

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Школа: Инженерная школа информационных технологий и робототехники
Направление подготовки 09.03.04: Программная инженерия
Отделение: Отделение информационных технологий

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Разработка виртуального полигона для беспилотного автомобиля

УДК 004.415:629.33.072:004.946

Студент

Группа	ФИО	Подпись	Дата
8К4А	Журбич Никита Игоревич		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР ТПУ	Фофанов Олег Борисович	К.Т.Н.		

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН ШБИП ТПУ	Петухов Олег Николаевич	К.Э.Н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Авдеева Ирина Ивановна			

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР ТПУ	Чердынцев Евгений Сергеевич	К.Т.Н.		

Томск – 2018 г.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ООП

Код результатов	Результат обучения (выпускник должен быть готов)	Требования ФГОС, критерии АИОР
Р1	Применять базовые и специальные естественнонаучные и математические знания в области информатики и вычислительной техники, достаточные для комплексной инженерной деятельности.	Требования ФГОС (ОК-1, 10, ПК-4, 5, 6), критерий 5 АИОР (п. 1.1)
Р2	Применять базовые и специальные знания в области современных информационных технологий для решения инженерных задач.	Требования ФГОС (ОК-11, 12, 13, ПК-1, 2, 11), критерий 5 АИОР (п.1.1, 1.2)
Р3	Ставить и решать задачи комплексного анализа, связанные с созданием аппаратно-программных средств информационных и автоматизированных систем, с использованием базовых и специальных знаний, современных аналитических методов и моделей.	Требования ФГОС (ОК-1, 8, ПК-2, 4, 6), критерий 5 АИОР (п. 1.2)
Р4	Разрабатывать программные и аппаратные средства (системы, устройства, блоки, программы, базы данных и т. п.) в соответствии с техническим заданием и с использованием средств автоматизации проектирования.	Требования ФГОС (ОК-2, 3, ПК-3, 4, 5), критерий 5 АИОР (п. 1.3)
Р5	Проводить теоретические и экспериментальные исследования, включающие поиск и изучение необходимой научно-технической информации, математическое моделирование, проведение эксперимента, анализ и интерпретация полученных данных, в области создания аппаратных и программных средств информационных и автоматизированных систем.	Требования ФГОС (ОК-6, ПК-6, 7), критерий 5 АИОР (п.1.4)
Р6	Внедрять, эксплуатировать и обслуживать современные программно-аппаратные комплексы, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять	Требования ФГОС (ОК-4, 15, 16, ПК-9, 10, 11), критерий 5 АИОР

	требования по защите окружающей среды.	(п. 1.5)
P7	Использовать базовые и специальные знания в области проектного менеджмента для ведения комплексной инженерной деятельности.	Требования ФГОС (ОК-1, 4, ПК-1, 6, 7), критерий 5 АИОР (п. 2.1)
P8	Владеть иностранным языком на уровне, позволяющем работать в иноязычной среде, разрабатывать документацию, презентовать и защищать результаты комплексной инженерной деятельности.	Требования ФГОС (ОК-14, ПК-7), критерий 5 АИОР (п. 2.2)
P9	Эффективно работать индивидуально и в качестве члена группы, состоящей из специалистов различных направлений и квалификаций, демонстрировать ответственность за результаты работы и готовность следовать корпоративной культуре организации.	Требования ФГОС (ОК-2, 3, 4), критерий 5 АИОР (п. 2.3, 2.4)
P10	Демонстрировать знания правовых, социальных, экономических и культурных аспектов комплексной инженерной деятельности.	Требования ФГОС (ОК-1, 5, 9), критерий 5 АИОР (п. 2.5)
P11	Демонстрировать способность к самостоятельной к самостоятельному обучению в течение всей жизни и непрерывному самосовершенствованию в инженерной профессии.	Требования ФГОС (ОК-6, 7), критерий 5 АИОР (п. 2.6)

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

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)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

бакалаврской работы

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

Студенту:

Группа	ФИО
8К4А	Журбичу Никите Игоревичу

Тема работы:

Разработка виртуального полигона для беспилотного автомобиля	
Утверждена приказом директора (дата, номер)	

Срок сдачи студентом выполненной работы:	20.06.2018 г.
--	---------------

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

<p>Исходные данные к работе <i>(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.).</i></p>	<p>Объектом исследования является виртуальный полигон для беспилотного автомобиля, с помощью которого можно проводить тестирование различных алгоритмов движения беспилотного автомобиля.</p> <p>Среда разработки программного обеспечения Unity3D и Microsoft Visual Studio 2017. Языки программирования C# и JavaScript.</p>
--	--

<p>Перечень подлежащих исследованию, проектированию и разработке вопросов <i>(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).</i></p>	<ol style="list-style-type: none"> 1. Обзор предметной области 2. Проектирование алгоритмов программного комплекса. 3. Разработка алгоритмов в виде программного комплекса 4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение 5. Социальная ответственность
--	--

<p>Перечень графического материала <i>(с точным указанием обязательных чертежей)</i></p>	<ol style="list-style-type: none"> 1. Диаграмма «Рыбий скелет». 2. Диаграмма вариантов использования 3. Диаграмма классов 4. Диаграмма потоков данных 5. Презентация в Microsoft PowerPoint
--	--

<p>Консультанты по разделам выпускной квалификационной работы <i>(с указанием разделов)</i></p>	
<p style="text-align: center;">Раздел</p>	<p style="text-align: center;">Консультант</p>
<p>Финансовый менеджмент, ресурсоэффективность и ресурсосбережение</p>	<p>Петухов Олег Николаевич</p>
<p>Социальная ответственность</p>	<p>Авдеева Ирина Ивановна</p>

<p>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</p>	<p>10.02.2018 г.</p>
--	----------------------

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР ТПУ	Фофанов Олег Борисович	К.Т.Н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К4А	Журбич Никита Игоревич		

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Школа: Инженерная школа информационных технологий и робототехники

Направление подготовки (специальность): Программная инженерия

Уровень образования: Бакалавр

Отделение школы (НОЦ): Отделение информационных технологий

Период выполнения: (осенний / весенний семестр 2017/2018 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:	20 июня 2018 г.
--	-----------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
02.05.2018	Раздел 1. Обзор предметной области	15
08.05.2018	Раздел 2. Проектирование программного комплекса	20
20.05.2018	Раздел 3. Реализация программного комплекса	25
25.05.2018	Раздел 4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	20
28.05.2018	Раздел 5. Социальная ответственность	20

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР ТПУ	Фофанов О.Б.	к.т.н.		

СОГЛАСОВАНО:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР ТПУ	Чердынцев Е.С.	к.т.н.		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

Группа	ФИО
8К4А	Журбичу Никите Игоревичу

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	Бакалавриат	Направление/специальность	09.03.04 Программная инженерия

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:	
1. Стоимость ресурсов научного исследования (НИИ): материально-технических, энергетических, финансовых, информационных и человеческих	1. З/п руководителя – 46284,4 руб. Стипендия студента – 12439,79 руб.
2. Нормы и нормативы расходования ресурсов	2. Расходы на электроэнергию – 1206,48 руб.
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	3. Отчисления во внебюджетные фонды – 13885,32 руб. Прочие расходы – 8053 руб.
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИИ с позиции ресурсоэффективности и ресурсосбережения	1. Потенциальные потребители результатов исследования; 2. Анализ конкурентных технических решений; 3. SWOT – анализ.
2. Планирование и формирование бюджета научных исследований	1. Структура работ в рамках научного исследования; 2. Определение трудоемкости выполнения работ и разработка графика проведения научного исследования; 3. Бюджет проекта.
3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	1. Определение интегрального финансового показателя разработки; 2. Определение интегрального показателя ресурсоэффективности разработки; 3. Определение интегрального показателя эффективности
Перечень графического материала (с точным указанием обязательных чертежей):	
1. Матрица SWOT 2. Таблица трудозатрат на выполнение проекта 3. Линейный график работ	

Дата выдачи задания для раздела по линейному графику	01.03.2018
---	------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент ОСГН ШБИП ТПУ	Петухов Олег Николаевич	К.Э.Н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К4А	Журбич Никита Игоревич		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8К4А	Журбичу Никите Игоревичу

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	Бакалавриат	Направление/специальность	09.03.04 Программная инженерия

Исходные данные к разделу «Социальная ответственность»:

1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения

Объект исследования – программный комплекс (ПК), который позволит проводить виртуальное испытание беспилотного автомобиля, а также алгоритмов автоматического управления транспортным средством. Данный ПК будет использоваться на персональном компьютере, соответственно, в этом разделе исследуется рабочее место человека, который будет работать с этим модулем.

Рабочая зона – аудитория, оборудованная системой отопления, кондиционирования воздуха, с естественным и искусственным освещением. Рабочее место – стационарное, оборудованное персональным компьютером и оргтехникой.

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Производственная безопасность

1.1. Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения

1.2. Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения

1.1. Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности:

- Повышенный уровень электромагнитных излучений
- Повышенный уровень шума
- Слабая освещённость рабочей зоны
- Отклонение микроклиматических условий
- Умственное перенапряжение

1.2. Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности:

- Опасность поражения электричеством
- Возможность возникновения короткого замыкания
- Статическое электричество

2. Экологическая безопасность

Анализ негативного воздействия на окружающую природную среду: утилизация персональных компьютеров и другой используемой оргтехники, а также люминесцентных ламп.

3. Безопасность в чрезвычайных ситуациях

Возможной чрезвычайной ситуацией техногенного характера для данной сферы деятельности является пожар в результате возгорания электропроводки, перегрева рабочих частей персонального компьютера.

	Создание общих правил поведения и рекомендаций во время пожара, разработка плана эвакуации, ознакомление с использованием огнетушителей типа ОУ-5.
4. Правовые и организационные вопросы обеспечения безопасности	<ul style="list-style-type: none"> - Регулирование организации рабочих мест при выполнении работ сидя согласно ГОСТ 12.2.032-78. - Организация рабочих мест с электронно-вычислительными машинами регулируется СанПиНом 2.2.2/2.4.1340 – 03.

Дата выдачи задания для раздела по линейному графику	01.03.2018
---	------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Авдеева Ирина Ивановна			01.03.2018

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К4А	Журбич Никита Игоревич		01.03.2018

Реферат

Дипломная работа содержит: 76 страниц, 12 рисунков, 21 таблицу, 17 источников.

Ключевые слова: беспилотный автомобиль, UML, Unity3D, алгоритмы движения беспилотного автомобиля, виртуальный полигон, программный комплекс.

Объектом исследования являются алгоритмы движения беспилотного автомобиля в виртуальном мире (полигоне).

Цель работы - создание программного комплекса, который позволит проводить виртуальное испытание алгоритмов беспилотного автомобиля.

Перечень условных обозначений, единиц и терминов

- Программный комплекс (ПК) - техническое решение концепции алгоритма работы сложной системы, управление которой осуществляется, как правило, исполнением кода из определённого базового набора команд (системы команд).
- Язык UML (Unified Modeling Language – унифицированный язык моделирования) – язык графического описания для объектного моделирования в области разработки программного обеспечения.
- DFD (Data Flow Diagram - диаграмма потоков данных) – описывает как каждый процесс преобразует свои входные данные в выходные, и выявляет отношения между процессами, протекающими в системе.
- Декомпозиция - научный метод, позволяющий заменить решение одной большой задачи решением серии меньших задач, пусть и взаимосвязанных, но более простых.
- Нотация - Набор символов и правил их использования для описания чего-либо.
- Беспилотный автомобиль - транспортное средство, оборудованное системой автоматического управления, которое может передвигаться без участия человека.
- WebGL (Web-based Graphics Library) — программная библиотека для языка программирования JavaScript.

Оглавление

Реферат	13
Введение.....	17
Глава 1. Обзор предметной области.....	18
1.1. Проблема, цель.....	18
1.2. Преимущества и недостатки разработки ПК.....	20
1.3. Анализ требований.....	21
Глава 2. Проектирование программного комплекса	23
2.1. Функции ПК	23
2.2. Диаграмма вариантов использования	24
2.3. Диаграмма классов.....	25
2.4. Диаграмма потоков данных	30
2.5. Выбор инструмента для разработки.....	31
Глава 3. Реализация программного комплекса	33
3.1. Обоснование выбора языка программирования.....	33
3.2. Создание виртуального полигона	34
3.3. Внедрение пешехода в систему виртуального полигона	36
3.4. Внедрение автомобиля в систему виртуального полигона	40
3.5. Дальнейшие планы разработки.....	42
Глава 4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение.....	45
4.1. Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения.....	45
4.2. Организация и планирование работ.....	47

4.3. Расчет сметы затрат на выполнение проекта	52
4.4. Выводы по главе.....	57
Глава 5. Социальная ответственность.....	58
5.1. Производственная безопасность	58
5.2. Экологическая безопасность	67
5.3. Безопасность в чрезвычайных ситуациях.....	68
5.4. Правовые и организационные вопросы обеспечения безопасности	69
5.5. Вывод по главе	71
Заключение	72
Список публикаций.....	73
Список использованных источников	74
Приложение А	76
Приложение Б	77
Приложение В.....	78
Приложение Д.....	79

Введение

С каждым днём к нам приближается недалекое и так отчетливо видимое электронное будущее, которое принесет нам массу нововведений. Уже сегодня мы можем наблюдать за рождением новых, ярких идей и технологий. Одной из наиболее интересных, перспективных и массовых технологий является идея создания беспилотного автотранспорта.

