «Программная инженерия»
Отделение школы (НОЦ): Отделение информационных технологий

УТВЕРЖДАЮ:
Руководитель ООП
_____ _____ Чердынцев Е.С.
(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Бакалаврской работы

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
8К82	Крымову Андрею Алексеевичу

Тема работы:

Разработка компьютерной игры в жанре "Шутер в виртуальной реальности" на базе игрового движка Unity	
Утверждена приказом директора (дата, номер)	№40-50/с от 09.02.2022 г.

Срок сдачи студентом выполненной работы:	10.06.2022 г.

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе *(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).*	Работа направлена на создание игры для платформы Android

Перечень подлежащих исследованию, проектированию и разработке вопросов *(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).*	Изучение рынка видеоигр, платформ, жанров, средств разработки. Проектирование и создание игрового прототипа
Перечень графического материала *(с точным указанием обязательных чертежей)*	UML-диаграммы, описывающие проектируемую игру, интерфейс разработанного приложения

Консультанты по разделам выпускной квалификационной работы
(с указанием разделов)

Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Гасанов Магеррам Али оглы
Социальная ответственность	Мезенцева Ирина Леонидовна

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	01.02.2022 г.

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Чердынцев Евгений Сергеевич	к.т.н		01.02.2022

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К82	Крымов Андрей Алексеевич		01.02.2022

ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
8К82	Крымову Андрею Алексеевичу

Школа	**ИШИТР**	**Отделение школы (НОЦ)**	**Отделение контроля и диагностики**
Уровень образования	Бакалавриат	**Направление/специальность**	Программная инженерия

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:	
1. *Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих*	*Оклад руководителя – 30000 руб.* *Оклад студента –4030 руб.*
2. *Нормы и нормативы расходования ресурсов*	*Премиальный коэффициент руководителя 30%;* *Премиальный коэффициент инженера 20%;* *Доплаты и надбавки руководителя 30%;* *Доплаты и надбавки руководителя 30%;* *Дополнительной заработной платы 12%;* *Накладные расходы 16%;* *Районный коэффициент 1,3%.*
3. *Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования*	*Коэффициент отчислений на уплату во внебюджетные фонды 30,2 %*
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. *Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения*	*Определение потенциального потребителя результатов исследования, SWOT-анализ разработанной стратегии*
2. *Планирование и формирование бюджета научных исследований*	*Определение структуры работы. Расчет трудоемкости выполнения работ. Подсчет бюджета исследования*
3. *Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования*	*Рассчитать показатели финансовой эффективности, ресурсоэффективности и эффективности исполнения*
Перечень графического материала *(с точным указанием обязательных чертежей)***:**	

1. *Оценка конкурентоспособности технических решений*
2. *Матрица SWOT*
3. *Альтернативы проведения НИ*
4. *График проведения и бюджет НИ*
5. *Оценка ресурсной, финансовой и экономической эффективности НИ*

Дата выдачи задания для раздела по линейному графику	03.03.2022

Задание выдал консультант:

Должность	**ФИО**	**Ученая степень, звание**	**Подпись**	**Дата**
Профессор	Гасанов Магеррам Али оглы	д.э.н.		03.03.2022

Задание принял к исполнению студент:

Группа	**ФИО**	**Подпись**	**Дата**
8К82	Крымов Андрей Алексеевич		03.03.2022

ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8К82	Крымов Андрей Алексеевич

Школа	**ИШИТР**	**Отделение школы (НОЦ)**	**Отделение контроля и диагностики**
Уровень образования	Бакалавриат	**Направление/специальность**	09.03.04 Программная инженерия

Тема ВКР:

Разработка компьютерной игры в жанре "Шутер в виртуальной реальности" на базе игрового движка Unity	
Исходные данные к разделу «Социальная ответственность»:	
Введение • Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика) и области его применения. • Описание рабочей зоны (рабочего места) при разработке проектного решения/при эксплуатации	*Объект исследования __Игровой движок Unity __* *Область применения _Информационные технологии, разработка видеоигр__* *Рабочая зона: __Комната, учебная аудитория__* *Размеры помещения:_____7*6 м._____* *Количество и наименование оборудования рабочей зоны ___Персональный компьютер с установленным Unity и периферия ___* *Рабочие процессы, связанные с объектом исследования, осуществляющиеся в рабочей зоне Разработка основных механик игры, создание и поиск графики, проектирование игровых уровней, создание пользовательского интерфейса, тестирование игры, в том числе и с помощью других людей, сбор информации по улучшению игрового процесса. Необязательно: подключение рекламных API, выпуск игры в Play Market.*
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Правовые и организационные вопросы обеспечения безопасности при разработке проектного решения: • специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; • организационные мероприятия при компоновке рабочей зоны.	1. ГОСТ 12.2.032-78 2. ГОСТ 21889-76 3. ГОСТ 22269-76 4. Конституция Российской Федерации 5. Трудовой кодекс РФ 6. СанПиН 1.2.3685-21

2. Производственная безопасность при <u>разработке проектного решения</u>: • Анализ выявленных вредных и опасных производственных факторов	Вредные и опасные производственные факторы в соответствии с ГОСТ 12.0.003-2015: • производственные факторы, связанные с электро-магнитными излучениями; • отсутствие или недостаток необходимого искусственного освещения; • повышенный уровень шума • монотонность труда, вызывающая монотонию • производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которого попадает работающий; • длительное сосредоточенное наблюдение Требуемые средства коллективной и индивидуальной защиты от выявленных факторов: • удобное посадочное место • источник искусственного освещения • изоляция электроники от попадания влаги, пыли
3. Экологическая безопасность при <u>разработке проектного решения:</u>	Воздействие на селитебную зону Нет Воздействие на литосферу Нет Воздействие на гидросферу Нет Воздействие на атмосферу Нагрев аппаратных средств, потребление электроэнергии
4. Безопасность в чрезвычайных ситуациях при <u>разработке проектного решения</u>	Возможные ЧС: • пожар на рабочем месте • землетрясение Наиболее типичная ЧС • пожар на рабочем месте
Дата выдачи задания для раздела по линейному графику	02.02.2022

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Мезенцева Ирина Леонидовна			02.02.2022

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К82	Крымов Андрей Алексеевич		02.02.2022

TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа: Инженерная школа информационных технологий и робототехники
Направление подготовки (специальность): 09.03.04 «Программная инженерия»
Уровень образования: Бакалавр
Отделение школы (НОЦ): Отделение информационных технологий
Период выполнения: осенний / весенний семестр 2021 / 2022 учебного года

Форма представления работы:

Бакалаврская работа
(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	10.06.2022 г.

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
25.05.2022 г.	Обзор предметной области	20
25.05.2022 г.	Проектирование игрового процесса	25
25.05.2022 г.	Реализация игрового процесса	25
25.05.2022 г.	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	15
25.05.2022 г.	Социальная ответственность	10

СОСТАВИЛ:
Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Чердынцев Евгений Сергеевич	к.т.н.		

СОГЛАСОВАНО:
Руководитель ООП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Чердынцев Евгений Сергеевич	к.т.н.		

ОГЛАВЛЕНИЕ

РЕФЕРАТ 14
ВВЕДЕНИЕ 15
ГЛАВА 1. Обзор предметной области 17
1.1 Общая информация 17
1.2 Анализ жанров 18
1.2.1 Казуал 20
1.2.2 Аркада 20
1.2.3 Головоломка 21
1.2.4 Экшен 21
1.2.5 Гонка 21
1.3 Сравнение игровых движков 22
1.3.1 Свободно распространяемые движки 23
1.3.1.1 Unity 23
1.3.1.2 Godot 24
1.3.1.3 Unreal Engine 4 25
1.4 Выводы по главе 26
ГЛАВА 2. Проектирование игрового процесса 26
2.1 Описание игрового процесса 27
2.1.1 Ключевая игровая механика 27
2.1.2 Возможные геймплейные механики 27
2.1.2.1 Подбор бонусов 28
2.1.2.2 Добавление таймера 28
2.1.2.3 Внутриигровая валюта 28
2.1.2.4 Система заданий 28
2.1.2.5 Кастомизация 28
2.1.2.6 Второй шанс 29
2.1.3 Возможные сложности в реализации геймплейных механик 29
2.2 Архитектура игры 29

2.3 Диаграмма состояний 31
2.4 Выводы по главе 32
ГЛАВА 3. Реализация игрового процесса 32
3.1 Структура проекта 33
3.2 Объект Green Car 33
3.3 Объект Control Button Controller 36
3.4 Объект Points 36
3.5 Объект Отметка 38
3.6 Реализация пользовательского интерфейса 38
3.7 Выводы по главе 38
ГЛАВА 4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение 39
4.1 Анализ конкурентных технических решений 40
4.2. SWOT-анализ 42
4.3. Планирование работ по научно-техническому исследованию 45
4.3.1. Структура работ в рамках научного исследования 46
4.3.2. Определение трудоемкости выполнения работ 47
4.3.3. Разработка графика проведения научного исследования 48
4.4. Бюджет технического проекта 50
4.4.1. Расчет материальных затрат 50
4.4.2. Расчет затрат на оборудование 51
4.4.3. Основная заработная плата исполнителей 52
4.4.4. Расчет дополнительной заработной платы 53
4.4.5 Отчисления во внебюджетные фонды 54
4.4.6. Накладные расходы 54
4.4.7 Формирование бюджета затрат проекта 55
4.5. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования 55
Глава 5. Социальная ответственность 57
Введение 57

5.1. Правовые и организационные вопросы обеспечения безопасности 58
5.2. Производственная безопасность 60
5.2.1.Анализ опасных и вредных факторов и обоснование мероприятий по снижению их
5.2.1.1. Производственные факторы, связанные с электромагнитными излучениями 61
5.2.1.2. Отсутствие или недостаток необходимого искусственного освещения 61
5.2.1.3. Повышенный уровень шума 62
5.2.1.4. Монотонность труда 63
5.2.1.5. Производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которого попадает работающий 63
5.2.1.6. Длительное сосредоточенное наблюдение 65
5.3. Экологическая безопасность 65
5.4. Безопасность в чрезвычайных ситуациях 66
Вывод по главе 67
ЗАКЛЮЧЕНИЕ 68
СПИСОК ЛИТЕРАТУРЫ 69

РЕФЕРАТ

Дипломная работа содержит: 88 страниц, 13 рисунков, 18 таблиц, 12 источников.

Ключевые слова:

1. Android – это операционная система для смартфонов, планшетов, электронных книг, цифровых проигрывателей, наручных часов, и т.п.
2. C# – объектно-ориентированный язык программирования, разработаный группой инженеров компании Microsoft как язык разработки приложений для платформы Microsoft .NET Framework.
3. GameObjects — это основные объекты в Unity, которые представляют все объекты. Представляют собой контейнеры для компонентов, реализующих реальную функциональность.
4. UML — язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур.
5. Архитектура — совокупность решений об организации программной системы. Включает выбор структурных элементов и их интерфейсов, с помощью которых составлена система, а также их поведения в рамках сотрудничества структурных элементов.
6. Ассет – это цифровой объект, преимущественно состоящий из однотипных данных, неделимая сущность, которая представляет часть игрового контента и обладает некими свойствами.
7. Видеоигра – это компьютерная программа, служащая для организации игрового процесса, связи с партнёрами по игре, или сама выступающая в качестве партнёра.
8. Геймплей – компонент игры, отвечающий за взаимодействие игры и игрока. Он описывает, как игрок взаимодействует с игровым миром, как игровой мир реагирует на действия игрока и как определяется набор действий, который предлагает игроку игра.

9. Движок – это объединенный в единое целое комплекс прикладных программ, с помощью которых обеспечивается графическая визуализация, звуковое сопровождение, логика внутриигровых объектов в соответствии со скриптами, а также игра по сети, встроенные графические сцены, соблюдение физических эффектов и законов.

10. Коллайдер – это невидимая упрощённая форма объекта, привязанная к нему, для расчёта столкновений с другими объектами.

11. Коллизия – это столкновение коллайдеров объектов в пространстве внутри игры.

12. Компонент – функциональная часть каждого GameObject. Они определяют поведение всех объектов в игре.

13. Консоль – это специализированное электронное устройство, предназначенное для видеоигр; для таких устройств, в отличие от персональных компьютеров, запуск и воспроизведение видеоигр является основной задачей.

14. Механика – это набор правил и способов, реализующий определённым образом некоторую часть интерактивного взаимодействия игрока и игры.

15. Платформа – компьютерная система, на которой запускается игра.

16. Префаб – это особый тип ассетов, позволяющий хранить весь GameObject со всеми компонентами и значениями свойств. Он выступает в роли шаблона для создания экземпляров хранимого объекта в сцене.

17. Прототип – это демоверсия с базовым функционалом и геймплеем, создаваемый с целью проверить и отработать все идеи, оценить их перспективность и снизить риск разработки неиграбельного проекта.

18. Скрипт – это небольшая программа, заточенная под определенное действие, представляющая собой последовательность команд для выполнения конкретных операций.

Объектом исследования является технологии разработки игр для платформы Android с помощью среды разработки «Unity».

Цель работы – разработка игры в жанре «Аркада» для платформы Android.

ВВЕДЕНИЕ

В современном мире видеоигры являются одним из наиболее крупных сегментов индустрии развлечений. Масштабы игровой индустрии сопоставимы, например, с производством кино. А по скорости роста за последние пять лет – существенно ее опережала. По степени влияния на потребителей и вовлеченности их в интерактивное окружение, предлагаемое видеоиграми, этот сегмент уже давно выделяется среди других видов развлечений.

Эксперты и участники отрасли отмечают развитие ряда сильных тенденций, которые определяют развитие глобальной игровой индустрии. Так, большую роль продолжают играть новые технологии (виртуальные, мобильные, облачные и др.), наблюдается слияние виртуальных и физических сред, социализация игр и т.д.

В настоящий момент мобильные игры – самый быстрорастущий сегмент игрового рынка с самой большой долей среди всех сегментов. Это объясняется растущей доступностью мобильных устройств и мобильного трафика, а также новыми возможностями для разработчиков, ввиду большого количества игровых движков для разных целей.

Цель работы – разработка игры в жанре «Аркада» для платформы Android.

Для достижения цели предстоит выполнить ряд поставленных задач:

1. Провести анализ рынка платформ, а также наиболее популярных жанров.
2. Описать геймплейный процесс, определить первостепенные и второстепенные механики, спроектировать разрабатываемую систему.
3. Разработать прототип с основными механиками, создать альфа-версию игры.
4. Провести внутреннее тестирование, выявить на его основе основные замечания и пожелания к геймплею.

ГЛАВА 1. Обзор предметной области

1.1 Общая информация

За последние пять лет рынок видеоигр продемонстрировал существенный рост в сравнении с другими сегментами индустрии развлечений. В 2020 году, пока на фоне пандемии многие сегменты рынка развлечений терпели убытки, видеоигры испытали небывалый подъём, который продолжается до сих пор, и согласно прогнозу портала «Marketwatch» (рисунок 1), будет продолжаться.