Беспилотный автомобиль - транспортное средство, оборудованное системой автоматического управления, которое может передвигаться без участия человека.

Целью проекта является разработка трехмерной реалистичной имитации процесса управления движением автомобиля в городской среде.

Создаваемый программный комплекс позволит разрабатывать аппаратные и программные системы беспилотного автомобиля и проводить виртуальное испытание разработанных систем, в том числе алгоритмов автоматического управления транспортным средством.

Данный программный комплекс разрабатывается на межплатформенной среде разработки компьютерных игр Unity3D с помощью инструментов Microsoft Visual Studio 2017 и MonoDeveloper.

Глава 1. Обзор предметной области

Отличительной особенностью данного проекта является реализация движения беспилотного автомобиля в виртуальной среде, что является новым подходом при создании беспилотного автомобиля. Большинство аналогов на рынке создают свои собственные полигоны для тестирования беспилотного автомобиля, что занимает большое количество времени и денежных средств.

Основные сферы применения данного программного комплекса:

- грузоперевозки;
- военная промышленность;
- гражданская авиация;
- сельское хозяйство.

Конкретные потребители данного продукта:

- предприниматели;
- армия;
- авиакомпании;
- сельское хозяйство.

В наше время развитие беспилотного автотранспорта разделилось на 3 основных направления:

- потребительское (личное авто, такси, городская авто транспортная сеть)
- промышленное (специализированная техника)
- военное (боевые машины различного спектра задач)

1.1. Проблема, цель

Для изучения и поиска причин рассматриваемой проблемы используется диаграмма «Рыбий скелет». С помощью данной диаграммы можно определить проблемы, которые могут повлиять на разработку рассматриваемого комплекса.

На рисунке 1 продемонстрирована диаграмма «Рыбий скелет» для анализа предметной области и поиска причины.

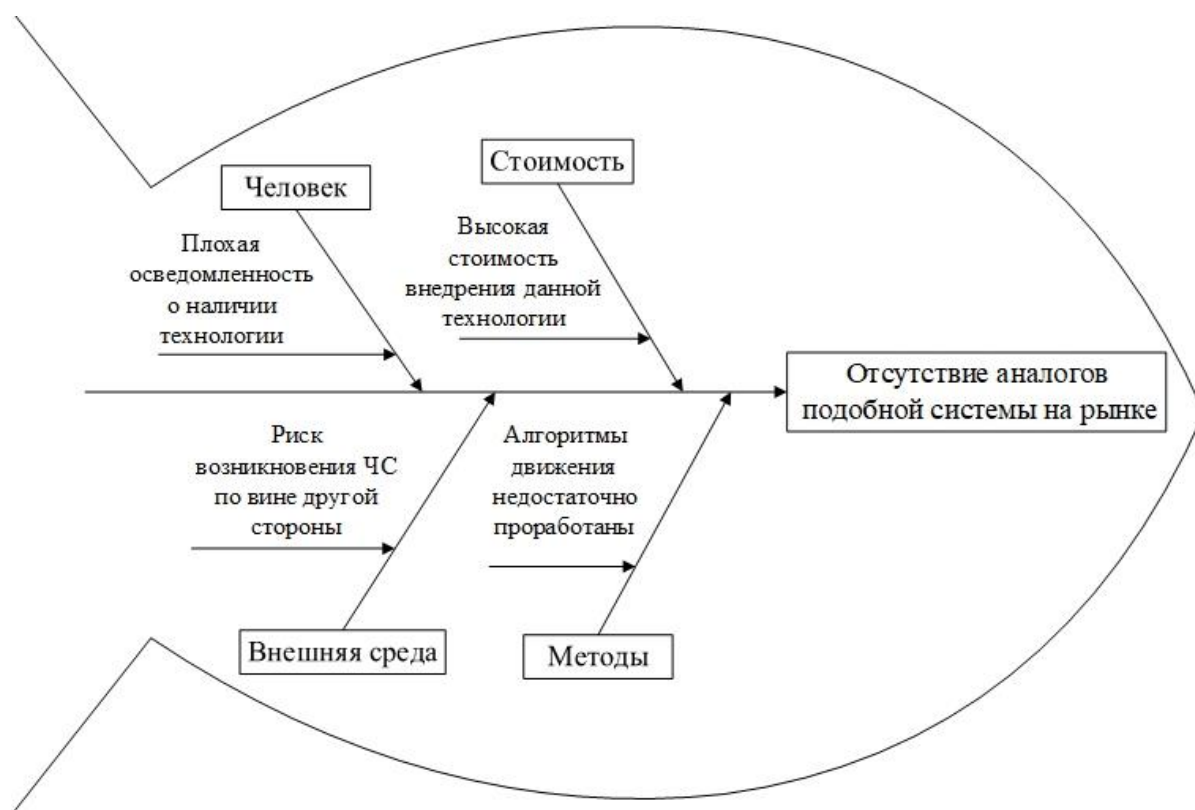


Рисунок 1. Диаграмма «Рыбий скелет» для данной предметной области

Диаграмма представляет собой графическое изображение, демонстрирующее определенные причины конкретных событий, действий, проблем и помогает сформулировать определенные выводы.

Проблема - отсутствие аналогов программного комплекса на рынке. Большинство компаний по разработке беспилотных автомобилей создают свои собственные полигоны для тестирования алгоритмов движения автомобиля. Создание таких полигонов занимает большое количество времени и требует больших затрат.

Цель - создание программного комплекса, который позволит проводить виртуальное испытание алгоритмов движения беспилотного автомобиля.

1.2. Преимущества и недостатки разработки ПК

Эра автомобилей-беспилотников наступит уже совсем скоро. Об этом говорят аналитики и эксперты. К этому стремятся и большинство производителей — такие, как Volvo, Mercedes, Honda, Nissan и Google. Однако беспилотные автомобили, как и любые технологии, имеют свои плюсы и минусы.

В таблице 1 приведены основные преимущества и недостатки внедрения беспилотного автомобиля в жизнь человека.

Таблица 1. Основные преимущества и недостатки внедрения беспилотного автомобиля

Преимущества	Недостатки
1.Транспортировка грузов	1. Потеря рабочих мест для людей
2.Экономичный расход топлива	2.Ненадежность программного обеспечения
3.Централизованное управление транспортом	3.Потеря контроля над автомобилем
4.Снижение цен на перевозку грузов и пассажиров в связи с экономией на заработной плате водителей	4. Ответственность за нанесение ущерба
5.Минимизация аварий и инцидентов на дорогах	5. Отсутствие опыта вождения в критических ситуациях (сильный дождь, снегопад, туман)
6.Возможность свободно передвигаться людям с ограниченными возможностями	6. Отсутствие опыта вождения в чрезвычайных ситуациях

Также необходимо выделить преимущества и недостатки разработки программного комплекса. Преимущества и недостатки данной системы представлены в таблице 2.

Таблица 2. Преимущества и недостатки разработки ПК

Преимущества	Недостатки
1. Возможность моделирования	1.Необходимость в

любой ситуации на дороге	квалифицированных специалистов
2. Уменьшение затрат на испытания автомобиля	2. Невозможность предугадать действия беспилотного автомобиля

1.3. Анализ требований

При анализе предметной области требования разделяют на функциональные (описывают то, что должна делать система) и нефункциональные (описывают то, каким образом эти требования должны выполняться).

Функциональные требования:

1. Возможность навигации по приложению (меню).

Меню должно включать в себя:

- возможность начать воспроизведение сцены ;
- возможность для настройки параметров системы (разрешение монитора, качество графики) ;
- краткая информация о приложении ;
- возможность выхода из приложения.

2. Создание моделей пешехода, автомобиля, дороги и других элементов.

3. Возможность пользовательских настроек.

- настройка погодных условий (снег, дождь) ;
- настройка времени дня (день/ночь).

4. Демонстрация алгоритмов движения:

- пешехода ;
- беспилотного автомобиля ;
- взаимодействие участников дорожного движения между собой.

5. Реализация алгоритмов работы светофоров на перекрестках по всему полигону.

Нефункциональные требования:

1. Разработка ПО должно выполняться в виде решения в среде Unity3D с использованием языка C# и JavaScript.

2. Интерфейс программы должен полностью быть на русском языке.

3. Окно пользователя не должно содержать более 7 управляющих элементов (кнопок, выпадающих списков и т.д.)

4. Модели различных зданий, пешеходов, беспилотных автомобилей и других элементов виртуального полигона могут создаваться в среде Blender и 3DMax.

Глава 2. Проектирование программного комплекса

Проектирование подразумевает выработку свойств и функций системы на основе анализа предметной области, а именно: моделей предметной области, требований к ПО. Модель предметной области накладывает ограничения на бизнес-логику и структуры данных разрабатываемого продукта.

В процессе проектирования ПО для выражения его характеристик используются различные нотации: блок-схемы, диаграммы классов, UML-диаграммы, диаграммы потоков данных.

2.1. Функции разрабатываемого комплекса

При составлении набора функций к данному продукту был применен метод MoSCoW. MoSCoW - это метод для задания приоритета функций в инкрементных и итерационных методах. Метод MoSCoW обеспечивает способ достижения общего понимания важности поставляемой функции. Все функции в журнале являются ценными, но зачастую не все из них могут быть поставлены в одно и то же время.

Все имеющиеся альтернативы необходимо определить в 4 группы («корзины»):

- М (must) — должно быть при любых условиях. Нельзя отказаться от реализации данной функции.
- S (should) – следовало бы иметь, но можно отложить на более позднее время.
- C (could) – можно было бы иметь, но, если нет возможности, можно и отложить на определённое время.
- W (won't this time) – в этот раз стоит отказаться от этого, но в следующий раз можно включить.

Данный метод представлен в таблице 3.

Таблица 3. Метод MoSCoW

MUST	<p>1.Навигация по приложению (Меню).</p> <p>2.Возможность изменения погодных условий во время работы приложения.</p> <p>3.Движение беспилотного автомобиля по прямой траектории.</p> <p>4.Движение пешехода по прямой траектории.</p> <p>5.Изменение времени дня (день/ночь) во время работы приложения.</p>
SHOULD	<p>1.Возможность включения и отключения фар автомобиля</p> <p>2.Наличие анимации движения у пешехода</p> <p>3.Корректная работа светофоров на перекрестках</p>
COULD	<p>1.Демонстрация поведения беспилотного автомобиля и пешехода в критических ситуациях (авария, плохие погодные условия и т.д.)</p> <p>2.Настройка качества графики</p>
WON'T	<p>1.Движение беспилотного автомобиля по карте полигона</p>

2.2. Диаграмма вариантов использования

Диаграмма прецедентов (диаграмма вариантов использования) в UML - диаграмма, отражающая отношения между актёрами и прецедентами.

Для отражения модели прецедентов на диаграмме используются:

- рамки системы – отделяет системы от её окружения;
- актер - обозначает набор ролей пользователя;
- прецедент - обозначает выполняемые системой действия.

В данной системе предусмотрено 3 актёра:

- пользователь ;
- пешеход ;
- беспилотный автомобиль.

На рисунке 2 представлена диаграмма варианты использования для рассматриваемого программного комплекса в нотации UML.

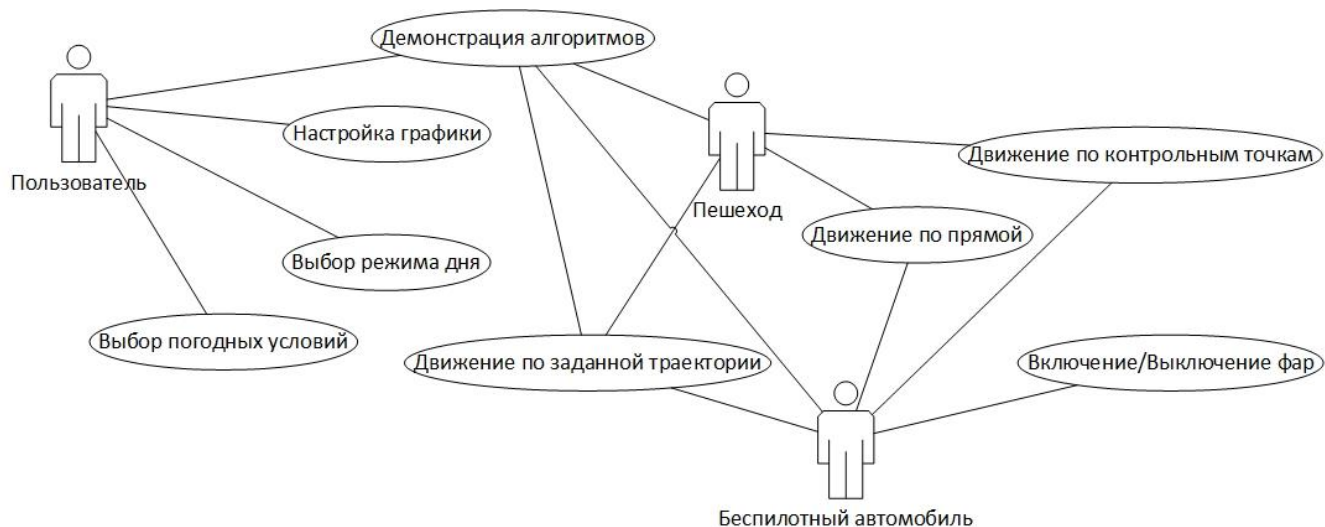


Рисунок 2. Диаграмма вариантов использования в нотации UML

2.3. Диаграмма классов

Диаграмма классов предназначена для демонстрации классов системы, их атрибутов (свойств) и связей между ними. На диаграмме классы изображены в рамках и состоят из следующих частей:

- имя класса;
- атрибут – данные об объекте;
- метод – определённая функция, которую должен выполнять данный класс.

В таблицах 4-12 приведены классы ПК и интерфейсы, которые будут реализованы.

Menu – класс, описывающий действия и параметры, с которыми может работать пользователь.

Таблица 4. Описание класса Menu.

Is_rain	«Включен» ли дождь на полигоне
Is_sun	«Включено» ли солнце на полигоне

Is_snow	«Включен» ли снег на полигоне
Is_night	«Включен» ли режим дня: ночь
Is_day	«Включен» ли режим дня: день
Get_info()	Получить информацию о продукте
Get_settings()	Просмотреть настройки приложения
Start()	Начать испытание

Road user – класс, описывающий действия участников дорожного движения (автомобиля и пешехода).

Таблица 5. Описание класса Road user.

ID	Идентификатор
Speed	Текущая скорость передвижения
Coordinate x	Координата x
Coordinate y	Координата y
Coordinate z	Координата z
Forward()	Движение вперед
Back()	Движение назад
Left()	Движение влево
Right()	Движение вправо

Weather – интерфейс, реализующий изменение погодных условий в виртуальном полигоне.

Таблица 6. Описание интерфейса Wheather.

Add_rain()	Дождливая погода
Add_snow()	Снежная погода
Add_sun()	Солнечная погода

Car – класс, описывающий автомобиль и его движение.

Таблица 7. Описание класса Car.

Max_speed()	Максимальная скорость автомобиля
Center_of_mass	Центр массы автомобиля

Pedestrian – класс, описывающий пешехода и его движение.

Таблица 8. Описание класса Pedestrian.

Average_speed	Средняя скорость передвижения
Change_direction()	Изменить направление движения

Wheel – класс, описывающий колесо автомобиля.

Таблица 9. Описание класса Wheel.

Wheel_radius	Радиус колеса
Move()	Движение колеса

Lights – класс, описывающий работу фар автомобиля.

Таблица 10. Описание класса Lights.

Is_turn	Включены ли фары автомобиля?
Turn_on()	Включить
Turn_off()	Выключить

Time of day – интерфейс, реализующий смену времени дня (день/ночь).

Таблица 11. Описание интерфейса Time of day.

Switch_day()	Смена времени дня
--------------	-------------------

Traffic lights – класс, описывающий работы светофоров в виртуальном полигоне.

Таблица 12. Описание класса Traffic lights.

Interval	Интервал переключения света светофора
Is_red	Горит ли красный свет?
Is_yellow	Горит ли желтый свет?
Is_green	Горит ли зеленый свет?

Switch()	Переключить свет
----------	------------------

На рисунке 3 представлена диаграмма классов в нотации UML для проектируемого программного комплекса.

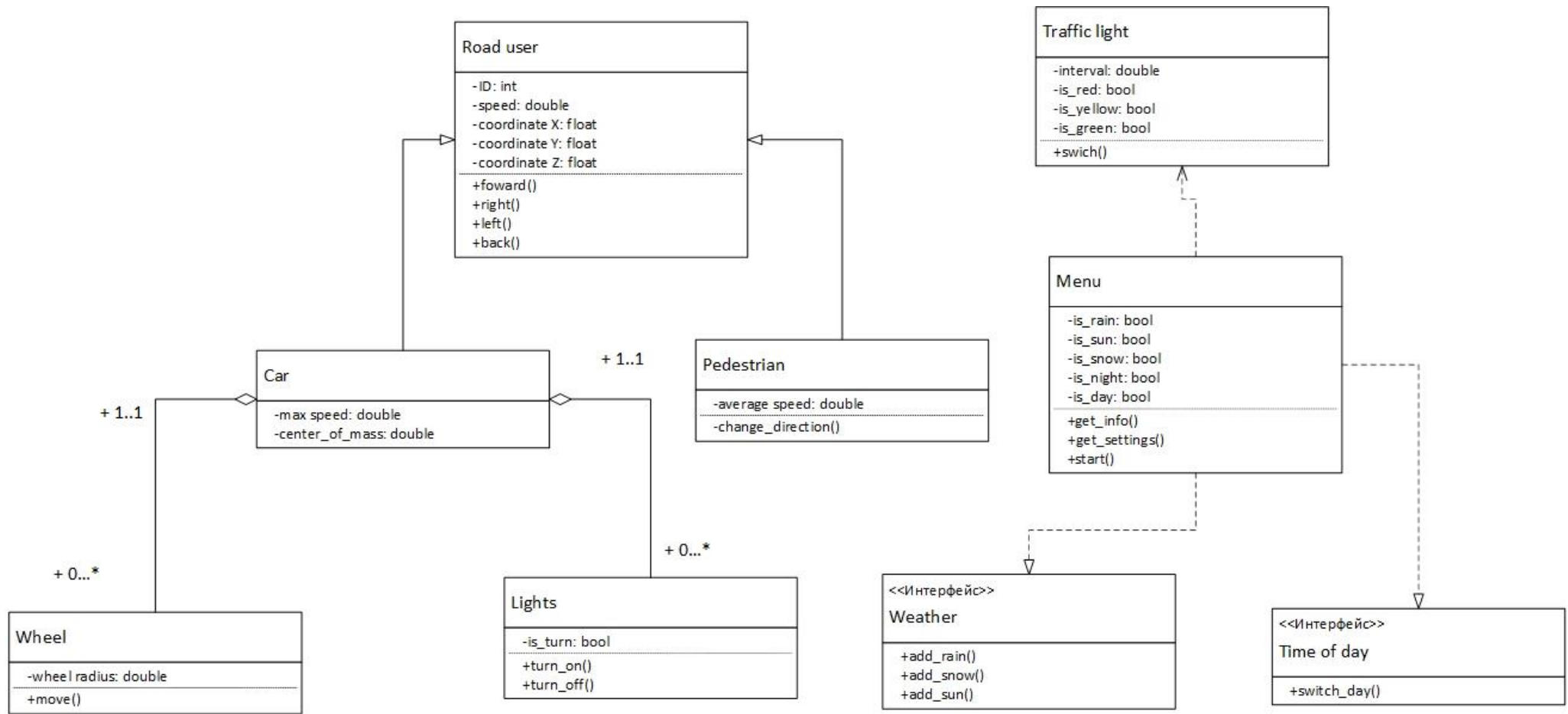


Рисунок 3. Диаграмма классов в нотации UML для программного комплекса

2.4. Диаграмма потоков данных

На рисунке 4 представлена контекстная диаграмма в нотации DFD. Уровень контекстной диаграммы определяет масштаб и границы системы.

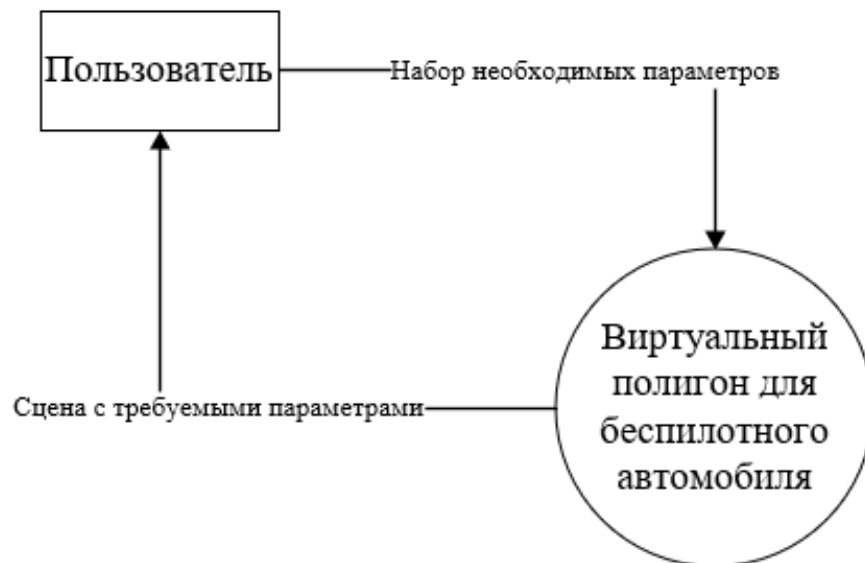


Рисунок 4. Контекстная диаграмма в нотации DFD для виртуального полигона

После составления контекстной диаграммы рассматриваемый комплекс необходимо разбить на модули, т.е. декомпозировать. На рисунке 5 представлена диаграмма функциональной декомпозиции модулей системы в нотации DFD.

Программный комплекс состоит из нескольких модулей:

1. Модуль виртуальной среды (работа с 3D графикой: добавление дорог, зданий и других моделей для наполнения полигона).
2. Модуль движения автомобиля (разработка алгоритмов движения автомобиля по полигону).
3. Модуль движения пешехода (разработка алгоритмов для движения пешехода по тротуарам, пешеходным переходам и т.д.).
4. Модуль управления светофорами (управление движением автомобиля и пешехода по виртуальному полигону с помощью алгоритмов).



Рисунок 5. Диаграмма функциональной декомпозиции в нотации DFD для виртуального полигона

2.5. Выбор инструмента для разработки

Для разработки данного программного комплекса была выбрана межплатформенная среда разработки компьютерных игр Unity3D.

Unity3D - это инструмент для разработки двух- и трёхмерных приложений, работающий под любыми операционными системами.

Основными преимуществами Unity являются наличие визуальной среды разработки, межплатформенной поддержки и модульной системы компонентов. К недостаткам относят работу с многокомпонентными схемами и подключение внешних библиотек.

Также еще одним преимуществом данного инструмента является возможность создавать приложения для запуска в браузерах с помощью специального подключаемого модуля Unity (Unity Web Player), а также с

помощью реализации технологии WebGL (Web-based Graphics Library — программная библиотека для языка программирования JavaScript).

Движок Unity3D поддерживает два сценарных языка: C#, и JavaScript (модификация). При разработке данного продукта будут использоваться оба языка программирования. Для реализации работы светофоров и других элементов полигона будет использоваться язык C#, для работы с алгоритмами движения пешехода и беспилотного автомобиля будет использоваться язык JavaScript.

Глава 3. Реализация программного комплекса

Для дальнейшей разработки полигона потребовалось решить следующие задачи:

- внедрение пешехода в систему полигона;
- движение пешехода по определенному маршруту;
- движение беспилотного автомобиля по определенному маршруту.

3.1. Обоснование выбора языка программирования

Для разработки виртуального полигона на платформе Unity3D была использована среда разработки MonoDeveloper и язык программирования JavaScript.

JavaScript является объектно-ориентированным языком, но используемое в языке прототипирование обуславливает отличия в работе с объектами по сравнению с традиционными класс-ориентированными языками. Кроме того, JavaScript имеет ряд свойств, присущих функциональным языкам — функции как объекты первого класса, объекты как списки, карринг, анонимные функции, замыкания — что придаёт языку дополнительную гибкость.

Несмотря на схожий с Си синтаксис, JavaScript по сравнению с языком Си имеет коренные отличия:

- объекты с возможностью интроспекции;
- функции как объекты первого класса;
- автоматическое приведение типов;
- автоматическая сборка мусора;
- анонимные функции.

В языке отсутствуют такие полезные вещи, как:

- стандартная библиотека: в частности, отсутствует интерфейс программирования приложений по работе с файловой системой, управлению потоками ввода-вывода, базовых типов для бинарных данных;

- стандартные интерфейсы к веб-серверам и базам данных;
- система управления пакетами, которая бы отслеживала зависимости и автоматически устанавливала их.

3.2. Создание виртуального полигона

Проект в Unity делится на сцены (уровни) — отдельные файлы, содержащие свои игровые миры со своим набором объектов, сценариев, и настроек. Сцены могут содержать в себе как, собственно, объекты (модели), так и пустые игровые объекты — объекты, которые не имеют модели.

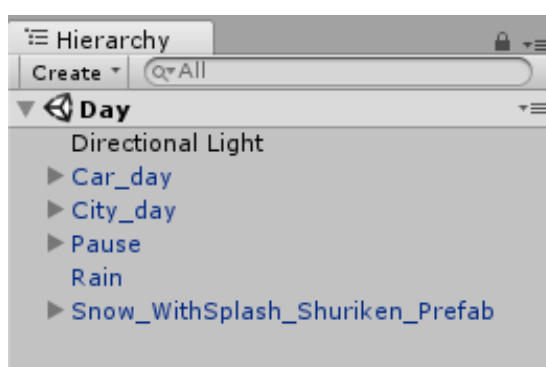


Рисунок 6. Пример сцен, использующихся в проекте

В данной работе используются сцены день/ночь с включением погодных условий (дождь/снег). Также присутствует навигационная сцена «Меню», «Настройки» (установка требуемых параметров графики).