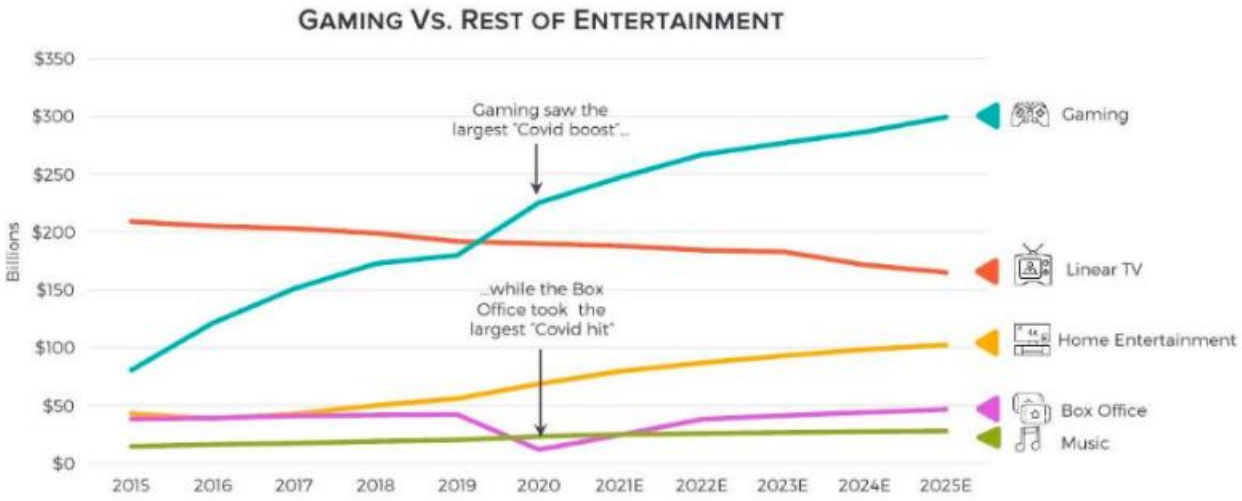

Рисунок 1. Доходы различных сегментов индустрии развлечений

Всю индустрию видеоигр можно разделить по платформам. Основными сегментами являются игры для консолей, персональных компьютеров, которые включают в себя непосредственно скачиваемые и браузерные игры, а также игры для мобильных платформ, которые также можно подразделить на смартфоны и планшеты. Согласно анализу, проведенному порталом «GamesIndustry.Biz» (рисунок 2), мобильные платформы занимают больше половины всего рынка видеоигр, и за 2021 год общая выручка мобильных игр составила более 92 миллиардов долларов, что на 7,3 % больше, чем в прошлом году. Также можно заметить, что несмотря на рост выручки для мобильных платформ, выручка с доли персональных компьютеров и консолей в сравнении с прошлым годом снизилась. И если снижение заработка с игр для персональных компьютеров в основном обусловлено все более непопулярными браузерными играми, то

доходы консолей упали на существенные 6,6 %. Хотя и обычные игры для персональных компьютеров не продемонстрировали существенного роста. Вероятнее всего, данные значения обусловлены дефицитом микросхем, используемых в производстве компьютеров и консолей, следовательно, ввиду отсутствия подходящих платформ спрос на игры для них падает. Большинство игр для мобильных устройств же не столь требовательны к ресурсам. А учитывая, что мобильные устройства становятся все более доступными и производительными, подъем сегмента мобильных видеоигр становится вполне логичным.

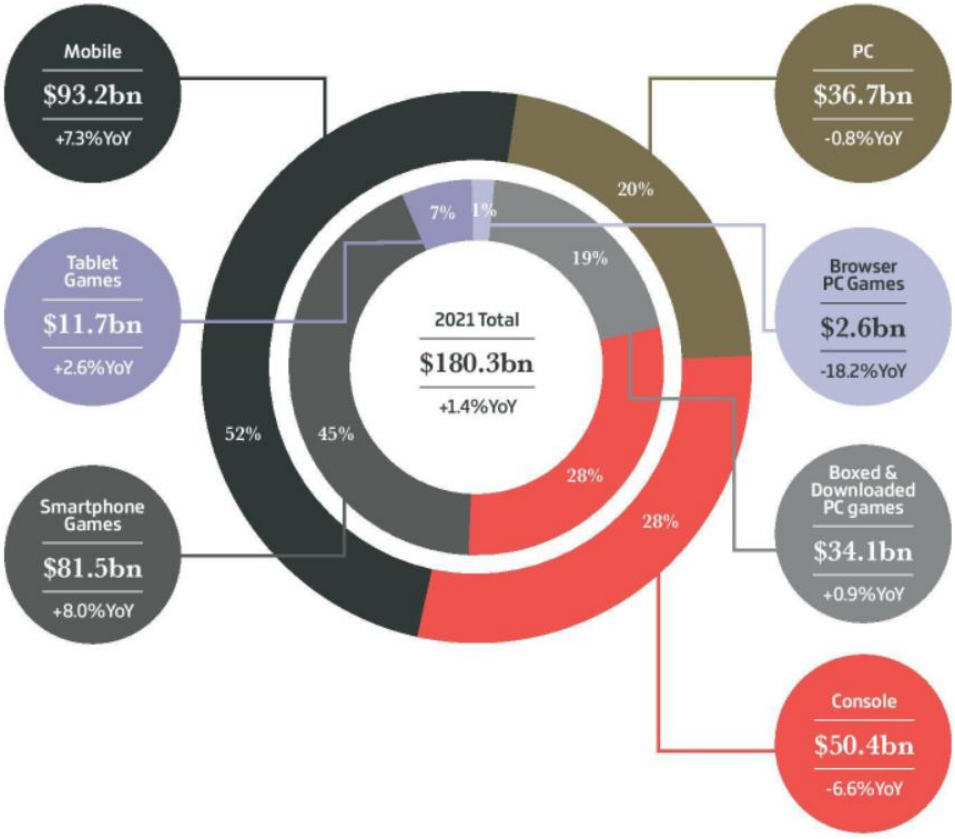

Рисунок 2. Сегментация рынка видеоигр по платформам

1.2 Анализ жанров

Сервис по отслеживанию эффективности маркетинга «SocialPeta» провел анализ рынка рекламы в мобильных играх за 2021 год. На графике (рисунок 3) приведено распределение числа рекламодателей в зависимости от жанра. Согласно данному графику, лидером в этом критерии является жанр казуальных

игр, занимающий первое место с двукратным отрывом. Также в рейтинге можно отметить жанры «аркада» и «головоломка».

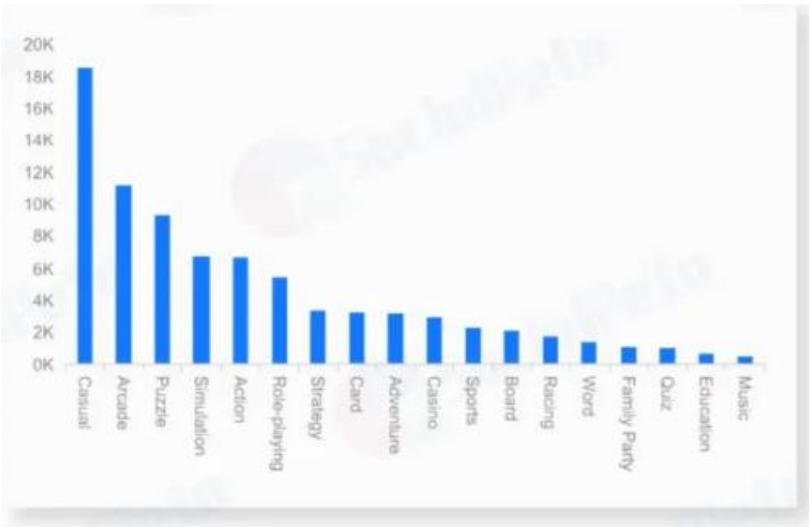

Рисунок 3. Распределение числа рекламодателей по жанрам в мобильных играх

Аналитики из «Statista» провели собственное исследование, в основе которого лежало определение наиболее популярных жанров путем отслеживания количества скачиваний в «Google Play» (рисунок 4). Наиболее популярными жанрами также являются «казуал», «аркада» и «головоломка», показатель проникновения на мобильные устройства которых составляет в среднем 57%. За ними следуют с куда меньшим, но достаточно высоким, процентом жанры «экшен» и «гонки», средний уровень проникновения которых составляет 32%.

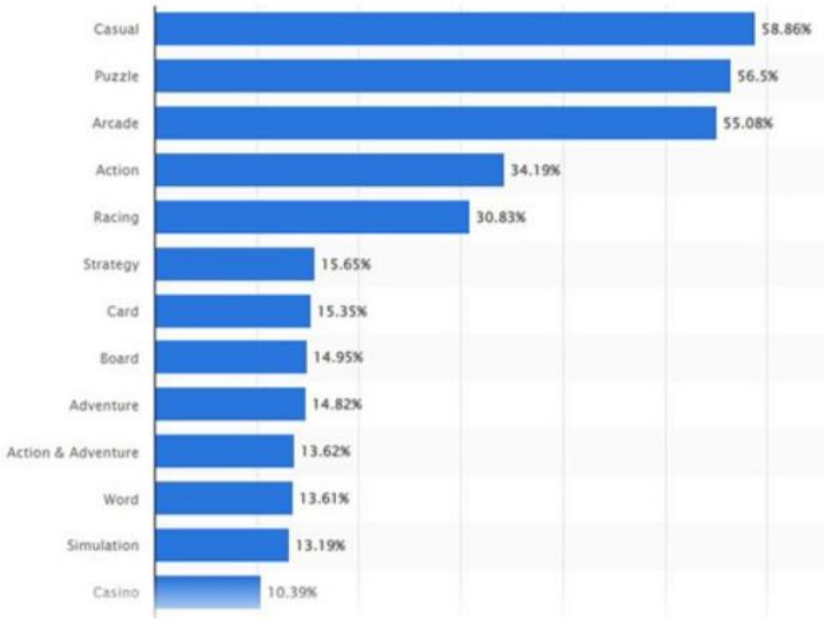

Рисунок 4. Распределение уровня проникновения по жанрам в мобильных играх

Согласно проведенному анализу популярности жанров мобильных игр, были выбраны пять наиболее популярных, в одном из которых и будет разрабатываться игра.

1.2.1 Казуал

Казуал — это жанр компьютерных игр, предназначенный для широкого круга пользователей. Казуальные игры отличаются простыми правилами и не требуют от пользователя затрат времени на обучение или каких-либо особых навыков; они дёшевы в разработке и при дистрибуции. Многие подобные игры обладают также яркой и привлекательной графикой и минимумом текста. В последние время жанр казуальных игр эволюционирует в жанр «гиперказуал».

Гиперказуал — это жанр видеоигр, не требующий больших усилий для игрового процесса. Особенность гиперказуальных игр в том, что в основном они бесплатны, имеют упрощенный пользовательский интерфейс, не требуют специального обучения или инструкции для игрового процесса. В основном применяется дизайн с простой цветовой схемой и легкие, не требующие особых усилий механики, часто являющиеся бесконечно-зацикленными. Часто, гиперказуальные игры обсуждаются как бизнес-модель, а не полноценный жанр мобильных игр. Из-за отсутствия устойчивой внутриигровой экономики и отсутствия цены большинства гиперказуальных игр доход в основном генерируется за счет рекламы.

1.2.2 Аркада

Аркада — жанр компьютерных игр, характеризующийся коротким по времени, но интенсивным игровым процессом. Классические аркады характеризуются следующими свойствами:

- *Игра на одном экране.* В классических аркадах весь игровой процесс сосредоточен на одном экране. Игрок в любой момент времени может видеть весь игровой мир и принимать решения, исходя из полной информации о его состоянии.

- *Бесконечная игра*. Потенциально игрок может играть в аркаду бесконечное количество времени, и соответственно, не может выиграть. В таком случае, цель игры состоит в том, чтобы продержаться в ней как можно дольше.
- *Игровой счёт*. Практически все классические аркады включают в себя игровой счёт, когда игрок получает очки за выполнение различных целей или задач. На основании данной особенности у аркад как правило есть таблица рекордов, где игрок может сравнивать себя с другими игроками.
- *Нет сюжета.* Классические аркады не ставят целью рассказать какую-либо историю, и такая тенденция продолжается и для современных аркад.

1.2.3 Головоломка

Головоломка — это жанр компьютерных игр, целью которых является решение логических задач, требующих от игрока использования логики, стратегии и интуиции. Предшественниками жанра являлись обыкновенные настольные, графические и механические головоломки — от кроссвордов до шахмат. Эти головоломки требовали от игрока логики и ловкости в решении, которые также стали играть важную роль в прототипах различных поджанров. Однако, современные головоломки давно вышли за пределы копирования реальных головоломок и могу даже представлять собой полноценные игры в 3D.

1.2.4 Экшен

Экшен — это жанр компьютерных игр, в котором делается упор на эксплуатацию физических возможностей игрока, в том числе координации глаз и рук и скорости реакции. Жанр представлен во множестве разновидностей от «файтингов», «шутеров» и «платформеров», которые считаются наиболее важными для жанра, до «MOBA» и некоторых стратегий в реальном времени, которые возможно отнести к жанру «экшен». К экшен-играм может быть отнесена любая игра, где победа над противником обеспечивается благодаря физическому превосходству, например, лучшим прицеливанием или меньшем временем реакции.

1.2.5 Гонка

Гонка — это жанр компьютерных игр, в которых игрок принимает участие в гоночном соревновании среди наземных, водных, воздушных или космических транспортных средств. Основой этих соревнований могут быть какие-то реально существующие гоночные серии, также они могут проходить в полностью вымышленных мирах. В общем случае гоночной игрой может быть компьютерная игра любого жанра: от автосимуляторов высокой степени реализма до относительно простых аркадных гоночных игр.

1.3 Сравнение игровых движков

Игровой движок – это объединенный в единое целое комплекс прикладных программ, с помощью которых обеспечивается графическая визуализация, звуковое сопровождение, логика внутриигровых объектов в соответствии со скриптами, а также игра по сети, встроенные графические сцены, соблюдение физических эффектов и законов.

Разделение между игрой и игровым движком часто неопределённо. Некоторые движки имеют разумное и ясное разделение, в то же время другие практически невозможно отделить от игры. Одним из признаков игрового движка является применение архитектуры управления данными. Это определяется тем, что если игра содержит жёстко фиксированные данные, влияющие на логику, правила игры, рисование объектов и тому подобное, то становится сложно применять данное программное обеспечение в разных играх.