Также в сценах используются объекты для построения требуемых моделей. Объекты содержат наборы компонентов, с которыми и взаимодействуют скрипты. Также у объектов есть название, может быть тег (метка) и слой, на котором он должен отображаться. Так, у любого объекта на сцене обязательно присутствует компонент Transform — он хранит в себе координаты местоположения, поворота и размеров объекта по всем трём осям. У объектов с видимой геометрией также по умолчанию присутствует компонент Mesh Renderer, делающий модель объекта видимой.

К объектам можно применять коллизии (в Unity так называются коллайдеры — collider). Существует несколько типов коллайдеров:

- Character controller — вид физической модели, созданный специально под использование его для игровых персонажей;
- Box collider (физическая модель образует куб, в который попадает вся модель объекта);
- Sphere collider (физическая модель образует сферу, в которую попадает вся модель объекта);
- Capsule collider (физическая модель образует капсулу, в которую попадает модель объекта. В отличие от предыдущего типа размеры можно менять и по одной, и по трём осям сразу);
- Mesh collider (физическая модель полностью повторяет реальную геометрию объекта);
- Wheel collider (физическая модель колеса);
- Terrain collider — тип физической модели, созданный специально для использования на объекте типа Terrain — земля, генерируемая редактором Unity с возможностями скульптинга и окрашивания местности.

Помимо пустого игрового объекта и моделей, на сцену можно добавлять ещё такие объекты типа GameObject:

- Камера;
- GUI текст;
- GUI текстура;
- 3D текст;
- Точечный свет;
- Направленный свет;
- Освещение территории;
- Источник света, имитирующий солнце;
- Стандартные примитивы;
- Деревья;
- Terrain (земля).

Некоторые объекты типа GameObject применялись в этом проекте, такие как Terrain (для создания ландшафта полигона), источник света, деревья, а также камера для управления автомобилем.

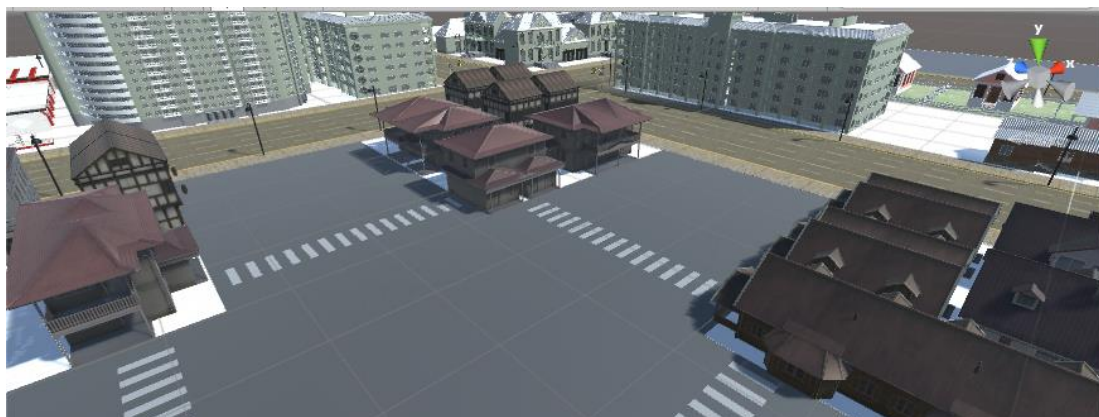


Рисунок 7. Результат создания виртуального полигона

3.3. Внедрение пешехода в систему виртуального полигона

Для создания модели и анимации пешехода использовался плагин Mecanim. Mecanim - это система анимации Unity. Эта система позволяет вам создавать «состояния», которые воспроизводят анимацию и определяют логику перехода, которая расскажет вашей игре или персонажу, как перемещаться между разными состояниями.

Основной задачей являлось не только верное конструирование модели пешехода, но и его анимации. Необходимое условие заключалось в том, чтобы создать скелет пешехода для использования анимации. Реализация данного этапа работы приведена на рисунке 7.

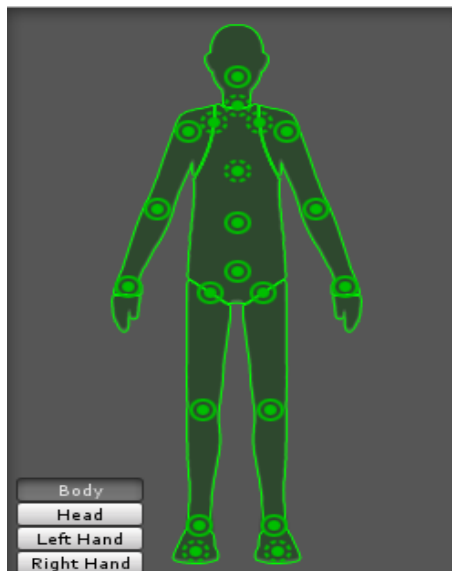


Рисунок 8. Составление скелета для пешехода

Следующим этапом являлось составление анимации. Для этого нужно добавить контроллер под названием Animator. Контроллер аниматора (Animator Controller) также требуется для применения анимации к скелету. Контроллер аниматора создается Unity и позволяет руководить набором анимаций для персонажа и переключаться между ними, когда выполняется некоторое условие. Например, можно переключиться от анимации походки к прыжку при нажатии клавиши пробела. Контроллер управляет переходами между анимациями, используя так называемую машину состояний (State Machine) - род некоторой программы написанной языком визуального программирования в Unity.

Для создания маршрута для пешехода предложено использовать набор пустых точек (Node) (см. рисунок 9). Точки объединяются родительским объектом - полоса движения (Point). Для построения маршрутов были выбраны точки, так как они самые легковесные и практически не несут нагрузку на вычислительные ресурсы.

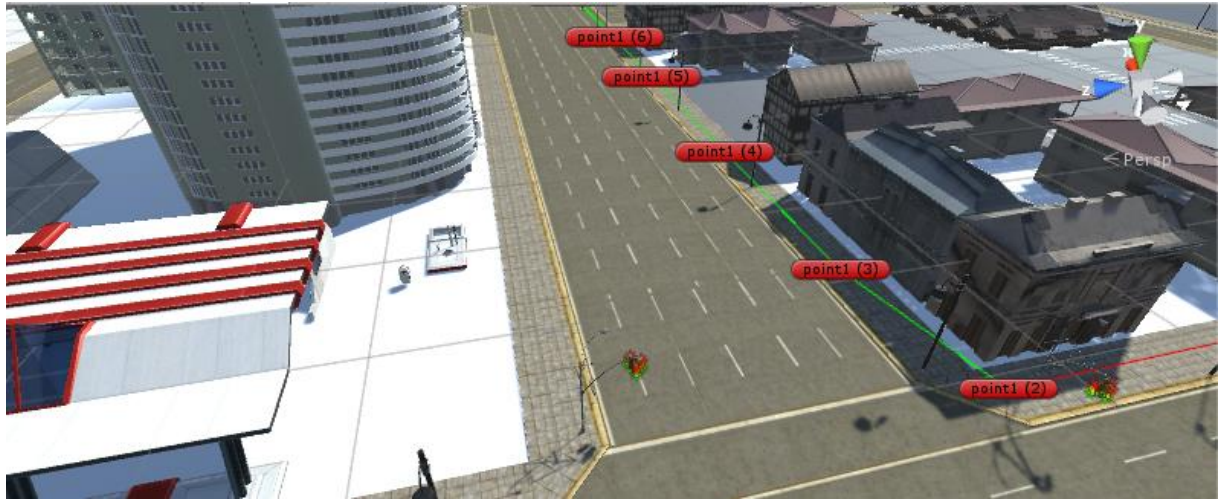


Рисунок 9. Набор точек (чекпоинтов)

На точки навешивается скрипт Waypoint.cs, который содержит информацию о точке: обязательна ли точка для возврата, является ли она точкой стыка, метка точки (start, end, center of rotation, none, constriction), ID, позиция, все возможные пути из этой точки.



Рисунок 10. Блок-схема алгоритма движения пешехода

Ниже продемонстрирована основная функция скрипта Waypoint.cs на языке JavaScript для построения необходимого маршрута:

```
function OnDrawGizmos () {
    if(isGrounded){ // если точка расположена на земле
        if (Physics.Raycast(transform.position, -Vector3.up, hit, tolerance)) {
            var y=hit.distance; transform.position.y -=y ;
            transform.position.y +=yOffset;
        }
    }
    var thispos = this.transform.position; // если совпадают по координатам
    Gizmos.DrawIcon (thispos, "Waypoint.tif"); // соединяем две точки между собой
    Gizmos.color = Color.white;
    Gizmos.DrawWireSphere (thispos, .3);
}
```

На пешехода навешивается скрипт FollowPath.cs также содержащий информацию о полосе. Основная функция скрипта – нарисовать в редакторе видимые линии для соединения точек необходимого маршрута.

Ниже продемонстрированы основные функции скрипта FollowPath.cs на языке JavaScript для построения необходимого маршрута:

- функция движения вперед:

```
function MoveForward(){
    FindDistance();    dir= target - transform.position;

    if(useSlerp){
        var rot: Quaternion = Quaternion.LookRotation(dir);
        transform.rotation = Quaternion.Slerp(transform.rotation, rot, speed *
Time.deltaTime);
    } else {
        transform.LookAt(target);
    }
    transform.Translate(Vector3(0, 0, speed * Time.deltaTime));
}
```

- функция для нахождения следующей точки (Point):

```
function NextWayPoint(){
if((curindex+1) < numpoints){
    curindex+=1;
}else{
    curindex=0;
}
if(usePingpong && curindex==0){

    //изменяем положение векторов
    var n=numpoints-1; var nn=(numpoints)/2;
    for(var a : int = 0; a <nn; a++){
        var tempVector=waypoints[a];
        waypoints[a] = waypoints[n-a];
        waypoints[n-a] = tempVector;
    }
    curindex=0; target=waypoints[curindex]; return;
}
// если нет цикла возвращаемся обратно
if(!closedLoop && curindex==0){
```

```

        curindex=1; target=waypoints[1];
        // переход к первой точки и выход из цикла
        transform.position=waypoints[0];
        transform.LookAt(target);
        return;
    }
    target=waypoints[curindex]; // если нет обновить маршрут
}

```

- функция нахождения расстояния между точками:

```

function FindDistance(){
var tempindex = curindex;
if((tempindex-1) >-0.5){
    tempindex-=1;
}else{
    tempindex=numpoints-1;
}
distofline = Vector3.Distance(target,waypoints[tempindex]);
disttotarget= Vector3.Distance(target,transform.position);
distnormal= disttotarget/distofline;
}

```

В результате реализации данной задачи достаточно определить маршрут, по которому следует двигаться пешеходу, и любой объект сможет двигаться по заданному пути.

3.4. Внедрение автомобиля в систему виртуального полигона

Для корректного движения беспилотного автомобиля необходимо определить его центр массы и правильный поворот колес во время движения.

```

public class WheelData
{
    public Transform wheelTransform;
    public WheelCollider col;
    public Vector3 wheelStartPos;
    public float rotation = 0.0f;
}

protected WheelData[] wheels;

private WheelData SetupWheels(Transform wheel, WheelCollider col) // устанавливаем
параметры колес автомобиля
{
    WheelData result = new WheelData();

    result.wheelTransform = wheel;
    result.col = col;
    result.wheelStartPos = wheel.transform.localPosition;

    return result;
}

private void UpdateWheels() // метод для обновления значений колес

```

```

{
    float delta = Time.fixedDeltaTime;

    foreach (WheelData w in wheels)
    {
        WheelHit hit;

        Vector3 lp = w.wheelTransform.localPosition;
        if (w.col.GetGroundHit(out hit))
        {
            lp.y -= Vector3.Dot(w.wheelTransform.position - hit.point,
transform.up) - wheelRadius;
        }
        else
        {
            lp.y = w.wheelStartPos.y - wheelOffset;
        }
        w.wheelTransform.localPosition = lp;

        w.rotation = Mathf.Repeat(w.rotation + delta * w.col.rpm * 360.0f / 60.0f,
360.0f);
        w.wheelTransform.localRotation = Quaternion.Euler(w.rotation, w.col.steerAngle,
0.0f);
    }
}

```

Следующий участок кода определяет методы для движения автомобиля:

```

private void CarMove(float accel, float steer) // метод для движения автомобиля
{
    foreach (WheelCollider col in WColForward)
    {
        col.steerAngle = steer * maxSteer;
    }
    if (Input.GetKeyUp("w"))
    {
        foreach (WheelCollider col in WColBack)
        {
            col.motorTorque = 0;
        }
    }
    if (accel == 0)
    {
        foreach (WheelCollider col in WColBack)
        {
            col.brakeTorque = maxBrake;
        }
    }
    else
    {
        foreach (WheelCollider col in WColBack)
        {
            col.brakeTorque = 0;
            col.motorTorque = accel * maxAccel;
        }
    }
}

```

```

    }
}
}

void FixedUpdate()// метод для обновления значений автомобиля
{
    bool brakeButton = Input.GetButton("Jump");

    float accel = 0;
    float steer = 0;

    accel = Input.GetAxis("Vertical");
    steer = Input.GetAxis("Horizontal");

    CarMove(accel, steer);
    foreach (WheelCollider col in WColBack)
    {
        if (brakeButton)
        {
            col.brakeTorque = maxBrake * 10000;
        }
    }
    UpdateWheels();
}

void Start() // метод для начала движения
{
    GetComponent<Rigidbody>().centerOfMass = centerOfMass.localPosition;

    wheels = new WheelData[WColForward.Length + WColBack.Length];

    for (int i = 0; i < WColForward.Length; i++)
    {
        wheels[i] = SetupWheels(wheelsF[i], WColForward[i]);
    }

    for (int i = 0; i < WColBack.Length; i++)
    {
        wheels[i + WColForward.Length] = SetupWheels(wheelsB[i], WColBack[i]);
    }
}

```

В ходе реализации данной задачи автомобиль может двигаться по виртуальному полигону.

3.5. Дальнейшие планы разработки

Следующим этапом разработки программного комплекса является отладка алгоритмов взаимодействия между автомобилем и пешеходом. Также в дальнейших планах числится увеличение участников дорожного движения для демонстрации корректной работы алгоритмов.

Для решения проблемы взаимодействия пешехода и беспилотного автомобиля понадобится построить карту путей игрового уровня.

Построение карты (графа) путей происходит единожды при запуске уровня. Для этого ищем все отмеченные определенным тегом объекты и привязываем к заданной нами координатной сетке. Для каждого объекта должна существовать заранее заданная карта его проходимости, размеченная при моделировании объекта. Получаем таким образом битовую карту проходимости.

Карта проходимости для игрового объекта также представляет собой битовую матрицу, привязанную к точке регистрации объекта. Для удобства точку регистрации объекта будем привязывать к ближайшему левому углу. Сам же объект при моделировании должен жестко привязываться к сетке, чтобы потом боты не бегали сквозь стены.

Строится граф следующим образом: сперва проверяется доступность соседней точки, т.е. смотрим, не находится ли соседняя точка в прямой видимости, и если нет, то применяем алгоритм нахождения пути A^* . Найдя путь через A^* , мы дополнительным проходом трассировки луча разбиваем его на отрезки, крайние точки которых находятся в прямой видимости друг от друга. Так мы избавимся от явной угловатости перемещений по клетчатой карте, вычисленной через A^* .

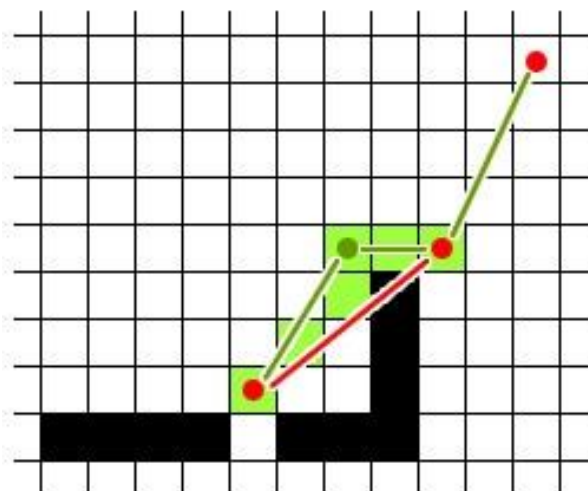


Рисунок 11. Построение карты(графа) путей

На рисунке красными точками показаны промежуточные чекпоинты; зеленые отрезки — это результат трассировки луча, дающий прямую видимость; зеленые квадраты — результат поиска пути через алгоритм A^* , т.к. между левой и средней точкой нет прямой видимости (красный отрезок); зеленая точка — дополнительная точка, дающая прямую видимость на пути, найденном через A^* .

Полученный граф должен быть еще немного модифицирован. Так, чтобы добраться из крайней правой в крайнюю левую точку, не нужно проходить среднюю красную. Можно сразу пройти в зеленую, а уже из нее в крайнюю левую красную. Но при этом важно помнить, что зеленая точка — это не одна из конечных точек, а только лишь промежуточная, поэтому ее нужно хранить отдельно.

При перемещении ботов из одной точки в другую неплохо было бы предусмотреть сглаживание движения, но это может повлечь неприятные последствия, когда бот одним боком проваливается в стену, которую пытается пройти. В этом случае лучше сглаживать траекторию только внутри одной клетки, но это уже может оказаться мало заметным визуально, а потому не имеет смысла.

Глава 4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

В работе осуществляется создание программного комплекса для тестирования алгоритмов движения беспилотного автомобиля в виртуальном полигоне. Целью данного раздела является экономическое обоснование данной научной разработки, а также определение и расчет трудовых и денежных затрат на её создание.

4.1. Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения

4.1.1. Потенциальные потребители результатов исследования

Для анализа потребителей результатов исследования необходимо рассмотреть целевой рынок и провести его сегментирование.

Целевой рынок – сегменты рынка, на котором будет продаваться в будущем разработка. В свою очередь, сегмент рынка – это особым образом выделенная часть рынка, группы потребителей, обладающих определенными общими признаками.

Сегментирование – это разделение покупателей на однородные группы, для каждой из которых может потребоваться определенный товар (услуга).

Предоставляемая услуга – тестирование алгоритмов движения беспилотного автомобиля в виртуальной среде.

Основные сферы применения данного программного комплекса:

- грузоперевозки;
- военная промышленность;
- гражданская авиация;
- сельское хозяйство.

Конкретные потребители данного продукта:

- предприниматели;
- армия;

- авиакомпании;
- сельское хозяйство;
- автотранспорт.

4.1.2. Анализ конкурентных технических решений

Детальный анализ конкурирующих разработок, существующих на рынке, необходимо проводить систематически, поскольку рынки пребывают в постоянном движении. Такой анализ помогает вносить коррективы в научное исследование, чтобы успешнее противостоять своим оппонентам. Важно реалистично оценить сильные и слабые стороны разработок конкурентов.

На момент проведения данного исследования конкурентные технические решения не были выявлены. Это может быть связано с весьма узким функционалом разрабатываемого программного обеспечения.

4.1.3. SWOT-анализ

SWOT – Strengths (сильные стороны), Weaknesses (слабые стороны), Opportunities (возможности) и Threats (угрозы) – представляет собой комплексный анализ научно-исследовательского проекта. SWOT-анализ применяют для исследования внешней и внутренней среды проекта.

Разработанная для данного исследования матрица SWOT представлена в таблице ниже.

SWOT-анализ используется для оценки факторов и явлений, влияющих на деятельность компании, а также на возникновение кризисных ситуаций. Для SWOT-анализа актуальны не все существующие на рынке возможности, а только те, которые можно использовать в данном случае. Преимущество SWOT-анализа заключается в том, что аналитическая работа не зациклена только на финансовом состоянии или на анализе конкурентов, а связывает разнообразные факторы внешней и внутренней среды воедино.

Таблица 13. Матрица SWOT-анализа

	Сильные стороны научно-исследовательского проекта: С1. Не требуется специализированного оборудования С2. Невысокие системные требования	Слабые стороны научно-исследовательского проекта: Сл1. Ограниченный функционал конечного ПО. Сл2. Малый опыт создания подобных систем.
Возможности: В1. Невысокий уровень конкуренции В2. Разработка применима для различных предприятий	Возможность захватить рынок в разных сферах до появления конкурентов.	Благодаря невысокому уровню конкуренции, расширение функционала не будет приоритетной задачей.
Угрозы: У1. Низкий спрос у потребителя У2. Появление конкурентов в данном виде услуг	Низкие требования могут привлечь клиентов. В случае появления конкурентов в качестве преимущества можно рассматривать опыт в данной работе и сформировать базу клиентов	Регулярная работа над проектом позволит найти новых клиентов, получить опыт и произвести новый функционал.

4.2. Организация и планирование работ

Для успешной организации процесса работы над конкретной задачей необходимо рационально спланировать занятость каждого из участников и сроки проведения отдельных этапов работы.

На данном этапе составляется полный список необходимых работ, назначаются их исполнители и продолжительность. Результатом планирования работ является линейный график реализации проекта.

Перечень этапов настоящей работы и продолжительность их выполнения в процентном соотношении для научного руководителя (НР) и исполнителя (И) представлены в таблице 14.

Таблица 14. Распределение занятости научного руководителя и исполнителя по этапам работы

Этапы работы	Исполнители	Загрузка исполнителей
Постановка целей и задач	НР	НР – 100 %
Разработка и утверждение ТЗ	НР, И	НР – 100 % И – 10 %
Подбор и изучение материалов по тематике	НР, И	НР – 40 % И – 100 %
Разработка календарного плана	НР, И	НР – 100 % И – 10 %
Обсуждение литературы	НР, И	НР – 30 % И – 100 %
Проектирование структуры ПО	НР, И	НР – 40 % И – 100 %
Разработка ПО	НР, И	НР – 30 % И – 100 %
Тестирование и отладка ПО	НР, И	НР – 30 % И – 100 %
Устранение проблем и оптимизация	И	И – 100 %
Оформление расчетно-пояснительной записки	И	И – 100 %
Оформление графического материала	И	И – 100 %

4.2.1. Продолжительность этапов работ

Для определения ожидаемой продолжительности работ $t_{ож}$ с помощью экспертных оценок были использованы следующие формулы:

$$t_{ож} = \frac{3t_{min} + 2t_{max}}{5} \quad (1)$$

$$t_{ож1} = \frac{3 \cdot 2 + 2 \cdot 4}{5} = 2.8 \quad (2)$$

(остальные значения рассчитаны по аналогии), где

t_{min} – минимальная продолжительность работ, дн.;

t_{max} – максимальная продолжительность работ, дн.

Длительность этапов в рабочих днях ТРД вычислялась по формуле:

$$T_{РД} = t_{ож} \cdot K_{Д} \quad (3)$$

$$T_{РД1} = 2,8 \cdot 1,2 = 3,36$$

(остальные значения рассчитаны по аналогии)

где K_d – коэффициент, учитывающий дополнительное время на компенсации и согласование работ ($K_d = 1,2$).

Расчет продолжительности этапа в календарных днях $T_{кд}$ ведется по формуле:

$$T_{кд} = T_{рд} \cdot T_k \quad (4)$$

$$T_{кд1} = 3,36 \cdot 1,244 = 4,11$$

(остальные значения рассчитаны по аналогии)

$T_{рд}$ – продолжительность выполнения этапа в рабочих днях;

T_k – коэффициент календарности.

Коэффициент календарности рассчитывается по формуле:

$$T_k = \frac{T_{кал}}{T_{кал} - T_{вд} - T_{пд}}, \quad (5)$$

где:

$T_{кал}$ – календарные дни ($T_{кал} = 366$);

$T_{вд}$ – выходные дни ($T_{вд} = 53$);

$T_{пд}$ – праздничные дни ($T_{пд} = 14$).

$$T_k = \frac{366}{366 - 53 - 14} \approx 1,244$$

Все расчеты по трудозатратам представлены в таблице 15. В ней итоги по продолжительности этапов работы в рабочих и календарных днях являются общими трудоемкостями для каждого из участников проекта. Далее они будут использованы для расчетов. Величины трудоемкости этапов по исполнителям $T_{кд}$ (данные столбцов 9 и 10 кроме итогов) позволяют построить линейный график осуществления проекта, приведенный на рисунке .

Таблица 15. Трудозатраты на выполнение проекта

Этап работы	Исполнители, %		Продолжительность работ, дни			Длительность работ, чел/дн.			
	НР	И	<i>t_{min}</i>	<i>t_{max}</i>	<i>t_{ож}</i>	ТР Д		ТК Д	
						НР	И	НР	И
1	2	3	4	5	6	7	8	9	10
Постановка целей задач	100	0	2	4	2,8	3,36	0	4,11	0
Разработка и утверждение технического задания (ТЗ)	100	10	2	5	3,2	3,84	0,38	4,7	0,47
Подбор и изучение материалов по тематике	40	100	12	16	13,6	6,53	16,32	7,99	19,98
Разработка календарного плана	100	10	2	4	2,8	3,36	0,34	4,11	0,41
Обсуждение литературы	30	100	4	7	5,2	1,87	6,24	2,29	7,64
Проектирование структуры ПО	40	100	11	16	13	6,24	15,6	7,64	19,09
Разработка ПО	30	100	20	25	22	7,92	26,4	9,69	32,31
Тестирование и отладка ПО	30	100	11	15	12,6	4,54	15,12	5,55	18,51
Устранение проблем и оптимизация	0	100	6	11	8	0	9,6	0	11,75
Оформление расчетно-пояснительной записки	0	100	7	14	9,8	0	11,76	0	14,39
Оформление графического материала	0	100	4	6	4,8	0	5,76	0	7,05
Итого:					97,8	37,656	107,52	46,09	131,60

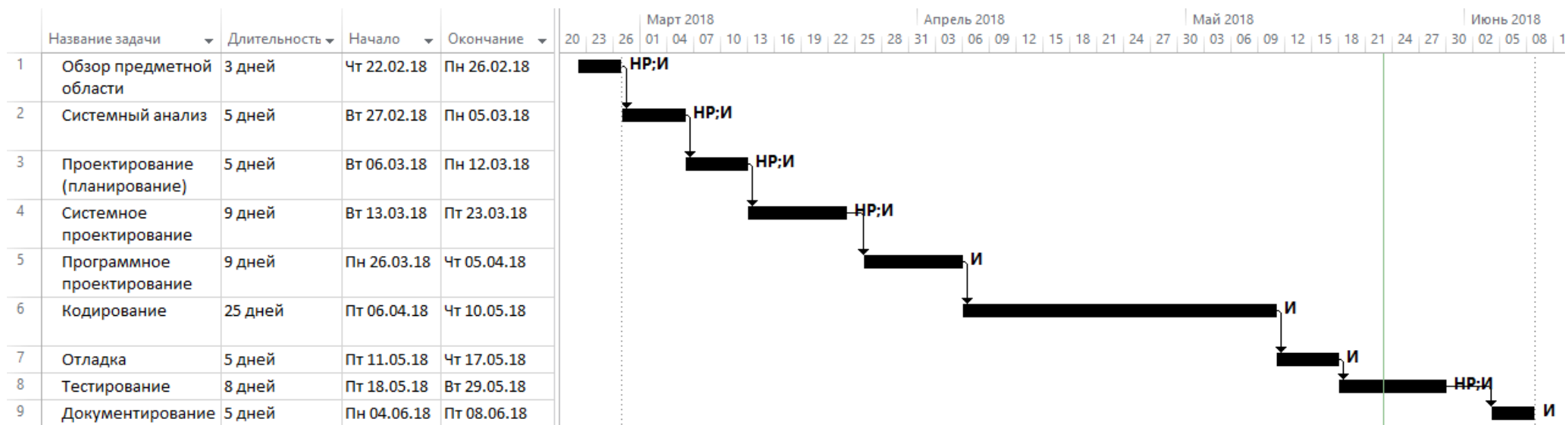


Рисунок 12. Линейный график работ для проекта

НР – Научный руководитель;

И – Исполнитель.