Большинство игровых движков разработано и настроено для того, чтобы запустить определённую игру на определённой платформе. И даже наиболее обобщённые многоплатформенные движки подходят для построения игр определённого жанра, например, головоломок или шутеров. В данном контексте можно сказать, что игровой движок становится не оптимальным при его применении не для той игры или той платформы, для которой разработан. Данный эффект проявляется от того, что программное обеспечение представляет собой набор компромиссов, основанных на тех предположениях, какой должна быть игра.

1.3.1 Свободно распространяемые движки

Во второй половине 1990-х и первой половине 2000-х многие студии использовали самописные движки, так как известные технологии были недоступны для небольших команд из-за высокой стоимости лицензирования. Сейчас ситуация изменилась: любой разработчик может выбрать оптимальную базу для своего проекта из множества вариантов — платных и бесплатных, для работы с 2D и 3D, универсальных и узкоспециализированных. Сайт «Game Data Crunch» провел анализ популярности движков по количеству игр в цифровом магазине «Steam» (рисунок 5). Основываясь на статистике за 2021 год, можно сделать вывод, что с отрывом первое место занимает Unity, следом за ним следует Unreal Engine. Также следует обратить внимание на набирающий популярность Godot.

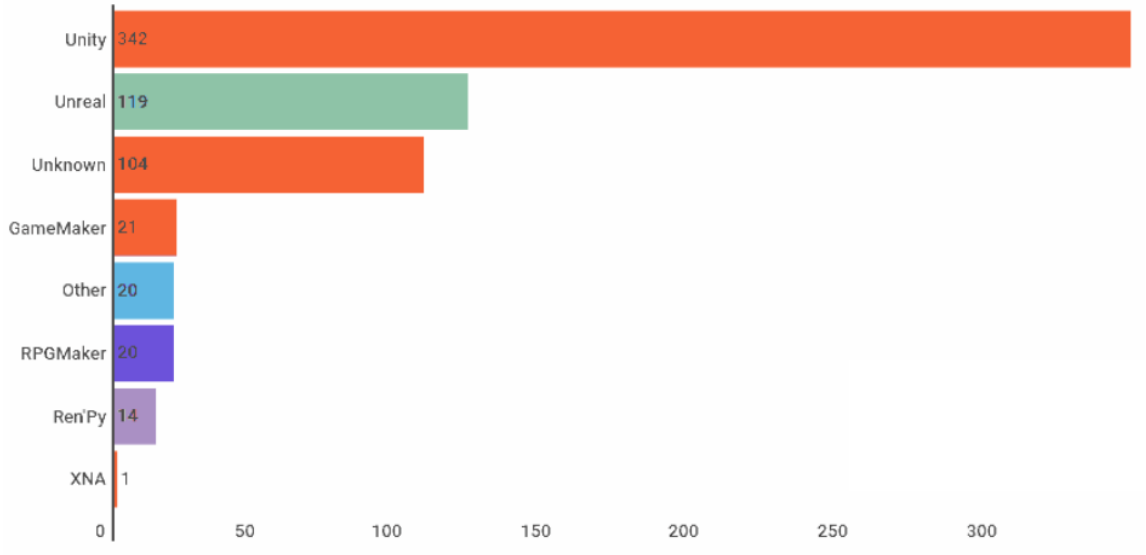

Рисунок 5. Наиболее популярные движки в «Steam»

1.3.1.1 Unity

Unity 3D — игровой движок, наиболее популярный среди инди-разработчиков. Его достаточно сложно освоить, но в этом помогают большое количество документации и видеоуроков. Основной язык программирования — C#, но имеющийся функционал позволяет создавать прототипы, не написав ни единой строчки кода. Встроенный магазин ассетов содержит десятки тысяч

платных и бесплатных моделей, шейдеров и прочих готовых ресурсов, что экономит время.

Достоинства:

- быстрое и удобное прототипирование;
- наличие бесплатной версии;
- совместимость с множеством платформой;
- возможность разработки как небольших, так и крупных проектов;
- высокая популярность движка;
- наличие собственного магазина с большим количеством ассетов.

Недостатки:

- медленная работа;
- закрытый исходный код;
- нестабильность редактора и отладчика.

1.3.1.2 Godot

Godot Engine — открытый кроссплатформенный 2D и 3D игровой движок. Отличается легкостью, мощностью и простотой в освоении. Godot поддерживает несколько языков программирования: C#, C++, собственный GDScript, основанный на Python, и язык визуального программирования. Архитектура игрового движка основана на дереве сцен, при этом каждый его элемент может стать сценой в любой момент. Поэтому архитектура проекта очень гибкая: она может изменяться и расширяться.

Еще одна особенность — все игровые ресурсы хранятся в папке проекта, как обычные файлы, и не являются частью базы данных. Что упрощает работу разработчикам в системе управлений версий. Это — не единственное удобство. В Godot минималистичный интерфейс и подробная документация, которая позволяет начинающему разработчику быстро освоиться и начать программировать практически с нуля.

Достоинства:

- простота в освоении;
- бесплатный;
- кроссплатформенность.

Недостатки:

- ограниченность функционала;
- невозможность создания проектов для всех платформ, в частности для игровых консолей.

1.3.1.3 Unreal Engine 4

Unreal Engine — игровой движок, разрабатываемый и поддерживаемый компанией «Epic Games». Достаточно труден в освоении, даже несмотря на встроенный язык сценариев Blueprints. Элементы UE4 распределяются на объекты, имеющие настраиваемые классы и определяемые ими характеристики. Среди основных классов выделяются актеры (действующие объекты), пешки (все, что управляется ИИ) и мир (все, что характеризует пространство). Язык, на котором функционирует движок — C++.

Создатели движка выпустили несколько часов обучающих роликов. Лучше всего возможности игрового движка Unreal Engine 4 раскрываются при разработке AAA-проектов. Он абсолютно бесплатный, однако, если игра зарабатывает менее 3000$ за квартал. Unreal Engine 4 используют в основном для AAA-проектов.

Достоинства:

- широкий набор инструментов;
- удобный интерфейс;
- большое количество документации и обучающих роликов;
- бесплатный, пока не начнет приносить прибыль;
- возможность кастомизации.

Недостатки:

- высокий порог вхождения;
- для комфортного использования требуется мощная конфигурация ПК;

- неудобные инструменты для создания 2D игр;
- многочисленные баги при разработки бесшовных миров;
- дорогие ассеты во встроенном магазине.

1.4 Выводы по главе

В ходе анализа платформ, жанров и перспектив разработки игры, был сделан ряд выводов. Игра будет разрабатываться для мобильных платформ, а в частности для платформы «Android». В следствие высокой популярности жанра «казуал», было бы логично выбрать его для разработки. Однако такая высокая популярность обусловлена тем, что таких игр очень много в цифровых магазинах (особенно в магазинах для «Android»), большинство из них очень похожи друг на друга и выделиться на их фоне будет достаточно сложно. Поэтому в качестве жанра будет выбрана «аркада», представляющая собой некое подобие гонки. В жанре «аркады» можно реализовать много необычных механик, способных привлечь игроков и надолго удержать их в игре. Также такую игру в перспективе будет проще выпустить в куда более требовательном к программному обеспечению «AppStore».

Анализируя выбранные движки, а также выбранную тематику разрабатываемой игры, можно понять, что Unreal Engine для мобильной игры подходит наименее всего. Если выбирать между Unity и Godot, ввиду большого количества образовательных материалов, документации для Unity, будет рациональнее остановиться именно на нем. Дополнительными преимуществами перед Godot является наличие собственного магазина ассетов, от моделей, до элементов UI. Все ассеты из магазина легко импортируются непосредственно в проект и сразу же готовы к использованию. Таким образом, окончательным движком разработки будет Unity.

ГЛАВА 2. Проектирование игрового процесса

2.1 Описание игрового процесса

2.1.1 Ключевая игровая механика

Основной геймплей заключается в управлении автомобилем. Обзор в игре осуществляется при помощи вида сверху, машина едет вперед автоматически, а с помощью кнопок, расположенных на экране, игрок может поворачивать машину.

Игровая локация представляет собой городской квартал. В некоторых местах находятся точки старта/финиша. Управляемая игроком машина появляется в одной из таких точек. Далее игроку отображается задача, в какую из других таких точек ему необходимо двигаться. Точки старта и финиша выбираются случайно до появления игрока на локации. Игроку необходимо добраться до заданной точки, не попав в ДТП.

При достижении заданной точки, на локации появляется ещё одна машина, которой теперь должен управлять игрок. Предыдущая же машина переходит под контроль искусственного интеллекта и полностью повторяет все перемещения, которые совершал игрок, пока управлял ею. Теперь, при построении маршрута, игроку необходимо учитывать маршрут предыдущей машины и не врезаться в нее. После достижения новой точки, на локации появляется новая машина, которая управляется игроком, а предыдущие две переходят под контроль ИИ. С каждым достижением игроком финиша, количество машин увеличивается. Игра продолжается до того момента, пока игрок не попадет в ДТП с окружением или другими машинами. Данная механика является главной особенностью игры, которая выделяет ее на фоне прочих игр.

2.1.2 Возможные геймплейные механики

Чтобы добавить интереса к игровому процессу или добавить новые источники дохода от игры, в планах реализовать некоторые игровые возможности. Однако, они являются второстепенными и реализовываться будут только после реализации основной.

2.1.2.1 Подбор бонусов

Чтобы упростить прохождение игроку локации, на карте могут появляться предметы, которые помогут ему в прохождении. Такими бонусами могут стать предметы, позволяющие проходить через машины, таранить их или перепрыгивать. Для поиска других идей для бонусов, рекомендуется изучить аркадные игры в жанре гонки.

2.1.2.2 Добавление таймера

Чтобы усложнить прохождение игроку локации, игроку будет дан таймер, по истечении которого игра заканчивается проигрышем игрока. Таймер не начинает идти по новой по достижении игроком финиша. Игрок может получать дополнительное время, используя бонусы, периодически появляющиеся на карте или по достижении финиша.

2.1.2.3 Внутриигровая валюта

Чтобы повысить интерес при прохождении, за как можно большее прохождение игроку начисляется количество монет или очков, которые он может тратить во внутриигровом магазине. Также внутриигровую валюту можно покупать за реальные деньги или за просмотр рекламы.

2.1.2.4 Система заданий

Чтобы повысить интерес при прохождении, игроку предлагается к выполнению некоторые задания, за выполнение которых он может получить внутриигровую валюту или предметы для кастомизации.

2.1.2.5 Кастомизация

Чтобы повысить интерес при прохождении а также дать игроку простор для фантазии, игроку доступна кастомизация. Она заключается как в визуальном изменении управляемой машины, так и в улучшении ее характеристик, таких как скорости, сцепления с дорогой и прочее. Улучшения или предметы игрок может получать за внутриигровую валюту или выполнение квестов.

2.1.2.6 Второй шанс

Чтобы упростить прохождение игроку локации, при провале игроку может быть дана возможность переиграть заезд на машине за просмотр рекламы или покупку дополнительных жизней за реальные деньги.

2.1.3 Возможные сложности в реализации геймплейных механик

Так как данная механика является уникальной, при ее разработке могут возникнуть некоторые трудности. Исходя из опыта разработки на Unity, можно отметить, что существуют способы, которые потенциально могут реализовать ключевую механику, на сказать точно можно будет только после начала разработки.

Также немаловажной проблемой является реализация появления новых автомобилей при прохождении игроком достаточно далеко. Основная сложность заключается в том, что в какой-то момент машин на локации станет больше, чем точек, и они начнут появляться друг в друге. Есть несколько способов устранения этой проблемы:

- Создавать новые машины немного в стороне от предыдущих.
- Создавать новые машины раньше или позже предыдущих, чтобы они успели уехать.
- При появлении новой машины отключать возможность сталкиваться с машинами, пока игрок не выйдет за пределы машины, внутри которой он появился.
- Когда количество машин станет равным количеству точек, удалять со сцены наиболее старые машины.

2.2 Архитектура игры

Проект Unity3D состоит из нескольких сцен (Scene), на которых расположены игровые объекты (GameObject) с прикрепленными к ним компонентами (Component). У каждого игрового объекта есть обязательный компонент Transform, отвечающий за положение объекта на сцене. Помимо

этого, могут быть подключены как готовые компоненты (Rigidbody, Collider), так и пользовательские компоненты (скрипты).

Программирование в Unity3D заключается в первую очередь в разработке пользовательских классов, которые подключаются к игровым объектам как компоненты. Все такие классы должны наследоваться от класса MonoBehaviour. Для удобного хранения и использования данных создаются классы, наследуемые от класса ScriptableObject. Создаваемые от них экземпляры используются в MonoBehaviour классах. На данный момент были спроектированы следующие классы игрового приложения, отвечающие за движение автомобиля и генерацию точек появления (рисунок 6).

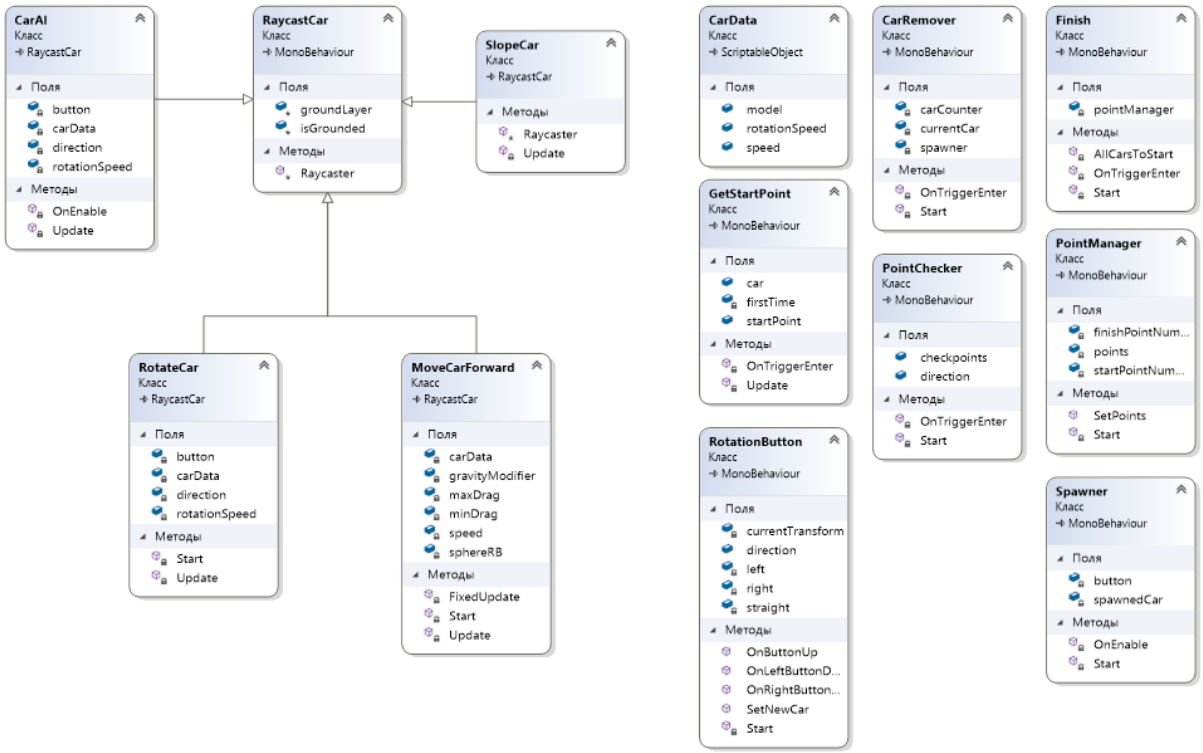

Рисунок 6. Диаграмма классов игры

Класс *RaycastCar* отвечает за получение информации, касается ли автомобиль земли.