4.3. Расчет сметы затрат на выполнение проекта

Для проекта по созданию программного обеспечения поиска автомобилей на картинке по цвету производится оценка затрат по следующим статьям:

- материалы и покупные изделия;
- заработная плата;
- социальный налог;
- расходы на электроэнергию (без освещения);
- амортизационные отчисления;

Так как работа выполнялась без привлечения сторонних организаций и для ее выполнения не требовалась аренда какого-либо имущества, а также не было необходимости в командировках, расходы по соответствующим статьям отсутствуют.

4.3.1. Расчет затрат на материалы

В материальных затратах будут учтены только расходы на канцелярские принадлежности и картриджи для принтера, так как все необходимые для работы над проектом материалы имелись в распоряжении кафедры, на которой велась разработка. Материалы, необходимые для выполнения данной работы, и расчет материальных затрат представлены в таблице 16.

Таблица 16. Материальные затраты

Наименование материалов	Цена за ед., руб.	Кол-во	Сумма, руб.
Бумага для принтера формата А4	240	1 уп.	240
Ручка шариковая	20	2 шт.	40
Картридж	1600	1 шт.	1600
Блокнот	50	2 шт.	100
Итого:			1980

4.3.2. Расчет заработной платы

Расчет основной заработной платы выполняется на основе трудоемкости выполнения каждого этапа и величины месячного оклада исполнителя. Месячный оклад (МО) НР, занимающего должность доцента и имеющего степень кандидата технических наук, составляет 23264 руб./мес. МО исполнителя, являющегося студентом, составляет 2200 руб./мес.

Исходя из того, что в месяце в среднем 24,83 рабочих дня при шестидневной рабочей неделе среднедневная тарифная заработная плата (ЗП_{дн-т}) рассчитывается по формуле:

$$ЗП_{дн-т} = MO/24,83 \quad (6)$$

Расчеты затрат на полную заработную плату приведены в таблице 17. Затраты времени по каждому исполнителю в рабочих днях с округлением до целого взяты из таблицы трудозатрат. Для учета в ее составе премий, дополнительной зарплаты и районной надбавки используется следующий ряд коэффициентов: районный $K_p = 1,3$. Таким образом, для перехода от тарифной (базовой) суммы заработка исполнителя, связанной с участием в проекте, к соответствующему полному заработку (зарплатной части сметы) необходимо учесть районный коэффициент $K_p = 1,3$.

Таблица 17. Затраты на заработную плату

Исполнитель	Оклад, руб./мес.	Среднедневная ставка, руб./раб.день	Затраты времени, раб.дни	Р-й коэффиц иент	Фонд з/платы, руб.
НР	24400	936,93	38	1,3	46284,4
И	2200	88,6	108	1,3	12439,79
Итого:					58724,19

4.3.1. Расчет затрат на отчисления во внебюджетные фонды

Отчисления во внебюджетные фонды включают в себя отчисления в пенсионный фонд, на социальное и медицинское страхование и составляют 30% от заработной платы участников проекта, стипендия не учитывается.

$C_{\text{соц}}$ определяется следующим образом:

$$C_{\text{соц.}} = C_{\text{зп}} \cdot 0,3 = 46284,4 \cdot 0,3 = 13885,32 \text{ руб.}$$

4.3.2. Расчет затрат на электроэнергию

Данный вид расходов включает в себя затраты на электроэнергию при работе оборудования, а именно компьютера и принтера. Затраты на электроэнергию при работе оборудования $C_{\text{эл.об.}}$ рассчитываются по формуле:

$$C_{\text{эл.об.}} = P_{\text{об}} \cdot Ц_{\text{э}} \cdot t_{\text{об}}, \quad (7)$$

где $P_{\text{об}}$ – мощность, потребляемая оборудованием, кВт;

$Ц_{\text{э}}$ – тарифная цена за 1 кВт·час;

$t_{\text{об}}$ – время работы оборудования, час.

Мощность $P_{\text{об}}$, потребляемая оборудованием, определяется по формуле:

$$P_{\text{об}} = P_{\text{ном.}} \cdot K_{\text{с}}, \quad (8)$$

где $P_{\text{ном.}}$ – номинальная мощность оборудования, кВт;

$K_{\text{с}}$ – коэффициент загрузки (для технологического оборудования малой мощности $K_{\text{с}} = 1$).

Номинальная мощность персонального компьютера составляет 0,3 кВт, принтера – 0,1 кВт. Для ТПУ с учетом налога на добавленную стоимость (НДС) $Ц_{\text{э}} = 5,257 \text{ руб./кВт·час}$.

Время работы оборудования $t_{\text{об}}$ для исполнителя вычисляется на основе данных таблицы трудозатрат:

$$t_{\text{об}} = T_{\text{рд}} \cdot K_{\text{т}}, \quad (9)$$

где $K_t \leq 1$ – коэффициент использования оборудования по времени, равный отношению времени его работы в процессе выполнения проекта к ТРД.

Из расчета, что продолжительность рабочего дня равна 8 часов, а работа выполнялась 108 рабочих дней, получим, что общее время выполнения проекта составляет 864 часа.

Так как работа на компьютере проводилась по 7 часов в день из 8, то $K_t = 0,88$. Тогда из 864 часов, потраченных исполнителем на осуществление проекта, 760 часов были проведены за компьютером. Принтер использовался примерно в течении 15 часов. Затраты на электроэнергию при работе оборудования сведены в таблицу 18.

Таблица 18. Затраты на электроэнергию для технологических целей

Наименование оборудования	Время работы оборудования $t_{об}$, час	Потребляемая мощность $P_{об}$, кВт	Затраты Эоб, руб.
Персональный компьютер	760	0,3	1198,60
Лазерный принтер	15	0,1	7,89
Итого:			1206,48

4.3.5. Расчет амортизационных расходов

В статье «Амортизационные отчисления» рассчитывается амортизация используемого оборудования за время выполнения работы. Амортизационные отчисления рассчитываются по времени использования компьютера по формуле:

$$C_{ам} = ((N_A * C_{об}) / F_d) * t_{рф} * n, \quad (10)$$

где:

N_A – годовая норма амортизации;

$C_{об}$ – цена оборудования;

F_d – действительный годовой фонд рабочего времени;

$t_{рф}$ – время работы вычислительной техники;

n – число задействованных единиц оборудования, $n = 1$.

Годовая амортизация H_A определяется как величина, обратная сроку амортизации оборудования C_A , который определяется согласно постановлению правительства РФ «О классификации основных средств, включенных в амортизационные группы». Для компьютера примем $C_A = 3$ года, тогда $H_A = 0,33$. Для принтера примем $C_A = 2$ года, тогда $H_A = 0,5$. Расчет затрат на амортизационные отчисления представлен в таблице 19.

Таблица 19. Затраты на амортизационные отчисления

Наименование оборудования	Норма амортизации оборудования H_A	Стоимость оборудования $C_{об}$, руб.	Фактич. время работы оборудования $t_{рф}$, ч	Действ. годовой фонд раб.времени F_d , ч	Амортизац. отчисления $C_{ам}$, руб.
Персональный компьютер	0,33	45000	760	2384	4734,06
Итого:					4734,06

4.3.6. Расчет прочих расходов

В данном разделе производится оценка расходов на выполнение проекта, которые не были учтены в предыдущих статьях, оплата услуг связи, копирование. материалов и др. Величина прочих расходов составляет 10% от суммы всех предыдущих затрат и вычисляется по следующей формуле:

$$C_{\text{проч}} = 0,1 * (C_{\text{мат}} + C_{\text{зп}} + C_{\text{соц}} + C_{\text{эл.об.}} + C_{\text{ам}}), \quad (11)$$

$$C_{\text{проч}} = 0,1 * (1980 + 58724,19 + 13885,32 + 1206,48 + 4734,06) = 8053.$$

Таким образом, накладные расходы составили 8053 руб.

4.3.7. Расчет общей себестоимости разработки

Общая стоимость разработки по созданию программного обеспечения поиска машин на картинке по цвету определяется путем суммирования затрат по всем статьям и представлена в таблице 20.

Таблица 20. Смета затрат на разработку проекта

Статья затрат	Условное обозначение	Сумма, руб.
Материалы и покупные изделия	$C_{\text{мат}}$	1980
Основная заработная плата	$C_{\text{зп}}$	58724,19
Отчисления в социальные фонды	$C_{\text{соц}}$	13885,32
Расходы на электроэнергию	$C_{\text{эл.об}}$	1206,48
Амортизационные отчисления	$\mathcal{E}_{\text{ам}}$	4734,06
Прочие расходы	$C_{\text{проч}}$	8053
Итого:		88583,05

Общая себестоимость проекта получилась равной 88583,05 рубля.

4.4. Выводы по главе

Проведено комплексное описание и анализ финансово-экономических аспектов выполненной работы.

Составлен перечень проводимых работ, их исполнителей и продолжительность выполнения этапов работ, составлен линейный график.

Рассчитана смета затрат на выполнение проекта, проведен расчет себестоимости и прибыли проекта.

Определены показатели эффективности проекта и проведена оценка его эффективности.

Глава 5. Социальная ответственность

В данной работе освещен комплекс мер организационного, правового, технического и режимного характера, которые минимизируют негативные последствия разработки программного комплекса, а также рассматриваются вопросы техники безопасности, охраны окружающей среды и пожарной профилактики, даются рекомендации по созданию оптимальных условий труда.

Объектом исследования выступает рабочее место программиста, разрабатывающего программный комплекс, который позволит проводить тестирование алгоритмов движения беспилотного автомобиля в виртуальном полигоне.

Рабочей зоной при разработке данного программного комплекса является учебная аудитория в Кибернетическом центре ТПУ, оборудованная системой отопления, кондиционирования воздуха, с естественным и искусственным освещением.

Рабочее место – стационарное, оборудованное персональным компьютером и оргтехникой.

5.1. Производственная безопасность

По природе возникновения вредные и опасные производственные факторы делятся на 4 группы:

- физические;
- химические;
- психофизические;
- биологические.

В данном случае биологические и химические факторы существенного влияния на состояние здоровье человека не оказывают, поэтому в данной главе подробнее будут рассмотрены лишь физические и психофизические факторы.

К вредным производственным факторам, при работе с компьютером следует отнести:

- повышенный уровень электромагнитных излучений;
- повышенный уровень шума;
- слабая освещённость рабочей зоны;
- отклонение микроклиматических условий;
- умственное перенапряжение.

К опасным производственным факторам, при работе с компьютером следует отнести:

- опасность поражения электричеством;
- возможность возникновения короткого замыкания;
- статическое электричество.

К вредным психофизиологическим факторам также относится умственное перенапряжение.

5.1.1. Электромагнитное излучение

Видеотерминалы (ВДТ) и системные блоки, как любые электрические приборы производят электромагнитное излучение, воздействие которого на организм человека напрямую зависит от напряжённостей электрического, магнитного поля, от потока энергии, частоты колебаний, а также от размера самого облучаемого тела.

Нарушения, возникающие в организме человека при воздействии электромагнитных полей низкой напряжённости, носят обратимый характер. Однако если напряжённость магнитных полей выше предельного допустимого уровня, то страдают сердечно-сосудистая и нервная системы, а также органы ЖКТ.

Большая часть электромагнитных излучений происходит не от экрана монитора, а от системного блока и видеокабеля. Современные машины выпускаются со специальной металлической защитой внутри системного блока для уменьшения фона электромагнитного излучения.

Согласно гигиеническим требованиям к персональным электронно-вычислительным машинам на расстоянии 50 см вокруг ВДТ напряженность электромагнитного поля по электрической составляющей должна быть не более:

25 В/м при частоте в диапазоне 5 Гц ÷ 2 кГц;

2,5 В/м при частоте в диапазоне 2 кГц ÷ 400 кГц.

Плотность магнитного потока не должна превышать:

250 нТл при частоте в диапазоне 5 Гц ÷ 2 кГц;

25 нТл при частоте в диапазоне 2 кГц ÷ 400 кГц.

Основные способы защиты от электромагнитного поля:

1) увеличить расстояние от источника, экран монитора не должен находиться ближе 50 см от пользователя;

2) использование приэкранного фильтра, специального экрана, а также других средств индивидуальной защиты, которые прошли испытание в аккредитованных лабораториях и имеют соответствующий гигиенический сертификат.

На рабочем месте уровень электромагнитного излучения не превышает допустимых норм, регламентированных СанПиН 2.2.2/2.4.1340-03. Для минимизации вредного влияния электромагнитного излучения на организм время работы за компьютером сокращено и чередуется с временем отдыха.

5.1.2. Уровень шума

Шум ухудшает условия труда, оказывают вредное воздействие на организм человека. Действие шума различно: он затрудняет разборчивость речи, вызывает снижение работоспособности, повышает утомляемость, вызывает необратимые изменения в органах слуха человека, ослабляет внимание, ухудшает память, снижает реакция, увеличивает число ошибок при работе.

Источником шума в учебных помещениях, эксплуатирующих вычислительную технику, являются сами вычислительные машины

(встроенные в стойки ЭВМ вентиляторы, принтеры и т.д.), центральная система вентиляции и кондиционирования воздуха и другое оборудование.

Характеристикой такого шума на рабочих местах являются уровни звукового давления в децибелах в октавных полосах частот со среднегеометрическими частотами 31,5; 63; 125; 250; 500; 1000; 2000; 4000; 8000 Гц, определяемые по формуле:

$$L=20 \times \lg(P/P_0), \quad (12)$$

где P – среднеквадратичная величина звукового давления, Па;

$P_0 = 2 \times 10^{-5}$ Па – исходное значение звукового давления в воздухе.

Шум от работающего компьютера создаёт $P = 0,05$ Па. Таким образом, $L = 68$ дБА.

Согласно требований ГОСТ 12.1.003- 2014 для рабочих мест в помещениях «...проектно-конструкторских бюро, расчётчиков, программистов вычислительных машин, в лабораториях для теоретических работ и обработки данных...» предельно допустимый уровень звукового давления составляет $L_{\max} = 75$ дБА. Иными словами, шум, создаваемый работой компьютеров в аудитории, по своим характеристикам удовлетворяет санитарным нормам.

В качестве мер по снижению шума применяют:

- подавление шума в источниках;
- звукоизоляция и звукопоглощение;
- увеличение расстояния от источника шума;
- проверка технического состояния и ремонт системного блока и принтера;
- рациональный режим труда и отдыха.

5.1.3. Освещенность рабочего места

Для обеспечения нормативных условий работы необходимо провести оценку освещенности рабочей зоны в соответствии с СанПиН 2.2.1/2.1.1.1278-03.

Правильное освещение рабочих мест и помещений является важным условием для создания безопасных и благоприятных условий труда.

Все поле зрения должно быть освещено равномерно – это является основным гигиеническим требованием. Другими словами, уровень естественного освещения рабочего места и яркость дисплея компьютера должны быть приблизительно одинаковыми, т.к. яркий свет в зоне периферийного зрения заметно увеличивает глазное напряжение, что приводит к их быстрой утомляемости.

Для создания равномерной освещённости рабочих мест светильники с люминесцентными лампами встраиваются непосредственно в потолок помещения и располагаются в равномерно-прямоугольном порядке. Наиболее желательное расположение светильников в непрерывный сплошной ряд вдоль длинной стороны помещения.

Чтобы поддерживать освещение в помещении по всем соответствующим нормам, необходимо хотя бы два раза в год мыть стекла и светильники, а также по мере необходимости заменять перегоревшие лампы.

В соответствии с СанПиН 2.2.1/2.1.1.1278-03 рекомендуемая искусственная освещённость для помещений, в которых выполняются работы высокой точности на светлом фоне при большом контрасте, составляет 400 лк с допустимым отклонением от -10% до +20%.

Таким образом, согласно СанПиН 2.2.1/2.1.1.1278-03 реальная освещённость в аудитории Кибернетического центра, где выполнялась разработка дипломного проекта, соответствует нормативным значениям освещенности рабочей зоны.

5.1.4. Отклонение микроклимата

Оптимальное состояние воздушной среды должно обеспечивать ощущение теплового комфорта в течение 8-часового рабочего дня, не вызывать отклонений в состоянии здоровья. Энергетические затраты организма измеряются в ккал/ч (Вт) и по затраченной энергии работы разделяются на категории. Так работа программиста относится к категории Ia – интенсивность энергозатрат до 120 ккал/ч (до 139 Вт) [8].

Работы производятся в основном сидя и сопровождаются незначительным физическим напряжением. Допустимые параметры микроклимата на рабочем месте для категории Ia приведены в таблице 21.

Таблица 21. Оптимальные величины показателей микроклимата на рабочих местах производственных помещений по СанПиН 2.2.2/2.4.1340-03.

Период года	Кат. работ по уровню энергозатрат, Вт	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	Ia	22-24	21-25	40-60	0,1
Теплый	Ia	23-35	22-26	40-60	0,1

Параметры микроклимата в помещении, где находится рабочее место, регулируются системой центрального отопления, кондиционером и естественной вентиляцией, и имеют следующие значения: влажность 40%, скорость движения воздуха 0,1 м/с, температура летом 23 – 25°С, зимой 21 – 23°С, что полностью соответствует нормам.

К мероприятиям по оздоровлению воздушной среды в производственном помещении относятся: правильная организация вентиляции и кондиционирования воздуха, отопление помещений. Вентиляция может осуществляться естественным и механическим путём.

5.1.5. Умственное перенапряжение

Для видов трудовой деятельности устанавливается 3 категории тяжести и напряженности работы с компьютером, которые определяются: для группы А - по суммарному числу считываемых знаков за рабочую смену (не более 60000 знаков за смену); для группы Б - по суммарному числу считываемых или вводимых знаков за рабочую смену (не более 40000 знаков за смену); для группы В - по суммарному времени непосредственной работы с компьютером за рабочую смену (не более 6 часов за смену).

При 8-часовой рабочей смене и работе на ПК регламентированные перерывы следует устанавливать:

Для третьей категории работ — через 1,5- 2,0 часа от начала рабочей смены и через 1,5-2,0 часа после обеденного перерыва продолжительностью 20 минут каждый или продолжительностью 15 минут через каждый час работы.

5.1.6 Опасность поражения электрическим током

Согласно СанПиН 2.2.2/2.4.1340-03. «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы», рабочее место должно находиться в безопасной зоне, которое не характеризуется наличием таких условий, как повышенная влажность (относительная влажность воздуха длительно превышает 75%), высокая температура (более 35°C), токопроводящая пыль, токопроводящие полы, возможность одновременного соприкосновения к имеющим соединения с землей металлическим элементам и металлическим корпусам электрооборудования.

Электрические установки, к которым относится ПК, представляют для человека большую потенциальную опасность, так как в процессе эксплуатации или проведения профилактических работ человек может коснуться комплектующих компьютера, находящихся под напряжением.

Специфическая опасность – корпуса ПК и прочего оборудования, оказавшегося под напряжением в результате повреждения или пробоя

изоляции, не подают каких-либо сигналов, которые предупреждают человека об опасности. Причинами электропоражений являются: провода с поврежденной изоляцией, розетки сети без предохранительных кожухов.

Для защиты от поражения электрическим током все токоведущие части защищены от случайных прикосновений кожухами, корпус устройства должен быть заземлен. Заземление выполняется изолированным медным проводом сечением 1.5 мм, который присоединяется к общей шине заземления с общим сечением 48 мм при помощи сварки. Общая шина присоединяется к заземлению, сопротивление которого не должно превышать 4 Ом.

Рабочее место программиста исследователя оборудовано таким образом, чтобы исключить взаимное соприкосновение кабелей и шнуров питания соседних компьютеров.

К организационно-техническим мероприятиям относится первичный инструктаж по технике безопасности. Первичный инструктаж по технике безопасности является обязательным условием для допуска к работе в данном помещении.

5.1.7. Возможность возникновения короткого замыкания

Короткие замыкания в электропроводке чаще всего происходят из-за нарушения изоляции токопроводящих частей в результате механического повреждения, старения, воздействия влаги и агрессивных сред, а также неправильных действий людей. При возникновении короткого замыкания возрастает сила тока, а количество выделяющейся теплоты, как известно, пропорционально квадрату тока.

Тепловое воздействие на изоляцию проводов резко снижает ее механические и диэлектрические свойства. Тепловое старение изоляции наиболее часто возникает из-за перегрузки электросетей токами, превышающими длительно допустимые для данного вида и сечений проводников. Тепловому разрушению подвержены и полимерные изоляционные материалы.

Пожарная опасность коротких замыканий электропроводки характеризуется следующими возможными проявлениями электрического тока: воспламенением изоляции проводов и окружающих горючих предметов и веществ; способностью изоляции проводов распространять горение при поджигании ее от посторонних источников зажигания; образованием при коротком замыкании расплавленных частиц металла, поджигающих окружающие горючие материалы.

Еще в начале развития электротехники появилась проблема защиты электрических устройств от чрезмерных токовых нагрузок, в том числе и короткого замыкания. Наиболее простым решением стала установка плавких предохранителей, которые перегорали от их нагревания вследствие превышения тока определенной величины.

Такие плавкие вставки функционируют и в настоящее время. Их основным достоинством является надежность, простота и невысокая стоимость. Однако имеются и недостатки. Простая конструкция предохранителя побуждает человека после сгорания плавкого элемента заменить его самостоятельно подручными материалами в виде скрепок, проволочек и даже гвоздей.

Такая защита не способна обеспечить необходимой защиты от короткого замыкания, так как она не рассчитана на определенную нагрузку. На производстве для отключения цепей, в которых возникло замыкание, используют электрические автоматы. Они намного удобнее обычных плавких предохранителей, не требуют замены сгоревшего элемента. После устранения причины замыкания и остывания тепловых элементов, автомат можно просто включить, тем самым подав напряжение в цепь.

Существуют также более сложные системы защиты в виде дифференциальных автоматов. Они имеют высокую стоимость. Такие устройства отключают напряжение цепи в случае наименьшей утечки тока. Такая утечка может возникнуть при поражении работника током.

5.1.8. Статическое электричество

Электрический ток оказывает на человеческий организм биологическое, электролитическое и термическое воздействие. Биологическое выражается в раздражении и возбуждении живых клеток организма, что приводит к непроизвольным судорожным сокращениям мышц, нарушению нервной системы, органов дыхания и кровообращения. При этом могут наблюдаться обмороки, потеря сознания, расстройство речи, судороги, нарушения дыхания (вплоть до остановки). При тяжелой электротравме смерть может наступить мгновенно. Электролитическое воздействие проявляется в разложении плазмы крови и других органических жидкостей, что может привести к нарушению их физико-химического состава. Термическое воздействие сопровождается ожогами участков тела и перегревом отдельных внутренних органов, вызывая в них различные, функциональные расстройства. Возникающая электрическая дуга вызывает местные повреждения тканей и органов человека.

Особую опасность для разработчика является накопленное статическое электричество на корпусе персонального ЭВМ. Разряд от статического электричества не представляет собой большую опасность, так как сила тока мала для сильного поражения от электролитического и термического воздействия. Искровой разряд статического электричества человек ощущает как толчок или судорогу, мышцы попавшие под разряд статического тока сокращаются. При таком разряде человек может сделать непроизвольные движение, приводящие к травме.

Для снижения величин токов статического электричества используются нейтрализаторы, местное и общее увлажнение воздуха, полы с антистатической пропиткой.

5.2. Экологическая безопасность

Разрабатываемый проект не имеет влияния на окружающую среду, так как сам продукт разрабатывается и используется внутри персональных

компьютеров, которые могут стать источниками различных угроз для загрязнения окружающей среды.

Согласно ГОСТ Р 56397-2015 «Техническая экспертиза работоспособности радиоэлектронной аппаратуры, оборудования информационных технологий, электрических машин и приборов. Общие требования» пункт 5.8.1, после проведения технической экспертизы, если оборудование не поддается ремонту, то оно признается неработоспособным и рекомендуется к списанию (замене); в случае отказа оборудования и нецелесообразности его ремонта и модернизации даются рекомендации о необходимости его списания и утилизации.

Согласно «Методики проведения работ по комплексной утилизации вторичных драгоценных металлов из отработанных средств вычислительной техники», утвержденной Государственным Комитетом РФ по телекоммуникациям от 19 октября 1999 г. В п.3.1.3. «Технология разборки универсальных ЭВМ» расписаны 4 этапа разборки и подготовки к утилизации внутренних частей ПК.

Люминесцентные лампы представляют собой «чрезвычайно опасные» виды отходов. Содержание ртути в любых люминесцентных лампах составляет от трех до пяти миллиграмм ртути. С учетом этого необходимо обеспечивать определенные условия хранения, их эксплуатации и утилизации. Согласно санитарным нормам хранить ртутосодержащие отходы необходимо в специальных герметичных контейнерах, доступ посторонним лицам к таким контейнерам должен быть запрещен. Транспортировка ламп на полигоны складирования должна выполняться организациями, которые специализируются на утилизации опасных отходов и имеют лицензию на данный вид деятельности.