Класс *MoveCarForward* отвечает за автоматическое движение машины прямо.

Класс *RotateCar* отвечает за поворот машины игроком.

Класс *CarAI* отвечает за поворот машины искусственным интеллектом.

Класс *SlopeCar* отвечает за наклон машины при заезде на склон.

Класс *CarData* хранит данные о машине, такие как скорость движения и скорость поворота.

Класс *RotationButton* отвечает за прием нажатия на кнопки интерфейса и передачу данных подконтрольному игроку автомобилю.

Класс *Spawner* отвечает за создание новой машины на сцене.

Класс *Finish* отвечает за проверку достижения финиша игроком..

Класс *PointManager* отвечает за генерацию точек старта и финиша.

Класс *GetStartPoint* отвечает за получение координаты точки появления машиной

Класс *CarRemover* отвечает за удаление со сцены лишних машин.

Класс *PointChecker* отвечает за определение правильного пути машиной под управлением искусственного интеллекта.

2.3 Диаграмма состояний

В результате проектирования была разработана UML-диаграмма состояний игрового приложения (рисунок 7).

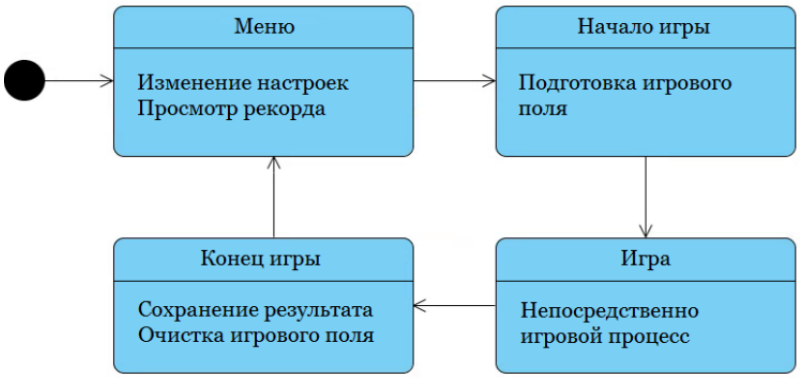

Рисунок 7. Диаграмма состояний игры

Состояние «Меню» – первое состояние игрового приложения. При переходе в него открывается главное меню и становится доступно изменение настроек игры. После команды игрока осуществляется переход в состояние «Начало игры».

Состояние «Начало игры» – при переходе в это состояние происходит загрузка сцены и инициализация необходимых для игры объектов. Переход в

следующее состояние осуществляется автоматически после инициализации игровых объектов.

Состояние «Игра» – когда игровое приложение находится в этом состоянии, игроку доступно управление машиной на игровом поле. При проигрыше осуществляется переход в следующее состояние.

Состояние «Конец игры» – при переходе в это состояние сохраняются результаты игры, открывается сцена главного меню и осуществляется переход в состояние «Меню».

2.4 Выводы по главе

В итоге, для игры определена основная концепция. Была описана ключевая геймплейная механика, а также представлены не обязательные для реализации механики. Сразу же был указан ряд проблем, которые могут возникнуть при разработке, что поможет избежать неприятностей непосредственно во время нее. Для более удобной реализации были составлены UML-диаграммы, такие как диаграмма классов и диаграмма состояний.

ГЛАВА 3. Реализация игрового процесса

3.1 Структура проекта

Разработка велась на языке программирования C# в среде разработки Visual Studio 2019.

На данный момент игра находится на стадии прототипа. В игре присутствует только одна сцена, на которой и происходит весь игровой процесс. На сцене предварительно находится несколько объектов (рисунок 8):

- Main Camera: отвечает за рендер всех видимых объектов на сцене.
- Directional Light: источник света на сцене.
- Plane: плоская поверхность, по которой ездит машина.
- Canvas: отвечает за расположение всех UI элементов.
- Right/Left Button: кнопки, отвечают за поворот машины.
- Control Button Controller: фиксирует нажатие на кнопки и передает данные с них машине.
- Points: объект, включающий в себя точки старта/финиша машины

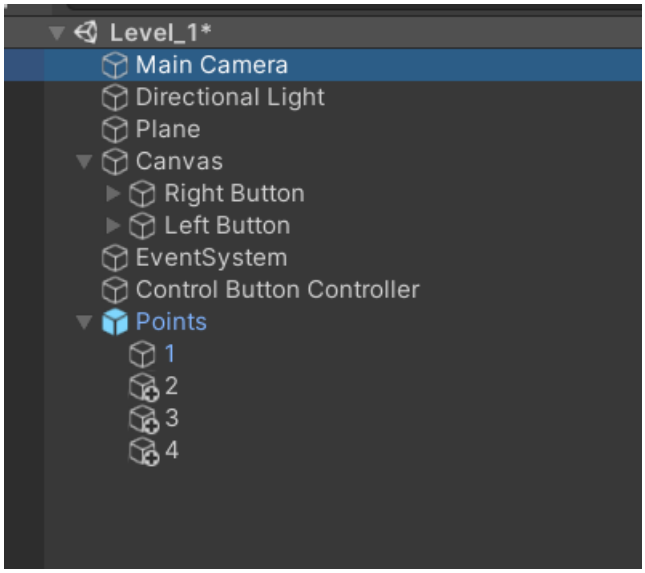

Рисунок 8. Иерархия объектов на сцене

3.2 Объект Green Car

Объект Green Car (рисунок 9) является машиной и контроливуемым игроком персонажем. Префаб машины содержит в себе несколько объектов

(рисунок 10), на которых расположены компоненты, осуществляющие управление машиной.

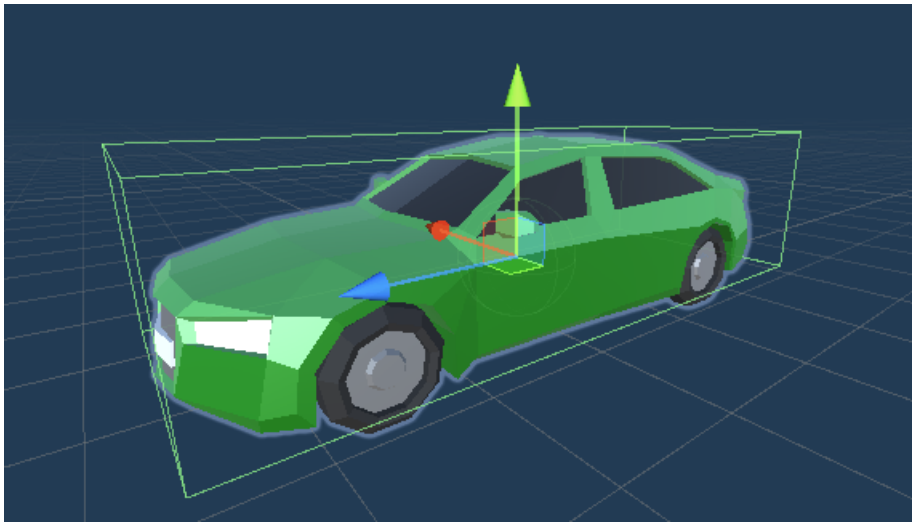

Рисунок 9. Префаб машины

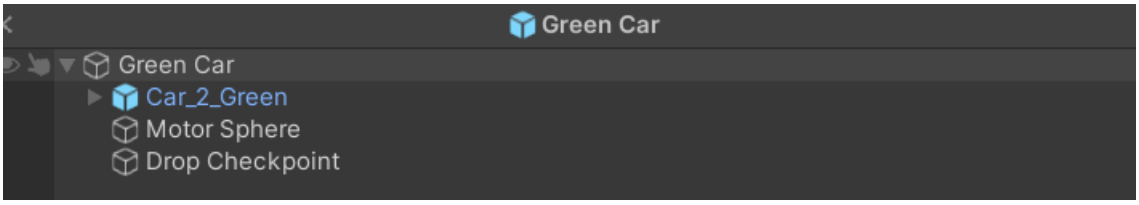

Рисунок 10. Иерархия объектов внутри префаба машины

Объект Green Car содержит в себе компоненты RaycastCar, MoveCarForward, RotateCar, SlopeCar и CarAI. Все эти компоненты являются скриптами.

Объект Car_2_Green содержит в себе компоненты MeshRenderer, Rigidbody, BoxCollider и PointChecker (скрипт). Объект используется для отрисовки модели машины и создания коллизии.

Объект Motor Sphere содержит в себе компоненты Rigidbody, SphereCollider и GetStartPoint (скрипт). Объект используется для реализации движения машины.

Объект Drop Checkpoint не содержит в себе компонентов и нужен только для определения позиции.

Компонент MeshRenderer отвечает за трехмерное графическое представление игрового объекта. Компонент CapsuleCollider, BoxCollider и

SphereCollider отвечают за обработку столкновений с другими объектами. Компонент Rigidbody добавляет игровому объекту физические свойства, такие как масса, скорость и ускорение.

Компонент MoveCarForward отвечает за движение машины строго вперед. Из объекта типа ScriptableObject компонент получает данные о скорости движения машины. Движение осуществляется с помощью объекта Motor Shpere. Каждый кадр сила толкает шар вперед, а модель машины телепортируется на позицию шара, в результате чего создается движение.

Компонент RotateCar отвечает за поворот машины игроком. Из объекта типа ScriptableObject компонент получает данные о скорости поворота машины. Поворот осуществляется с помощью данных о направлении поворота, получаемых с кнопок Left Button и Right Button. Каждый кадр функция поворачивает машину со скоростью умноженную на направление прямо (умножается на 0), налево (умножается на -1) или направо (умножается на 1), в результате чего получается поворот. Если не нажато никаких кнопок, стандартным коэффициентом поворота принимается 0.

Компонент RaycastCar отвечает за определение положения машины на земле. Из машины вертикально вниз направляется луч. Если луч касается земли, то машина может двигаться и поворачиваться. Если луч не касается земли, тогда машина не может двигаться и поворачиваться.

Компонент SlopeCar отвечает за наклон машины в зависимости от угла наклона поверхности. Из компонента RaycastCar компонент получает данные о нахождении машины на земле. Компонент высчитывает угол между лучом и землей и наклоняет машину на соответствующий угол.

Компонент CarAI отвечает за поворот машины без участия игрока. Из объекта типа ScriptableObject компонент получает данные о скорости поворота машины. При коллизии с отметкой, машина получает из нее данные о повороте. Каждый кадр функция поворачивает машину со скоростью умноженную на направление прямо (умножается на 0), налево (умножается на -1) или направо (умножается на 1), в результате чего получается поворот. Если машина не

встретила на пути отметки, она движется в направлении последней отметки. Изначальным коэффициентом поворота принимается 0.

Компонент PointChecker отвечает за фиксацию отметок о повороте. При коллизии с отметкой, компонент проверяет принадлежность отметки самой себе, а также направление поворота и присваивает переменной значение коэффициента поворота, после чего деактивирует отметку, чтобы не активировать ее второй раз и нарушить маршрут.

Компонент GetStartPoint отвечает за определение точки, на которой появилась машина. При появлении происходит триггер стартовой точки, после чего компонент запоминает ее данные, а также данные машины, к которой объект прикреплен.

3.3 Объект Control Button Controller

Объект Control Button Contoller является пустым объектом. Control Button Contoller содержит в себе компонент RotationButton, являющийся скриптом.

Компонент RotationButton отвечает за передачу данных от кнопок до машины. При нажатии на кнопку машина присваивает переменной значение коэффициента поворота, после чего создает отметку с соответствующим направлением поворота.

3.4 Объект Points

Объект Points (рисунок 11) представляет собой префаб, содержащий в себе некоторое количество точек старта/финиша, по умолчанию одну (рисунок 12). Для наглядности каждая точка является сферой, которая меняет цвет в зависимости от ее текущего состояния (белый – базовое состояние, красный – старт, зеленый – финиш). Points содержит в себе компонент PointManager. Объект Start/Finish Point содержит в себе компоненты Start, Finish и CarRemover. Все компоненты на этих двух объектах являются скриптами.

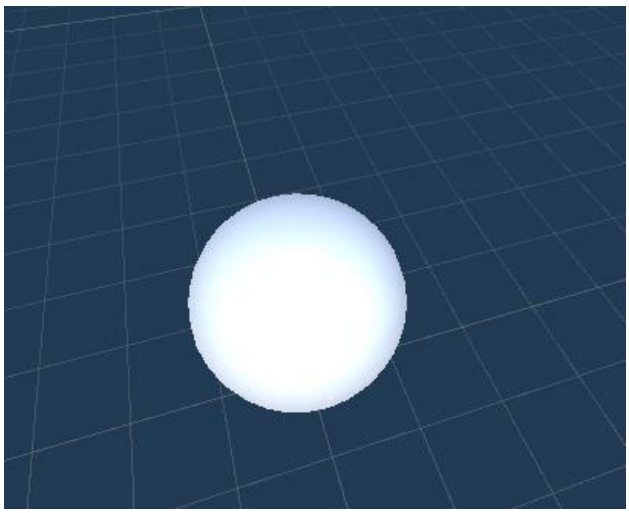

Рисунок 11. Префаб точки старта и финиша

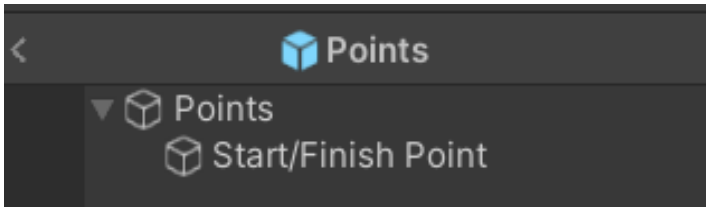

Рисунок 12. Иерархия объектов внутри префаба Points

Компонент PointManager отвечает за случайный выбор точек для старта и финиша. При запуске компонент находит все точки. Сначала компонент выбирает случайный старт, причем пока машин на сцене меньше, чем точек, все время выбирается разная точка. Если машин на поле столько же, сколько и точек, компонент выбирает случайную точку из всех. Выбранная точка окрашивается в красный цвет. После выбора точки старта из оставшихся выбирается точка финиша. Выбранная точка окрашивается в зеленый цвет. При достижении игроком финиша процесс повторяется.

Компонент Spawner отвечает за создание новой машины, управляемой игроком. Машина создается в позиции и повороте как у точки. При создании новой машины управление кнопками переходит к ней.

Компонент Finish отвечает за проверку достижения игроком точки финиша и выполнение соответствующих функций. При достижении игроком финиша все машины на карте перемещаются на старт, все неактивные отметки

поворота машин под управлением компьютером активируются, генерируются новые точки старта и финиша.

Компонент CarRemover отвечает за удаление лишних машин с поля. Компонент получает данные о вновь появившейся машине. Если на точке, в которой она появилась, уже есть машина, то старая машина и все ее отметки поворота удаляются.

3.5 Объект Отметка

Объект Отметка представляет собой тонкий коллайдер. Points содержит в себе компонент CapsuleCollider. Объекты отличаются друг от друга только тегами и названием (Left, Right и Straight). При коллизии с точкой машине поступают данные о коэффициенте поворота (-1, 1 и 0 соответственно).

3.6 Реализация пользовательского интерфейса

Объект Canvas отвечает за отображение элементов графического интерфейса (рисунок 13). В него вложены объекты Left Button и Right Button.

Объекты Left Button и Right Button отвечают за управление игроком машины, а также за появление точек поворота.

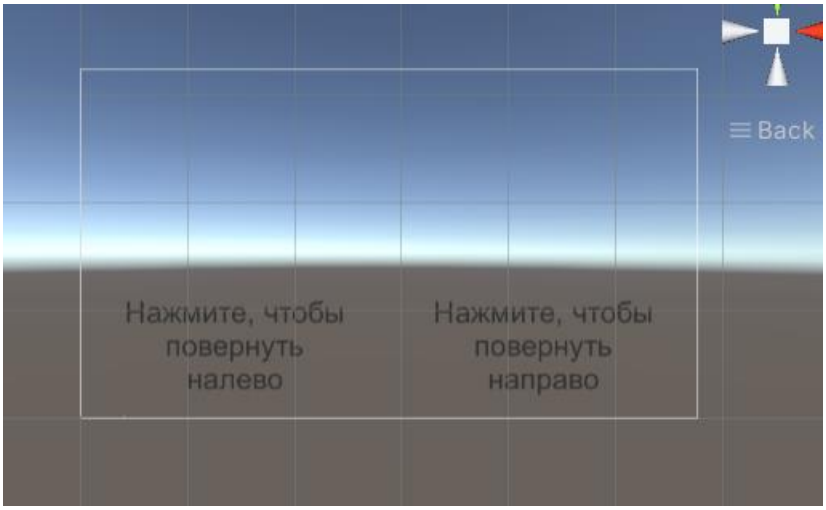

Рисунок 13. Canvas с кнопками управления

3.7 Выводы по главе

В результате разработки был представлен прототип, реализующий ключевую механику. Были описаны все объекты, входящие в состав игры, что они делают, а также принцип их работы.

Однако работа над игрой продолжается. Необходимо доделать все основные механики, разработать пользовательский интерфейс, создать карту, исправить все недоработки и баги и в целом довести игру до играбельного состояния.

ГЛАВА 4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

Целью выпускной квалификационной работы является проектирование и разработка игры в жанре «Аркада» для платформы Android.

В разделе финансового менеджмента, ресурсоэффективности и ресурсосбережения необходимо определить продолжительность работ и произвести расчет трудовых затрат проекта и эффективно организовать производство для уменьшения экономических затрат. Для эффективной организации производства необходимо экономически обосновать все инженерные решения.

Для достижения поставленной цели необходимо выполнить следующий ряд задач:

- оценка конкурентоспособности технических решений;
- проведение SWOT-анализа для выявления сильных и слабых сторон проекта;
- планирование проведения работ с построением диаграммы Ганта;
- расчет бюджета проекта;
- оценка ресурсной, финансовой и экономической эффективности.

4.1 Анализ конкурентных технических решений

Для понимания перспективности разработки информационной системы, в первую очередь необходимо провести анализ приложений-конкурентов. При таком анализе крайне важно реалистично оценить сильные и слабые стороны своей системы и системы-конкурента.

В качестве конкурирующих проектов были выбраны две игры для платформы Android: "Does Not Commute", на карте обозначена как Б1 и "Traffic Run", обозначена на карте как Б2. Данные игры являются бесплатными, представляют собой аркады про машины. Первая игра геймплейно похожа на разрабатываемую игру, а вторая - простая гиперказуальная игра про машины. Для

анализа были выбраны именно эти игры, потому что первая из них является референсом на разрабатываемую игру, а вторая - очень популярным представителем жанра. Разрабатываемая игра указана на карте как Б. Оценочная карта представлена в таблице 1.

Позиция выбранной разработки и альтернативных вариантов оценивается по каждому показателю экспертным путем по пятибалльной шкале, где 1 – наиболее слабая позиция, а 5 – наиболее сильная. Веса показателей, определяемые экспертным путем, в сумме должны составлять 1 (100%).

Таблица 1 – Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	**Вес**	**Баллы**			**Конкурентоспособность**		
		Б	Б1	Б2	К	К1	К2
Технические критерии оценки ресурсоэффективности							
Игровые механики	0,2	5	5	2	1	1	0,4
Производительность	0,2	4	5	5	0,8	1	1
Графика	0,15	3	5	2	0,45	0,75	0,3
Управление	0,15	4	5	5	0,6	0,75	0,75
Реиграбельность	0,1	5	2	3	0,5	0,2	0,3
Экономические критерии оценки эффективности							
Возможности для монетизации через рекламу	0,1	5	5	5	0,5	0,5	0,5
Возможности для монетизации через внутриигровые покупки	0,1	4	1	3	0,4	0,1	0,3
Итог	**1**	**30**	**28**	**25**	**4,25**	**4,3**	**3,55**

Пример оценки конкурентоспособности приведем для разрабатываемой игры: где К1, К2 – конкурентоспособности игр "Does Not Commute" Б1 и "Traffic Run"; вес критерия (в долях единицы, в сумме равняется 1), выбирается экспертным путем;

Согласно оценочной карте, наиболее конкурентоспособной игрой является игра "Does Not Commute" с показателем конкурентоспособности 4,3 условных

единиц, однако отставание разрабатываемой игры от нее минимально – 4,25. "Does Not Commute" совмещает в себе приятную графику, интересный и необычный геймплей, однако разрабатываемая игра имеет схожие геймплейные механики, а также предполагает реиграбельность и куда большие возможности для монетизации.

4.2. SWOT-анализ

SWOT-анализ применяется для оценки факторов и явлений, сопутствующих разработке и пользованию системы. Он проводится в несколько этапов. Первый этап представляет собой выявление сильных и слабых сторон проекта, а также возможностей и угроз.

Сильные стороны — это факторы, которые положительно сказываются на развитии проекта, в них включается все, что превращает функционирование в успешную и конкурентную работу.

Слабые стороны – это недостатки, упущения или ограниченность проекта, которые препятствуют достижению его целей. Это то, что плохо получается в рамках проекта или где он располагает недостаточными возможностями или ресурсами по сравнению с конкурентами.

Возможности включают в себя любую предпочтительную ситуацию в настоящем или будущем, возникающую в условиях окружающей среды проекта: тенденцию, изменение или предполагаемую потребность, которая поддерживает спрос на результаты проекта и позволяет руководству проекта улучшить свою конкурентную позицию.

Угроза представляет собой любую нежелательную ситуацию, тенденцию или изменение в условиях окружающей среды проекта, которые имеют разрушительный или угрожающий характер для его конкурентоспособности в настоящем или будущем. В качестве угрозы может выступать барьер, ограничение или что-либо еще, что может повлечь за собой проблемы, разрушения, вред или ущерб, наносимый проекту.

На первом этапе SWOT анализа в таблице 2 были описаны сильные и слабые стороны проекта, выявлены возможности и угрозы реализации игры.

Таблица 2 – Матрица SWOT анализа

Сильные стороны	Возможности во внешней среде
С1. Бесплатное распространение; С2. Уникальная игровая механика; С3. Популярность платформы, под которую разрабатывается игра.	В1. Распространение на другие платформы; В2. Интеграция монетизации разными способами; В3. Пострелизное насыщение контентом.
Слабые стороны	Угрозы внешней среды
Сл1. Достаточно узкая целевая аудитория; Сл2. Медленное распространение на начальном этапе; Сл3. Геймплей может показаться сложным.	У1. Отказ от технической поддержки проекта после выпуска; У2. Невозможность удержать аудиторию.

Второй этап состоит в выявлении соответствия сильных и слабых сторон проекта внешним условиям окружающей среды. Это соответствие или несоответствие должны помочь выявить степень необходимости проведения стратегических изменений. В рамках данного этапа необходимо построить интерактивную матрицу проекта. Ее использование помогает разобраться с различными комбинациями взаимосвязей областей матрицы SWOT. Возможно использование этой матрицы в качестве одной из основ для оценки вариантов стратегического выбора. Каждый фактор помечается либо знаком «+» (означает сильное соответствие сильных сторон возможностям), либо знаком «-» (что означает слабое соответствие); «0» – если есть сомнения в том, что поставить «+» или «-». Интерактивная матрица проекта представлена в табл. 3 и 4.

Таблица 3 - Интерактивная матрица сильных и слабых сторон и возможностей

	Сильные стороны				Слабые стороны		
Возможности проекта		С1	С2	С3	Сл1	Сл2	Сл3
	В1	+	+	0	-	-	0
	В2	+	+	+	+	-	0
	В3	-	+	+	-	+	+

Таблица 4 - Интерактивная матрица сильных сторон и слабых сторон и угроз

	Сильные стороны				Слабые стороны		
Угрозы проекта		С1	С2	С3	Сл1	Сл2	Сл3
	У1	+	+	-	+	-	+
	У2	+	+	-	+	+	+

Анализ интерактивных таблиц представляется в форме записи сильно коррелирующих сильных сторон и возможностей или слабых сторон и возможностей:

- В1С1С2;В2С1С2С3;

- В2Сл1;

- У1У2С1С2;

- У1Сл1Сл3; У2Сл1Сл2Сл3.

Самой большой угрозой для проекта является малый спрос из-за сложности игрового процесса, а также медленное распространение игры.

Для того, чтобы устранить некоторые слабые стороны проекта, необходимо увеличить финансирование рекламной кампании игры, а также провести ряд тестирований среди игроков для некоего упрощения игрового процесса.

В рамках третьего этапа составляется итоговая матрица SWOT-анализа, представленная в таблице 5.

Таблица 5 - Итоговая матрица SWOT-анализа

	Сильные стороны: С1. Бесплатное распространение; С2. Уникальная игровая механика; С3. Популярность платформы, под которую разрабатывается игра.	Слабые стороны: Сл1. Достаточно узкая целевая аудитория; Сл2. Медленное распространение на начальном этапе; Сл3. Геймплей может показаться сложным.
Возможности: В1. Распространение на другие платформы; В2. Интеграция монетизации разными способами; В3. Пострелизное насыщение контентом.	Игра может подняться на вершины рейтингов цифровых магазинов благодаря уникальной механике, что отлично скажется на доходе от нее.	Несмотря на необычную игровую механику, она может показаться сложной значительному числу игроков, что существенно отразится на популяризации игры.
Угрозы: У1. Отказ от технической поддержки проекта после выпуска; У2. Невозможность удержать аудиторию.	Отсутствие интереса к созданным игровым механикам может сказаться на распространении игры.	Аудитория игры может сокращаться из-за того, что без какого-либо нового контента игра надоест пользователю, или из-за отсутствия реакции на технические проблемы игроков.

4.3. Планирование работ по научно-техническому исследованию

4.3.1. Структура работ в рамках научного исследования

Проектирование комплекса предполагаемых работ осуществляется в следующем порядке:

- определение структуры работ в рамках проектирования и разработки игры;
- определение участников каждой работы;
- установление продолжительности работ;
- построение графика проведения технического проекта.

Для разработки игры в рамках выпускной квалификационной работы формируется рабочая группа, в состав которой входят разработчик и научный руководитель. Порядок составления этапов и работ от разработки технического задания и до оформления итогового отчета, распределение исполнителей по данным видам работ приведен в таблице 6.

Таблица 6 – Перечень этапов работ при проектировании

Основные этапы	№ работы	Содержание работы	Исполнитель
Разработка темы и задания	1	Утверждение темы, составление технического задания	разработчик; науч. рук.
	2	Составление календарного плана графика выполнения бакалаврской работы	разработчик; науч. рук.
Выбор направления исследования	3	Подбор и изучение материалов по теме работы	разработчик; науч. рук.
	4	Анализ предметной области	разработчик; науч. рук.
Теоретические исследования	5	Выбор программных решений для разработки клиентской части приложения	разработчик
	6	Проектирование системы	разработчик
Процесс разработки	7	Разработка основных игровых механик	разработчик
	8	Создание рабочего прототипа игры	разработчик
	9	Тестирование	разработчик
	10	Устранение выявленных ошибок	разработчик

Заключительный этап	11	Составление отчета о проделанной работе и оценка эффективности полученных результатов	разработчик; науч. рук.
	12	Защита дипломного проекта	разработчик

4.3.2. Определение трудоемкости выполнения работ

Трудовые затраты в большинстве случаях образуют основную часть стоимости разработки, поэтому важным моментом является определение трудоемкости работ.

Трудоемкость выполнения проекта оценивается экспертным путем в человеко-днях и носит вероятностный характер, который зависит от множества трудно учитываемых факторов. Для определения ожидаемого (среднего) значения трудоемкости используется следующая формула:

$$t_{\text{ож}i} = \frac{3t_{\min i} + 2t_{\max i}}{5},$$

где $t_{\text{ожі}}$ – ожидаемая трудоемкость выполнения і-ой работы чел.-дн.;

$t_{\min i}$– минимально возможная трудоемкость выполнения заданной і-ой работы, чел.-дн.;

$t_{\max i}$– максимально возможная трудоемкость выполнения заданной і-ой работы, чел.-дн.;

Исходя из ожидаемой трудоемкости работ, определяется продолжительность каждой работы в рабочих днях , учитывающая параллельность выполнения работ по нескольким исполнителями.

$$T_{\text{рі}} = \frac{t_{\text{ож}i}}{\text{Ч}_i},$$

где $T_{\text{рі}}$ – продолжительность одной работы, раб.дн.;

$t_{\text{ожі}}$ – ожидаемая трудоемкость выполнения одной работы, чел.-дн.;

Ч_i – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, чел.

4.3.3. Разработка графика проведения научного исследования

Наиболее удобным и наглядным представлением проведения научных работ является построение ленточного графика в форме диаграммы Ганта.

Диаграмма Ганта – горизонтальный ленточный график, на котором работы по теме представляются протяженными во времени отрезками, характеризующимися датами начала и окончания выполнения данных работ.