5.3. Безопасность в чрезвычайных ситуациях

К наиболее вероятным ЧС можно отнести следующие: пожар (взрыв) в здании, авария на коммунальных системах жизнеобеспечения, землетрясение. Источниками возгорания может стать электропроводка,

внутренние работающие устройства ПК, взрывоопасные предметы в помещении исследователя согласно ГОСТ 12.1.044-89 «Система стандартов безопасности труда. Пожаровзрывоопасность веществ и материалов. Номенклатура показателей и методы их определения».

Превентивными мерами по предупреждению ЧС могут служить системы звукового и визуального оповещения персонала лаборатории и кабинетов об опасности, обучение персонала, методам работы с компьютером, наличие средств пожаротушения и информационных досок с планами эвакуации.

В случае угрозы возникновения ЧС необходимо отключить электропитание, вызвать по телефону пожарную команду, эвакуировать людей из помещения согласно плану эвакуации. При наличии небольшого очага пламени можно воспользоваться подручными средствами с целью прекращения доступа воздуха к объекту возгорания. В качестве подручных средств можно использовать углекислотные огнетушители ОУ-5 высокого давления с зарядом жидкой двуокиси углерода (по ГОСТ 8050-85), расположение которых можно найти на плане эвакуации людей при пожаре и других ЧС из помещения Кибернетического центра ТПУ.

5.4. Правовые и организационные вопросы обеспечения безопасности

Рабочие места должны соответствовать требованиям ГОСТ 12.2.032-78 «ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования» и ГОСТ 12.2.061-81 «ССБТ. Оборудование производственное. Общие требования безопасности к рабочим местам».

Работа с применением персональных компьютеров сопряжена со значительными зрительными и нервно-психологическими нагрузками, что повышает требования к организации труда пользователей ПК. Конструкция рабочей мебели должна обеспечивать возможность индивидуальной регулировки, соответственно росту работающего, и создавать удобную позу. Часто используемые предметы труда и органы управления должны находиться в оптимальной рабочей зоне. Конструкция рабочего стола должна

обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования с учетом его количественных и конструктивных особенностей, а также характера выполняемой работы. Высота рабочей поверхности стола должна регулироваться в пределах 680-800 мм, при отсутствии такой возможности его высота должна быть не менее 725 мм.

Конструкция рабочего стула должна обеспечивать поддержание рациональной рабочей позы при работе, что позволит изменять позу для снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения развития утомления.

Согласно требований СанПиН 2.2.2/2.4.1340 – 03 при организации работы на ПЭВМ должны выполняться следующие условия:

- персональный компьютер (ПК), и, соответственно, рабочее место должно располагаться так, чтобы свет падал сбоку, лучше слева;

- расстояние от ПК до стен должно быть не менее 1 м, поэтому по возможности следует избежать расположения рабочего места в углах помещения либо лицом к стене;

- ПК лучше установить так, чтобы, подняв глаза от экрана, можно было увидеть какой-нибудь удаленный предмет в помещении или на улице. Перевод взгляда на дальнее расстояние является одним из наиболее эффективных способов разгрузки зрительного аппарата при работе на ПК;

- окна в помещениях с ПЭВМ должны быть оборудованы регулируемыми устройствами (жалюзи, занавески, внешние козырьки и т.д.);

- монитор, клавиатура и корпус компьютера должны находиться прямо перед оператором; высота рабочего стола с клавиатурой должна составлять 680 – 800 мм над уровнем пола; а высота экрана (над полом) –900–1280см;

- монитор должен находиться от оператора на расстоянии 60 – 70 см на 20 градусов ниже уровня глаз;

- рабочее кресло должно иметь мягкое сиденье и спинку, с регулировкой сиденья по высоте, с удобной опорой для поясницы;

-положение тела пользователя относительно монитора должно соответствовать направлению просмотра под прямым углом или под углом 75 градусов.

5.5. Вывод по главе

В данной главе были проанализированы опасные и вредные факторы труда разработчика программного комплекса, а также предложены меры защиты от них, оценены условия труда рабочей зоны; рассмотрены требования по технике безопасности, электробезопасности, пожарной безопасности, экологической безопасности. В результате анализа было установлено, что аудитория Кибернетического центра ТПУ удовлетворяет всем требованиям, предъявляемым к нему нормативными документами в области охраны труда и окружающей природной среды.

Заключение

В данной работе был разработан программный комплекс, позволяющий производить тестирование алгоритмов движения беспилотного автомобиля в виртуальном полигоне. В ходе работы был проведен анализ предметной области, выявлены основные проблемы, построены диаграммы для понимания процессов, протекающих в программном комплексе.

В процессе разработки были решены следующие задачи:

- создание виртуального полигона;
- разработка анимации и алгоритма движения пешехода;
- разработка алгоритма управления системой светофоров на полигоне.

Среда разработки Unity3D является оптимальным решением для разработки данного программного комплекса, так как не требует материальных затрат. Данный инструмент разработки удобен в использовании и открывает различные возможности перед разработчиком. Следующим этапом разработки программного комплекса является отладка взаимодействий пешехода и беспилотного автомобиля.

Список публикаций

1. Журбич Н.И., Беспилотный автомобиль в среде Unity3D/ Журбич Н.И., Исаев А.С./ Технологии Microsoft в теории и практике программирования. - 2016. - №3. - С. 82-84.

Список использованных источников

1. Япония потратит более \$25 млрд на беспилотные автомобили [Электронный ресурс]. Федеральное агентство новостей. URL: <https://riafan.ru/789196-yaponiya-potratit-bolee> (дата обращения: 15.03.2018 г.).
2. Беспилотные автомобили интересуют больше половины россиян [Электронный ресурс]. 3D news. URL: <https://3dnews.ru/947327> (дата обращения: 28.03.2018).
3. Диаграмма вариантов использования. [Электронный ресурс]. Informicus. URL: http://www.informicus.ru/default.aspx?SECTION=6&id=73&subdivision_id=4 (дата обращения: 26.03.2018).
4. DFD методология. [Электронный ресурс]. На заметку. URL: <https://e-educ.ru/bd14.html> (дата обращения: 01.04.2018).
5. Метод «Диаграмма Исикавы». [Электронный ресурс]. Центр креативных технологий. URL: <https://www.inventech.ru/pub/methods/metod-0019/> (дата обращения: 15.02.2018).
6. СанПиН 2.2.2/2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы: с изменениями от 3 сентября 2010 г. – М.: Информационно-издательский центр Минздрава России, 2003.
7. СанПиН 2.2.4.548-96 Гигиенические требования к микроклимату производственных помещений. – М.: Информационно-издательский центр Минздрава России, 1997.
8. ГОСТ Р 55090-2012 Ресурсосбережение. Обращение с отходами. Рекомендации по утилизации отходов бумаги.
9. Инструкция по охране труда для работников на персональных электронно-вычислительных машинах (ПЭВМ). Киров: ГБОУ ВПО Кировская ГМА Минздрава России, 2012. – 6 с.
10. ГОСТ 12.2.061-81. Система стандартов безопасности труда. Оборудование производственное. Общие требования безопасности к рабочим местам.

11. ГОСТ 12.1.044-89. Система стандартов безопасности труда. Пожаровзрывоопасность веществ и материалов. Номенклатура показателей и методы их определения.
12. СанПиН 2.2.4.3359-16. Санитарно-эпидемиологические требования к физическим факторам на рабочих местах.
13. ГОСТ 12.3.031-83. Система стандартов безопасности труда. Работы со ртутью. Требования безопасности.
14. ГОСТ Р 56397-2015. Техническая экспертиза работоспособности радиоэлектронной аппаратуры, оборудования информационных технологий, электрических машин и приборов. Общие требования.
15. СанПиН 2.2.1/2.1.1.1278-03. Гигиенические требования к естественному, искусственному и совмещенному освещению жилых и общественных зданий.
16. ГОСТ 12.1.005-88 Система стандартов безопасности труда (ССБТ). Общие санитарно-гигиенические требования к воздуху рабочей зоны (с Изменением N 1).
17. СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы.

Приложение А

MainMenu.cs

```
var Play = false;
var Settings = false;
var Author = false;
var Quit = false;
//Меню настроек
var Back = false;

//Графика
var Low = false;
var Medium = false;
var High = false;
var Fantastic = false;
var BackM = false;

var Day= false;
var Night = false;

function OnMouseEnter() {
    GetComponent.<Renderer>().material.color = Color.blue;
}

function OnMouseExit() {
    GetComponent.<Renderer>().material.color = Color.white;
}

function OnMouseUp() {
    if (Play == true) {
        Application.LoadLevel(4);
    }
    if(Settings == true) {
        Application.LoadLevel(3);
    }

    if (Author == true) {
        Application.LoadLevel(5);
    }

    if (Quit == true) {
        Application.Quit();
    }
    if(Back == true) {
        Application.LoadLevel(0);
    }

    if(Low == true){
        QualitySettings.currentLevel = QualityLevel.Simple;
    }

    if(Medium == true){
        QualitySettings.currentLevel = QualityLevel.Good;
    }

    if(High == true){
        QualitySettings.currentLevel = QualityLevel.Beautiful;
    }

    if(Fantastic == true){
        QualitySettings.currentLevel = QualityLevel.Fantastic;
    }
}
```

```

}

//
if(BackM == true) {
    Application.LoadLevel(0);
}
if(Day == true) {
    Application.LoadLevel(1);
}
if(Night == true) {
    Application.LoadLevel(2);
}
}

```

Приложение Б

GamePause.cs

```

using UnityEngine;
using System.Collections;

public class GamePause : MonoBehaviour
{
    public float timing;
    public bool isPaused;
    public GameObject menu;

    void Start()
    {
    }

    void Update()
    {
        Time.timeScale = timing;

        if (Input.GetKeyDown(KeyCode.Escape) && isPaused == false)
        {
            isPaused = true;
        }
        else if (Input.GetKeyDown(KeyCode.Escape) && isPaused == true)
        {
            isPaused = false;
        }

        if (isPaused == true)
        {
            timing = 0;
            menu.SetActive(true);
        }
        else if (isPaused == false)
        {
            timing = 1;
            menu.SetActive(false);
        }
    }

    public void ResumeButton(bool state)
    {
        isPaused = state;
    }
}

```

```

public void ToMenuButton()
{
    Application.LoadLevel(0);
}
public void QuitButton()
{
    Application.Quit();
}
}

```

Приложение В

Lights.cs

```

using UnityEngine;
using System.Collections;

public class TR : MonoBehaviour {
    public Light Red ;
    public Light Green;
    public Light Yellow ;
    void Start() {
        StartCoroutine(Gren());
    }

    IEnumerator Gren() {
        Yellow.enabled = false;
        Green.enabled = true;
        Red.enabled = false;
        yield return new WaitForSeconds(15);
        StartCoroutine(Yellow1());
    }

    IEnumerator Yellow1() {
        Yellow.enabled = true;
        Green.enabled = false;
        Red.enabled = false;
        yield return new WaitForSeconds(5);
        StartCoroutine(Red1());
    }

    IEnumerator Red1() {
        Yellow.enabled = false;
        Green.enabled = false;
        Red.enabled = true;
        yield return new WaitForSeconds(15);
        StartCoroutine(Yellow2());
    }

    IEnumerator Yellow2() {
        Yellow.enabled = true;
        Green.enabled = false;
        Red.enabled = false;
        yield return new WaitForSeconds(5);
        StartCoroutine(Gren());
    }
}

public class TRBack : MonoBehaviour {
    public Light Red ;
    public Light Green;
    public Light Yellow ;
    void Start() {

```

```

        StartCoroutine(Red1());
    }

    IEnumerator Gren() {
        Yellow.enabled = false;
        Green.enabled = true;
        Red.enabled = false;
        yield return new WaitForSeconds(15);
        StartCoroutine(Yellow1());
    }

    IEnumerator Yellow1() {
        Yellow.enabled = true;
        Green.enabled = false;
        Red.enabled = false;
        yield return new WaitForSeconds(5);
        StartCoroutine(Red1());
    }

    IEnumerator Red1() {
        Yellow.enabled = false;
        Green.enabled = false;
        Red.enabled = true;
        yield return new WaitForSeconds(15);
        StartCoroutine(Yellow2());
    }

    IEnumerator Yellow2() {
        Yellow.enabled = true;
        Green.enabled = false;
        Red.enabled = false;
        yield return new WaitForSeconds(5);
        StartCoroutine(Gren());
    }
}

```

Приложение Д

Headlight.cs

```

using UnityEngine;
using System.Collections;

public class Fari : MonoBehaviour {
    public Light light1;
    public Light light2;
    public Light light3;
    public Light light4;
    public Light light5;
    public Light light6;

    // Use this for initialization
    void Start () {

        light1.enabled = false;
        light2.enabled = false;

    }

    // Update is called once per frame
    void FixedUpdate ()
    {
        bool brakeButton = Input.GetButton("Jump");
        if (Input.GetKeyDown("1"))

```

```
{
    light1.enabled = true;
    light2.enabled = true;
}
if (Input.GetKeyDown("k"))
{
    light1.enabled = false;
    light2.enabled = false;
}
if (brakeButton)
{
    light3.intensity = 8;
    light4.intensity = 8;
    light5.intensity = 8;
    light6.intensity = 8;
}
else
{
    light3.intensity = 1;
    light4.intensity = 1;
    light5.intensity = 1;
    light6.intensity = 1;
}
}
```