Для удобства построение графика, длительность каждого из этапов работ из рабочих дней следует перевести в календарные дни. Для этого необходимо воспользоваться следующей формулой:

$$T_{ki} = T_{pi} \bullet k_{кал},$$

где T_{ki} – продолжительность выполнения i-й работы в календарных днях;

T_{pi} – продолжительность выполнения i-й работы в рабочих днях;

$k_{кал}$– коэффициент календарности.

Коэффициент календарности определяется по следующей формуле:

$$k_{кал} = \frac{T_{кал}}{T_{кал} - (T_{вых} + T_{пр})},$$

где $T_{кал}$ – количество календарных дней в году;

$T_{вых}$ – количество выходных дней в году;

$T_{пр}$ – количество праздничных дней в году.

Расчет коэффициента календарности:

$$k_{кал} = \frac{T_{кал}}{T_{кал} - (T_{вых} + T_{пр})} = \frac{365}{365 - 118} = 1{,}48$$

Таблица 7 – Временные показатели научного исследования

Номер работы	**Трудоёмкость работ, чел.-дн.**			**Длительность работ, дн.**			
	tmin	tmax	tож	Тр		Тк	
				науч. рук.	**разраб.**	**науч. рук.**	**разраб.**
1	4	6	4,8	2,4	2,4	3,552	3,552
2	1	2	1,4	0,7	0,7	1,036	1,036
3	3	7	4,6	2,3	2,3	3,404	3,404

4	2	3	2,4	1,2	1,2	1,776	1,776
5	1	2	1,4	0	1,4	0	2,072
6	7	14	9,8	0	9,8	0	14,504
7	10	14	11,6	0	11,6	0	17,168
8	15	20	17	0	17	0	25,16
9	2	4	2,8	0	2,8	0	4,144
10	5	10	7	0	7	0	10,36
11	7	10	8,2	4,1	4,1	6,068	6,068
12	1	1	1	0,5	1	0,74	1,48
Итог	**58**	**93**	**72**	**11,2**	**61,3**	**16,576**	**90,724**

На основе полученной таблицы строится календарный план-график (таблица 8).

Таблица 8 – Календарный план-график проведения работ

Номер работы	**Исполнитель**		**Продолжительность выполнения работ**										
			февраль		**март**			**апрель**			**май**		
			2	**3**	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**
1	инженер; науч. рук.	3,552											
2	инженер; науч. рук.	1,036											
3	инженер; науч. рук.	3,404											
4	инженер; науч. рук.	1,776											
5	инженер	2,072											
6	инженер	14,504											
7	инженер	17,168											
8	инженер	25,16											
9	инженер	4,144											
10	инженер	10,36											
11	инженер; науч. рук.	6,068											

12	инженер	1,48											

синий – инженер, оранжевый – науч. рук.

4.4. Бюджет технического проекта

При планировании бюджета ТП должно быть обеспечено полное и достоверное отражение всех видов расходов, связанных с его выполнением. В процессе формирования бюджета ТП используется следующая группировка затрат по статьям:

- материальные затраты ТП;
- затраты на оборудование;
- амортизационные отчисления;
- заработная плата исполнителей темы;
- отчисления во внебюджетные фонды;
- накладные расходы.

4.4.1. Расчет материальных затрат

При планировании бюджета научно-техническое исследование должно быть обеспечено полное и достоверное отражение всех видов расходов, связанных с его выполнением.

Расчет материальных затрат осуществляется по формуле:

$$З_М = (1 + k_T) \cdot \sum_{i=1}^{m} Ц_i \cdot N_{расхi},$$

где m – количество видов материальных ресурсов, потребляемых при выполнении научного исследования;

$N_{расхi}$ – количество материальных ресурсов i-го вида, планируемых к использованию при выполнении научного исследования (шт., кг, м, м2 и т.д.);

$Ц_i$ – цена приобретения единицы i-го вида потребляемых материальных ресурсов (руб./шт., руб./кг, руб./м, руб./м2 и т.д.);

Таблица 9 – Материальные затраты

Наименование	Единица измерения	Цена за ед., руб.	Кол-во, ед.	Сумма, руб.
Электроэнергия	кВт*ч	300	3,5	1050

Распечатка	шт.	5	120	600
Итого				1650

4.4.2. Расчет затрат на оборудование

При планировании бюджета научно-техническое исследование должно быть обеспечено полное и достоверное отражение всех видов расходов, связанных с его выполнением.

Расчет материальных затрат осуществляется по формуле:

$$З_М = (1 + k_Т) \cdot \sum_{i=1}^{m} Ц_i \cdot N_{расхi},$$

где m – количество видов материальных ресурсов, потребляемых при выполнении научного исследования;

$N_{расхi}$ – количество материальных ресурсов i-го вида, планируемых к использованию при выполнении научного исследования (шт., кг, м, м2 и т.д.);

$Ц_i$ – цена приобретения единицы i-го вида потребляемых материальных ресурсов (руб./шт., руб./кг, руб./м, руб./м2 и т.д.);

$k_Т$ – коэффициент, учитывающий транспортно-заготовительные расходы.

В данную статью включают все затраты, связанные с приобретением специального оборудования (приборов, контрольно-измерительной аппаратуры, стендов, устройств и механизмов), необходимого для проведения работ. Определение стоимости спецоборудования производится по действующим прейскурантам.

При выполнении научно-исследовательской работы использовался персональный компьютер. Его стоимость представлена в таблице 10.

Таблица 10 – Расчет бюджета затрат на приобретение оборудования

Наименование оборудования	**Количество, шт.**	**Цена за 1 ед., руб.**	**Стоимость, руб.**
Ноутбук	1	90 000	90 000
Монитор	1	5 000	5 000
Компьютерная мышь	1	1 000	1 000
Итого: 96 000 рублей			

4.4.3. Основная заработная плата исполнителей

Статья включает основную заработную плату работников, непосредственно занятых выполнением проекта, (включая премии, доплаты) и дополнительную заработную плату и рассчитывается по формуле:

$$З_{зп} = З_{осн} + З_{доп}$$

где $З_{осн}$ – основная заработная плата;

$З_{доп}$ – дополнительная заработная плата (12–20 % от $З_{осн}$).

Основная заработная плата руководителя рассчитывается по следующей формуле:

$$З_{осн} = З_{дн} \cdot Т_р$$

где $З_{осн}$– основная заработная плата одного работника;

$Т_р$ – продолжительность работ, выполняемых научно-техническим работником, раб. дн.;

$З_{дн}$ – среднедневная заработная плата работника, руб.

Среднедневная заработная плата рассчитывается по формуле:

$$З_{дн} = \frac{З_м \cdot М}{F_д}$$

где $З_м$ – месячный должностной оклад работника, руб.;

М – количество месяцев работы без отпуска в течение года:

при отпуске в 24 раб. дня М =11,2 месяца, 5–дневная неделя;

при отпуске в 48 раб. дней М=10,4 месяца, 6–дневная неделя;

$F_д$ – действительный годовой фонд рабочего времени научно–технического персонала, раб. дн.

Таблица 24 - Баланс рабочего времени (для 6-дневной недели)

Показатели рабочего времени	**Дни**
Календарные дни	365
Нерабочие дни (праздники/выходные)	118
Потери рабочего времени (отпуск/невыходы по болезни)	66
Действительный годовой фонд рабочего времени	181

Месячный должностной оклад работника (руководителя):

$$З_м = З_{тс} \cdot (1 + k_{пр} + k_д) \cdot k_р$$

где $З_{тс}$ – заработная плата по тарифной ставке, руб.;

$k_{пр}$ – премиальный коэффициент, равный 0,3 (т.е. 30 процентов от $З_{тс}$);

$k_д$ – коэффициент доплат и надбавок составляет примерно 0,2 – 0,5;

$k_р$ – районный коэффициент, равный 1,3 (для Томска).

Таким образом, расчёт основной заработной платы представлен в таблице 11.

Таблица 11 – Расчёт основной заработной платы

Исполнители	**$З_{тс}$, руб.**	**$k_{пр}$**	**$k_д$**	**$k_р$**	**$З_м$, руб.**	**М**	**$F_д$**	**$З_{дн}$, руб.**	**$Т_р$, раб. дн.**	**$З_{осн}$, руб.**
Инженер	20000	0,3	0,2	1,3	40560	10,4	188	2253	72,3	162 892
Научный руководитель	47317	0,3	0,4	1,3	111952	10,4	188	6219	9,3	57 837
Итог: 220 729 рублей										

4.4.4. Расчет дополнительной заработной платы

Дополнительная заработная плата учитывает величину предусмотренных Трудовым кодексом РФ доплат за отклонение от нормальных условий труда, а также выплат, связанных с обеспечением гарантий и компенсаций (при исполнении государственных и общественных обязанностей, при совмещении работы с обучением, при предоставлении ежегодного оплачиваемого отпуска и т.д.).

Расчет дополнительной заработной платы рассчитывается по формуле:

$$З_{доп} = k_{доп} \cdot З_{осн},$$

где $k_{доп}$ – коэффициент дополнительной заработной платы, принятый на стадии проектирования за 0,12.

Расчёт дополнительной заработной платы приведён в таблице 12.

Таблица 12 – Расчёт дополнительной заработной платы

Исполнители	$З_{осн}$, руб.	$k_{доп}$	$З_{доп}$, руб.
Инженер	162 892	0,12	19 547,04
Руководитель	57 837	0,12	6 940,44
			Итог: 26 487 рублей

4.4.5 Отчисления во внебюджетные фонды

В данной статье расходов отражаются обязательные отчисления по установленным законодательством Российской Федерации нормам органам государственного социального страхования (ФСС), пенсионного фонда (ПФ) и медицинского страхования (ФФОМС) от затрат на оплату труда работников.

Величина отчислений во внебюджетные фонды определяется исходя из формулы:

$$З_{внеб} = k_{внеб} \cdot (З_{осн} + З_{доп})$$

где $k_{внеб}$ – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.).

В соответствии с Федеральным законом от 24.07.2009 №212-ФЗ установлен размер страховых взносов равный 30,2%.

Отчисления во внебюджетные фонды представлены ниже (таблица 13).

Таблица 13 – Отчисления во внебюджетные фонды

Исполнители	$З_{осн}$, руб.	$З_{доп}$, руб.	$k_{внеб}$	$З_{внеб}$, руб
Инженер	162 892	19 547,04	0,3	48 867
Руководитель	57 837	6 940,44	0,3	17 351
				Итог: 66 218 рублей

4.4.6. Накладные расходы

Накладные расходы учитывают прочие затраты организации, не попавшие в предыдущие статьи расходов. Их величина определяется по формуле:

$$З_{накл} = (\sum статей) \cdot k_{нр}$$

где $k_{нр}$ – коэффициент, учитывающий накладные расходы.

Величину коэффициента накладных расходов можно взять в размере 16%.

Расчет накладных расходов представлен в таблице 14.

Таблица 14 – Накладные расходы

$З_{мат}$, руб.	**$З_{м}$, руб.**	**$З_{осн}$, руб.**	**$З_{доп}$, руб.**	**$З_{внеб}$, руб.**
1650	96 000	220 729	26487	66 218
Итог: 65 773 рублей				

4.4.7 Формирование бюджета затрат проекта

Рассчитанная величина затрат проектировочной работы является основной для формирования бюджета затрат проекта, который при формировании договора с заказчиком защищается проектной организацией в качестве нижнего предела затрат на разработку. Данные бюджета затрат приведены в таблице 15.

Таблица 15 – Бюджет затрат

Наименование	**Сумма, руб.**	**Удельный вес, %**
Материальные затраты	1650	0,35
Затраты на оборудование	96 000	20,13
Затраты на основную заработную плату	220 729	46,29
Затраты на дополнительную заработную плату	26 487	5,55
Отчисления во внебюджетные фонды	66 218	13,89
Накладные расходы	65 773	13,79
Общий бюджет	476 857	100

4.5. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

Оценка сравнительной эффективности исследования основывается на определении интегрального показателя ресурсоэффективности, который можно определить следующим образом:

$$I_{\text{pi}} = \sum_{i=1}^{n} a_i \times b_i$$

где I_{pi} – интегральный показатель ресурсоэффективности для i-го варианта исполнения разработки;

a_i – весовой коэффициент i-го варианта исполнения разработки;

b_i^a, b_i^p – бальная оценка i-го варианта исполнения разработки, устанавливается экспертным путем по выбранной шкале оценивания;

n – число параметров сравнения.

Расчет интегрального показателя ресурсоэффективности представлен в таблице 16.

Таблица 16 – Оценка характеристик исполнения проекта

Критерий	**Весовой коэффициент**	**Балльная оценка разработки**
Предоставляет игроку уникальный игровой опыт	0,25	5
Удобное управление	0,2	4
Энергосбережение	0,1	5
Надежность	0,25	4
Возможность сопровождения и расширения функционала	0,2	5
Итог	1	4,55

Интегральный показатель ресурсоэффективности составил 4,55 баллов из 5 возможных, что свидетельствует об эффективности реализации проекта.

Глава 5. Социальная ответственность

Введение

Разработка игры производится группой работников, состоящей из двух человек – руководителя и студента. Выпускная квалификационная работа заключается в создании игры для платформы Android в жанре Аркада с помощью среды разработки игр Unity.

В качестве потребителя результатов проведенной разработки выступают пользователи девайсов, работающих на операционной системе Android. Конкретной возрастной категории нет, но так как игра представляет собой гонку, большая часть аудитории будет мужского пола.

Разработка игры проводилась в Томске.

В данном разделе проведен анализ вредных и опасных факторов труда, определен комплекс мер организационного, правового, технического и режимного характера, который должен способствовать снижению возможности возникновения негативных последствий работы разработчика.

Выпускная квалификационная работа по разработке игры для платформы Android в жанре Аркада на базе игрового движка Unity выполнялась в ходе преддипломной в учебной аудитории КЦ. Проектируемое рабочее место представляет собой комнату, в которой работал разработчик.

Характеристика помещения:

- ширина – 7 м, длина – 6 м, высота – 3 м;
- площадь – 42 м^2;
- объем – 126 м^3;
- в помещении установлен кондиционер, имеется естественная вентиляция – вытяжное вентиляционное отверстие, дверь, окно, щели;
- в помещении установлено искусственное освещение, имеется естественное освещение.

В данном помещении максимальное количество сотрудников в одну смену – 5. В среднем на одного сотрудника приходится 8,4 м^2 площади и 25,2 м^3 объема

помещения. Данное размещение сотрудников удовлетворяет санитарным нормам, согласно которым на одного работника должно приходиться не менее 6 $м^2$ площади и 24 $м^3$.

5.1. Правовые и организационные вопросы обеспечения безопасности

Конституция РФ является основным законом Российской Федерации. Она имеет высшую юридическую силу, прямое действие и применяется на территории всей страны.

Согласно статье 37 главы 1 Конституции РФ, каждый имеет право на труд в условиях, отвечающих требованиям безопасности и гигиены, каждый имеет право на отдых.

Трудовой Кодекс РФ устанавливает права и обязанности работника и работодателя, регулирует вопросы охраны труда, трудоустройства, правила оплаты и нормирования труда, порядок разрешения трудовых споров и другое.

Согласно статье 108 Трудового Кодекса РФ «Перерывы для отдыха и питания» в течение рабочего дня работнику должен быть предоставлен перерыв продолжительностью не более двух часов и не менее 30 минут, который в рабочее время не включается.

Согласно статье 212 Трудового Кодекса РФ «Обязанности работодателя по обеспечению безопасных условий и охраны труда», работодатель обязан обеспечить:

• безопасность работников при эксплуатации зданий, оборудования, осуществлении технологических процессов, применяемых материалов;

• создание и функционирование системы управления охраной труда;

Согласно статье 219 Трудового Кодекса РФ «Право работника на труд в условиях, отвечающих требованиям охраны труда», каждый работник имеет право на:

• соответствующее требованиям охраны труда рабочее место;

• обязательное социальное страхование от несчастных случаев на производстве и профессиональных заболеваний;

• получение достоверной информации от работодателя об условиях и охране труда на рабочем месте, о существующем риске повреждения здоровья, мерах защиты от воздействия вредных и опасных факторов производства;

• отказ от выполнения работ в случае возникновения опасности для его жизни и здоровья вследствие нарушения требований охраны труда до устранения такой опасности.

ГОСТ 12.2.032-78 «Система стандартов безопасности труда содержит требования, по которым должна производиться организация рабочего места при выполнении работы. Рабочее место при выполнении работ сидя» и соблюдением трудовых норм, регулирующихся Трудовым кодексом РФ.

Согласно требованиям ГОСТ 12.2.032-78:

• конструкция рабочего места и взаимное расположение всех его элементов должны соответствовать физиологическим и психологическим требованиям, а также характеру работы.

• высота рабочего стола с клавиатурой должна составлять 680 - 800 мм над уровнем стола;

• высота экрана над полом – 900-1280 мм, монитор должен находиться в 600-700 мм от работника на 20 градусов ниже уровня глаз;

ГОСТ 21889-7б «Система «человек-машина». Кресло человека-оператора. Общие эргономические требования» содержит требования, предъявляемые к креслу человека-оператора.

Согласно требованиям ГОСТ 21889-7б:

- кресло должно обеспечивать длительное поддержание основной рабочей позы в процессе трудовой деятельности.
- кресло должно создавать условия для поддержания корпуса человека в физиологически рациональном положении с сохранением естественных изгибов позвоночника.

ГОСТ 22269-7б «Система «человек-машина». Рабочее место оператора. Взаимное расположение элементов рабочего места» содержит требования, предъявляемые к рабочему месту человека-оператора.

Согласно требованиям ГОСТ 22269-7б:

- Взаимное расположение элементов рабочего места должно обеспечивать возможность осуществления всех необходимых движений и перемещений для эксплуатации и технического обслуживания оборудования.
- Взаимное расположение элементов рабочего места должно способствовать оптимальному режиму труда и отдыха, снижению утомления оператора, предупреждению появления ошибочных действий.

На рабочем месте, предоставленном для работы с выпускной работой, были учтены и соблюдены все требования по организации труда.

5.2. Производственная безопасность

В процессе работы химические и биологические факторы не оказывают влияния на состояние здоровья. В ходе раздела рассматриваются физические и психофизиологические факторы.

К вредным и опасным производственным факторам относятся:

- повышенный уровень электромагнитного излучения;
- недостаточная освещенность рабочей зоны;
- монотонность процесса работы;
- нарушение правил электробезопасности;
- эмоциональные перегрузки.

Таблица 17 - Возможные опасные и вредные факторы производственные факторы на рабочем месте за персональным компьютером

Факторы (ГОСТ 12.0.003-2015)	Нормативные документы
1. Производственные факторы, связанные с электромагнитными излучениями;	СанПиН 1.2.3685-21
2. Отсутствие или недостаток необходимого искусственного освещения;	СП 52.13330.2016
3. Повышенный уровень шума;	СанПиН 1.2.3685-21
4. Монотонность труда, вызывающая монотонию;	–
5. Производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которого попадает работающий;	ГОСТ 12.1.019-2017
6. Длительное сосредоточенное наблюдение	–

5.2.1. Анализ опасных и вредных факторов и обоснование мероприятий по снижению их воздействия

5.2.1.1. Производственные факторы, связанные с электромагнитными излучениями

Персональный компьютер подвергает пользователя вредному электромагнитному излучению. Электромагнитное излучение компьютера изменяется в диапазоне частот от 0 Гц до 1000 МГц. Такое излучение состоит из электрической и магнитной составляющих.

Норма допустимых уровней напряженности полей и излучений регламентируются СанПиН 1.2.3685-21 "Санитарно-эпидемиологические требования к физическим факторам на рабочих местах". Согласно установленным нормам время пребывания работника в рабочей зоне вычисляется по формуле:

$$T = (50/E) – 2.$$

Время пребывания в рабочей зоне составляет примерно 8 часов в день. На рабочем месте уровень напряженности электрических полей не выше 4 кВ/м. При котором разрешенное время пребывания в рабочей зоне может составлять до 10,5 часов. Следовательно, уровень электромагнитных излучений на рабочем месте в норме.

5.2.1.2. Отсутствие или недостаток необходимого искусственного освещения

Недостаточная освещенность рабочей зоны возникает вследствие отсутствия должного количества источников освещения в рабочей зоне и снижает работоспособность, значительно влияет на здоровье работников, а именно на их качество зрения.

В СП 52.13330.2016 зрительная работа сотрудника, работающего с персональным компьютером, охарактеризована как работа разряда Б – высокой точности, подразряда 1. В таблице 18 представлены требования к освещению рабочего помещения для вышеуказанного разряда.

Таблица 18 — Требования к освещению рабочего помещения для разряда Б1 -

Освещенность на рабочей поверхности от системы	Цилиндрическая освещенность, лм	Объединенный показатель дискомфорта, не более	Коэффициент пульсации освещенности Кп, %, не более
300	100	21	15

Для снижения влияния фактора недостаточной освещенности на рабочем месте необходимо, чтобы уровень естественного освещения и яркость экрана персонального компьютера были приблизительно одинаковыми, так как яркий свет в зоне периферийного зрения заметно увеличивает глазное напряжение и приводит к быстрой утомляемости. Путем решения проблемы недостаточной освещенности помещения может стать установка качественных источников искусственного освещения.

5.2.1.3. Повышенный уровень шума

Повышенный уровень шума на рабочем месте обусловлен использованием персональных компьютеров, наличием центральной системы вентиляции и кондиционирования воздуха. В процессе длительной работы за персональным компьютером пользователю становится сложнее слышать, работоспособность снижается и повышается утомляемость.

Ниже представлены предельно допустимые уровни звукового давления, уровни звука и эквивалентные уровни звука для разработчиков программного обеспечения и людей, работающих с программным обеспечением, описанные в СанПиН 1.2.3685-21.

Таблица 19 - Предельно допустимые уровни звукового давления, уровни звука и эквивалентные уровни звука

Вид трудовой деятельности, рабочее место	Уровни звукового давления, дБ в октавных полосах со среднегеометрическими частотами, Гц									Уровни звука и эквивалентные уровни звука, дБА
	31,5	63	125	250	500	1000	2000	4000	8000	
Конструирование и проектирование, программирование.	86	71	61	54	49	45	42	40	39	54

Существуют следующие пути уменьшения воздействий шума: экранирование рабочих мест; чистка оборудования от пыли, т.к. любое оборудование при загрязнении увеличивает уровень шума.

5.2.1.4. Монотонность труда

Многие виды работы требуют от сотрудника длительного выполнения однообразных действий или непрерывной и устойчивой концентрации внимания. Поэтому монотонность является серьезным негативным фактором

В условиях монотонной работы с организмом человека могут произойти такие изменения как:

- изменение функционального состояния центральной нервной системы;
- снижение уровня бодрствования;
- нарушение автоматизма деятельности и способности к переключениям;
- изменение биологического ритма.

Так как деятельность разработчика программных систем связана только с работой на ПК, она является монотонной. Такая работа требует непрерывной концентрации внимания на протяжении длительного времени и является однообразной.

Для снижения уровня монотонности можно проводить следующие мероприятия:

- внедрение режима труда и отдыха;
- частые, но кратковременные перерывы во время работы;
- выполнение физических упражнений в течение перерывов
- посещение специальных помещений отдыха.

5.2.1.5. Производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которого попадает работающий

Наличие большого количества электрических приборов и вычислительных машин на рабочем месте обуславливают важность электробезопасности на производстве.

При работе с электроприборами необходимо соблюдать технику безопасности. Общие правила по электробезопасности регламентируются ГОСТ 12.1.019-2017.

Серьезной проблемой является опасность поражения электрическим током. Человеческие органы чувств не могут обнаружить наличие электрического напряжения на расстоянии.

При повышенной влажности (относительная влажность воздуха выше 75 %) возрастает риск поражения электрическим током. Высокая температура воздуха и поверхностей (более 35 °С) также повышает вероятность распространения электрического тока. Наличие токопроводящей пыли и токопроводящих полов повышает риск поражения электрическим током.

Также существует риск возникновения опасности:

- при прикосновении к токоведущим частям (во время ремонта ПК);
- при прикосновении к нетоковедущим частям, которые оказались под напряжением (при нарушении изоляции);
- при соприкосновении с полом или стенами, оказавшимися под напряжением (при нарушении электрической сети);
- при коротком замыкании в высоковольтных блоках.

Место, в котором выполнялась работы, не относится к помещениям повышенной опасности электропоражения. В помещении используются стандартные бытовые приборы, потребляющие напряжение 220 В переменного тока с частотой 50 Гц.

Для предотвращения возникновения опасных ситуаций обязательны следующие меры предосторожности:

- перед началом работы необходимо убедиться, что выключатели и розетки закреплены и не имеют оголенных токоведущих частей;
- взаимодействовать с электроприборами исключительно сухими руками;
- при обнаружении неисправности оборудования и приборов, необходимо сообщить ответственному лицу, не делая никаких самостоятельных исправлений;
- запрещено загромождать рабочее место лишними предметами.

5.2.1.6. Длительное сосредоточенное наблюдение

Умственный труд разработчика заключается в приеме информации, ее переработке и выработке решения. От напряженного умственного труда страдают зрительные и слуховые анализаторы, центральная нервная система, в особенности высшие психические функции.

К факторам возникновения эмоциональных перегрузок можно отнести: длительное эмоциональное напряжение, хроническую усталость, нарушение режимов труда и отдыха, жизненные трудности и так далее.

Эти факторы сказываются на снижении интереса к работе и работоспособности, появлении раздражительности и конфликтности, повышается количество ошибок в работе, психоэмоциональные сдвиги.

Для снижения эмоциональных перегрузок необходимы:

- умственные тренировки;
- повышение квалификации;
- умеренные и постоянные производственные нагрузки;
- правильное трудовое, психологическое и эстетическое воспитание;
- умение переносить стрессовые состояния;
- повышение интереса к работе;
- наличие источников положительных эмоций;
- оптимальное расписание отдыха.

5.3. Экологическая безопасность

Разработка проектного решения экологически безопасна, однако может

косвенно влиять на атмосферу, так как работа компьютера связана с потреблением электроэнергии и нагревом аппаратных средств.

Электроэнергия вырабатывается на тэц, где сжигаются углеродные соединения. Компьютер, который использовался для разработки системы, потребляет около 85 Ватт электроэнергии в час. Энергоблок 200 МВт потребляет 90 тонн угля в час. За час работы над проектным решением сжигается 40г угля в час. Если работа над проектным решением заняла 30 дней при работе 4 часа в день, то за все время работы было сожжено 4 кг 800 г угля. Соответственно выброс CO2 в атмосферу составит около 9 кубических метров.

Этот показатель не является существенным, однако для сокращения влияния на атмосферу и экономии денежных средств следует выключать компьютер в нерабочее время.

5.4. Безопасность в чрезвычайных ситуациях

Чрезвычайной ситуацией называется обстановка, сложившаяся в результате аварии, опасного природного явления, катастрофы или другого бедствия, которая может повлечь за собой человеческие жертвы, ущерб здоровью людей или окружающей среде, значительные материальные потери и нарушение условий жизнедеятельности людей.

Пожар – наиболее вероятной чрезвычайной ситуацией для представленного рабочего помещения. Нарушение техники использования электрических приборов и ПК, нарушениях разводки электрических сетей и ряда других причин могут привести к пожару.

Рабочее помещение, представленное для выполнения выпускной квалификационной работы, согласно СанПиН 1.2.3685-21, можно отнести к категории В (пожароопасное).

Главные причины возникновения пожара:

- короткое замыкание;
- перегрузка сетей, которая ведет за собой сильный нагрев токоведущих частей и загорание изоляции;

- запуск оборудования после некорректного ремонта.

Чтобы обеспечить состояние защищенности пользователя и имущества от пожара, необходимо соблюдать правила пожарной безопасности.

Для защиты от коротких замыканий и перегрузок необходимо правильно выбирать, устанавливать и использовать электрические сети и средства автоматизации.

Для предупреждения возникновения пожаров необходимо исключить образование горючей среды, удостовериться в несгораемости и трудносгораемости материалов, примененных при строительстве и отделке помещений, проверить помещение на наличие средств пожаротушения.

Пожарно-профилактические мероприятия:

- организационные мероприятия, касающиеся технического процесса с учетом пожарной безопасности объекта (обучение правилам техники безопасности, распространение планов эвакуации);
- эксплуатационные мероприятия, рассматривающие эксплуатацию используемого оборудования (соблюдение техники безопасности при эксплуатации оборудования, обеспечение свободного подхода к оборудованию, поддержание исправности изоляции проводников);
- технические и конструктивные мероприятия, связанные с правильным размещением и монтажом электрооборудования и отопительных приборов.

Для повышения устойчивости рабочего помещения к ЧС необходимо наличие систем противопожарной сигнализации, реагирующих на дым и другие продукты горения, установку огнетушителей.

В случае возникновения возгорания, необходимо вызвать пожарную службу по телефону 101 и сообщить место возникновения ЧС, предпринять меры по эвакуации в соответствии с планом эвакуации. При отсутствии прямых угроз здоровью и жизни произвести попытку тушения возникшего возгорания имеющимися углекислотными огнетушителями.

Вывод по главе

В результате проведенного анализа был рассмотрен процесс разработки системы с различных точек зрения, таких как правовая, экологическая, производственная, а также с позиции обеспечения безопасности при чрезвычайных ситуациях. Рабочее место соответствует всем необходимым нормам, кроме норм освещенности, однако из-за наличия естественных источников освещения, разработчик не испытывал каких-либо неудобств. По критериям электробезопасности помещение относится к категории безопасное. Тяжесть труда во время рабочего процесса по созданию информационной системы относится к категории I в соответствии с СанПин 1.2.3685-21. Рабочее помещение, представленное для выполнения ВКР, согласно СанПиН 1.2.3685-21, можно отнести к категории В (пожароопасное).

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы был проведен полный комплекс мероприятий по разработке игры. В первую очередь, был проведен анализ рынка и перспективы разработки видеоигр, а также мобильных игр. После того, как стало ясно, что разработка видеоигр является актуальной на сегодняшний день, предстояло определиться с жанром. Сравнив популярность жанров у игроков, рекламодателей а также в целом, что представляет из себя жанр, было принято решение разрабатывать игру в жанре аркады.

Следующим этапом стало проектирование игры. Сначала была описана идея игры, подробно расписаны основные механики, дальнейшие возможности и перспективы добавления новых механик, а также трудности, с которыми можно столкнуться во время разработки. Для демонстрации архитектуры игры были созданы UML-диаграммы, такие как диаграмма классов и последовательности.

В результате разработки был представлен прототип, реализующий ключевую механику. Были описаны все объекты, входящие в состав игры, что они делают, а также принцип их работы.

Однако работа над игрой продолжается. После добавления остальных важных механик, настройки камеры, создания полноценного пользовательского интерфейса и первой игровой локации необходимо провести внутреннее тестирование. Группа игроков должна опробовать игру и указать на недостатки игры, которые впоследствии должны быть по возможности исправлены.

СПИСОК ЛИТЕРАТУРЫ

1. Седых И.А. ИНДУСТРИЯ КОМПЬЮТЕРНЫХ ИГР. Москва: Издательский дом ВШЭ, 2020. 120 с
2. Анализ популярных движков [Электронный ресурс] URL: https://www.gamedatacrunch.com (дата обращения: 08.04.2022).
3. Описание движков [Электронный ресурс] URL: https://blackcaviar.games/obzor_igrovyh_dvizhkov (дата обращения: 08.04.2022).
4. Популярные движки в Steam [Электронный ресурс] URL: https://www.gamedeveloper.com/business/game-engines-on-steam-the-definitive-breakdown (дата обращения: 08.04.2022).
5. Анализ рынка и креативные стратегии [Электронный ресурс] URL: https://apptractor.ru/marketing-monetization/app-promotion-campaign-analytics/socialpeta-2022-mobile-game-ad-ultimate-guide-analiz-rynka-i-kreativnye-strategii.html (дата обращения: 09.04.2022).
6. Тенденции развития мобильных игр [Электронный ресурс] URL: https://rb.ru/opinion/mobile-game-2021 (дата обращения: 09.04.2022).
7. Рынок видеоигр [Электронный ресурс] URL: https://newzoo.com/insights/articles/the-games-market-in-2021-the-year-in-numbers-esports-cloud-gaming/ (дата обращения: 09.04.2022).
8. Сегменты рынка видеоигр [Электронный ресурс] URL: https://www.gamesindustry.biz/articles/2021-12-21-gamesindustry-biz-presents-the-year-in-numbers-2021 (дата обращения: 09.04.2022).
9. Популярные жанры мобильных игр [Электронный ресурс] URL: https://www.statista.com/statistics/515399/global-reach-popular-android-game-genres/#:~:text=Casual%20games%20were%20the%20most,penetration%20rate%20of%2056.5%20percent.&text=The%20top%20Android%20game%20by%20reach%20worldwide%20in%202019%20was%20Subway%20Surfers. (дата обращения: 11.04.2022).
10. Популярные мобильные игры [Электронный ресурс] URL: https://hc.games/trendy-mobilnyh-igr-2022/ (дата обращения: 11.04.2022).

11. Документация Unity // Docs Unity URL: https://docs.unity3d.com/ru/530/Manual/ (дата обращения: 10.05.2022).
12. C# Programming Guide. [Электронный ресурс] URL: https://msdn.microsoft.com/en-us/library/67ef8sbd.aspx (дата обращения: 10.05.2020).

ПРИЛОЖЕНИЕ А. Класс CarAI

```
using UnityEngine;
public class CarAI : RaycastCar
{
    [SerializeField] private CarData carData;
    [SerializeField] private PointChecker button;
    private float rotationSpeed;
    private int direction;
    private void OnEnable()
    {
        button = FindObjectOfType<PointChecker>();
        rotationSpeed = carData.rotationSpeed;
    }
    void Update()
    {
        direction = button.direction;
        Raycaster();
        if (isGrounded)
            transform.Rotate(0, direction * rotationSpeed *
Time.deltaTime, 0, Space.World);
    }
}
```

ПРИЛОЖЕНИЕ Б. Класс MoveCarForward

```
using UnityEngine;
public class MoveCarForward : RaycastCar
{
    [SerializeField] private CarData carData;
    [SerializeField] private Rigidbody sphereRB;
    [SerializeField] private float gravityModifier = 30f;
    [SerializeField] private float minDrag = 0f;
    [SerializeField] private float maxDrag = 4f;
    private float speed;
    void Start()
    {
        speed = carData.speed;
        sphereRB.transform.parent = null;
    }
    void Update()
    {
        transform.position = sphereRB.transform.position;
    }
    private void FixedUpdate()
    {
        Raycaster();
        if (isGrounded)
        {
            sphereRB.AddForce(transform.forward * speed);
            sphereRB.drag = maxDrag;
        }
        else
        {
            sphereRB.AddForce(transform.up * -gravityModifier);
            sphereRB.drag = minDrag;
        }
    }
}
```

ПРИЛОЖЕНИЕ В. Класс RotateCar

```
using UnityEngine;
public class RotateCar : RaycastCar
{
    [SerializeField] private CarData carData;
    private RotationButton button;
    private float rotationSpeed;
    private int direction;
    void Start()
    {
        button = FindObjectOfType<RotationButton>();
        rotationSpeed = carData.rotationSpeed;
    }
   void Update()
    {
        direction = button.direction;
        Raycaster();
        if (isGrounded)
            transform.Rotate(0,   direction   *   rotationSpeed   *
Time.deltaTime, 0, Space.World);
    }
}
```

ПРИЛОЖЕНИЕ Г. Класс PointChecker

```
using UnityEngine;
using System.Collections.Generic;
public class PointChecker : MonoBehaviour
{
    public int direction = 0;
    public List<GameObject> checkpoints = new List<GameObject>();
    void Start()
    {
    }
    private void OnTriggerEnter(Collider other)
    {
        if (other.tag == "Left" && transform.parent.name == other.name)
        {
            if (enabled)
            {
                direction = -1;
                other.gameObject.SetActive(false);
            }
            checkpoints.Add(other.gameObject);
        }
        if (other.tag == "Right" && transform.parent.name == other.name)
        {
            if (enabled)
            {
                direction = 1;
                other.gameObject.SetActive(false);
            }
            checkpoints.Add(other.gameObject);
        }
        if (other.tag == "Straight" && transform.parent.name == other.name)
        {
```

```
            if (enabled)
            {
                direction = 0;
                other.gameObject.SetActive(false);
            }
            checkpoints.Add(other.gameObject);
        }
    }
}
```

ПРИЛОЖЕНИЕ Д. Класс RaycastCar

```
using UnityEngine;
public class RaycastCar : MonoBehaviour
{
    [SerializeField] protected LayerMask groundLayer;
    protected bool isGrounded;
    protected virtual void Raycaster()
    {
        RaycastHit hit;
        isGrounded    =    Physics.Raycast(transform.position,    -
transform.up, out hit, groundLayer);
    }
}
```

ПРИЛОЖЕНИЕ Е. Класс SlopeCar

```
using UnityEngine;
public class SlopeCar : RaycastCar
{
    void Update()
    {
        Raycaster();
    }
    protected override void Raycaster()
    {
        RaycastHit hit;
        isGrounded    =    Physics.Raycast(transform.position,    -
transform.up, out hit, groundLayer);
        transform.rotation                                        *=
Quaternion.FromToRotation(transform.up, hit.normal);
    }
}
```

ПРИЛОЖЕНИЕ Ж. Класс GetStartPoint

```
using UnityEngine;
public class GetStartPoint : MonoBehaviour
{
    public Transform startPoint;
    public Transform car;
    private bool firstTime = true;
    private void Update()
    {
        if (car == null)
            Destroy(gameObject);
    }
    private void OnTriggerEnter(Collider other)
    {
        //Получение координат стартовой позиции при появлении
        if (other.tag == "Respawn" && firstTime)
        {
            startPoint = other.transform;
            firstTime = false;
        }
    }
}
```

ПРИЛОЖЕНИЕ 3. Класс RotationButton

```
using UnityEngine;
public class RotationButton : MonoBehaviour
{
    [SerializeField] private GameObject left;
    [SerializeField] private GameObject right;
    [SerializeField] private GameObject straight;
    [SerializeField] private Transform currentTransform;
    public int direction = 0;
    private void Start()
    {
        SetNewCar();
    }
    public void OnLeftButtonDown()
    {
        direction = -1;
        GameObject point = Instantiate(left,
currentTransform.position, currentTransform.rotation) as GameObject;
        point.name = currentTransform.parent.name;
    }
    public void OnRightButtonDown()
    {
        direction = 1;
        GameObject point = Instantiate(right,
currentTransform.position, currentTransform.rotation) as GameObject;
        point.name = currentTransform.parent.name;
    }
    public void OnButtonUp()
    {
        direction = 0;
        GameObject point = Instantiate(straight,
currentTransform.position, currentTransform.rotation) as GameObject;
        point.name = currentTransform.parent.name;
    }
    public void SetNewCar()
```

```
    {
        currentTransform = GameObject.FindGameObjectWithTag("Drop
Checkpoint").GetComponent<Transform>();
    }
}
```

ПРИЛОЖЕНИЕ И. Класс Spawner

```
using UnityEngine;
public class Spawner : MonoBehaviour
{
    [SerializeField] private GameObject spawnedCar;
    [SerializeField] private RotationButton button;
    void Start()
    {
        button = FindObjectOfType<RotationButton>();
    }
    private void OnEnable()
    {
        GameObject car = Instantiate(spawnedCar, transform.position,
transform.rotation) as GameObject;
        car.name = gameObject.name + "car";
        button.SetNewCar();
    }
}
```

ПРИЛОЖЕНИЕ К. Класс Spawner

```
using UnityEngine;
public class Finish : MonoBehaviour
{
    private PointManager pointManager;
    private void Start()
    {
        pointManager = GameObject.FindObjectOfType<PointManager>();
    }
    private void OnTriggerEnter(Collider other)
    {
        if (!enabled) return;
        if (other.gameObject.tag == "Player")
        {
            // Все машины перемещаются на свои стартовые позиции
            AllCarsToStart();
            other.tag = "Untagged";
            GameObject.FindGameObjectWithTag("Drop  Checkpoint").tag
= "Untagged";
            other.GetComponentInParent<RotateCar>().enabled = false;
            other.GetComponentInParent<CarAI>().enabled = true;
            other.GetComponent<PointChecker>().enabled = true;
            pointManager.SetPoints();
            foreach(GameObject                obj                in
GameObject.FindGameObjectsWithTag("Left"))
            {
                obj.SetActive(true);
            }
            foreach           (GameObject           obj           in
GameObject.FindGameObjectsWithTag("Right"))
            {
                obj.SetActive(true);
            }
            foreach           (GameObject           obj           in
GameObject.FindGameObjectsWithTag("Straight"))
```

```
            {
                obj.SetActive(true);
            }
            // Увеличение счёта
        }
    }
    void AllCarsToStart()
    {
        foreach (GameObject sphere in
GameObject.FindGameObjectsWithTag("Motor Sphere"))
        {
            sphere.GetComponent<Rigidbody>().velocity = new
Vector3(0, 0, 0);
            GetStartPoint point =
sphere.GetComponentInChildren<GetStartPoint>();
            sphere.transform.position = point.startPoint.position;
            point.car.rotation = point.startPoint.rotation;
        }
    }
}
```

ПРИЛОЖЕНИЕ Л. Класс PointManager

```
using System.Collections.Generic;
using UnityEngine;
public class PointManager : MonoBehaviour
{
    [SerializeField] private List<Spawner> points = new
List<Spawner>();
    private List<int> startPointNumbers = new List<int>();
    private List<int> finishPointNumbers = new List<int>();
    void Start()
    {
        for (int i = 0; i < FindObjectsOfType<Spawner>().Length;
i++)
        {
            startPointNumbers.Add(i);
            finishPointNumbers.Add(i);
        }
        SetPoints();
    }
    public void SetPoints()
    {
        //Обнуление всех точек
        points = new List<Spawner>(FindObjectsOfType<Spawner>());
        foreach (Spawner point in points)
        {
            point.gameObject.GetComponent<Spawner>().enabled =
false;
            point.gameObject.GetComponent<Finish>().enabled = false;
            point.gameObject.GetComponent<Renderer>().material.color
= Color.white;
            point.gameObject.tag = "Untagged";
        }
        int number;
        int subNumber;
        if (startPointNumbers.Count > 0)
```

```
        {
            subNumber = Random.Range(0, startPointNumbers.Count);
            number = startPointNumbers[subNumber];
            startPointNumbers.Remove(number);
        }
        else
        {
            number = Random.Range(0, points.Count);
        }
        points[number].gameObject.tag = "Respawn";
        points[number].gameObject.GetComponent<Spawner>().enabled   =
true;
points[number].gameObject.GetComponent<Renderer>().material.color   =
Color.red;
        finishPointNumbers.Remove(number);
        var x = number;
        subNumber = Random.Range(0, finishPointNumbers.Count);
        number = finishPointNumbers[subNumber];
        finishPointNumbers.Add(x);
        points[number].gameObject.GetComponent<Finish>().enabled    =
true;
points[number].gameObject.GetComponent<Renderer>().material.color   =
Color.green;
    }
```

ПРИЛОЖЕНИЕ М. Класс CarRemover

```
using UnityEngine;
public class CarRemover : MonoBehaviour
{
    [SerializeField] private GameObject currentCar;
    private Spawner spawner;
    private int carCounter = 0;
    void Start()
    {
        spawner = GetComponent<Spawner>();
    }
```

```
    private void OnTriggerEnter(Collider other)
    {
        if(other.tag == "Player" && spawner.enabled && carCounter <
1)
        {
            currentCar = other.gameObject;
            carCounter++;
        }
        else if (other.tag == "Player" && spawner.enabled &&
carCounter >= 1)
        {
            Destroy(currentCar.transform.root.gameObject);
            foreach (GameObject obj in
currentCar.GetComponent<PointChecker>().checkpoints)
                Destroy(obj);
            currentCar = other.gameObject;
        }
}
```

ПРИЛОЖЕНИЕ Н. Класс PointManager

```
using UnityEngine;
[CreateAssetMenu(fileName = "CarData", menuName = "ScriptableObjects/CarData")]
public class CarData : ScriptableObject
{
    public float speed;
    public float rotationSpeed;
    public GameObject model;
}
```