TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

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

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

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Разработка системы искусственного интеллекта для управления объектами двумерной игровой сцены

УДК 004.896:004.925.84:004.428

Студент

Группа	ФИО	Подпись	Дата
8К81	Диденко Назар Иванович		

Руководитель ВКР

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

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

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

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Страший преподаватель ООД ШБИП ТПУ	Мезенцева Ирина Леонидовна			

ДОПУСТИТЬ К ЗАЩИТЕ:

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

Томск – 2022 г.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ООП

Код результата	Результат обучения (выпускник должен быть готов)
Р1	Применять базовые и специальные естественнонаучные и математические знания в области информатики и вычислительной техники, достаточные для комплексной инженерной деятельности.
Р2	Применять базовые и специальные знания в области современных информационных технологий для решения инженерных задач.
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.
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.
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.
Р6	Внедрять, эксплуатировать и обслуживать современные программно-аппаратные комплексы, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять требования по защите окружающей среды.
Р7	Использовать базовые и специальные знания в области проектного менеджмента для ведения комплексной инженерной деятельности.
Р8	Владеть иностранным языком на уровне, позволяющем работать в иноязычной среде, разрабатывать документацию, презентовать и защищать результаты комплексной инженерной деятельности.
Р9	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.
Р10	Демонстрировать знания правовых, социальных, экономических и культурных аспектов комплексной инженерной деятельности.
Р11	Демонстрировать способность к самостоятельной к самостоятельному обучению в течение всей жизни и непрерывному самосовершенствованию в инженерной профессии.

TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

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

Школа: Инженерная школа информационных технологий и робототехники
Направление подготовки: 09.03.04 «Программная инженерия»
Отделение школы: 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**
8К81	Диденко Назару Ивановичу

Тема работы:

Разработка системы искусственного интеллекта для управления объектами двумерной игровой сцены	
Утверждена приказом директора (дата, номер)	

Срок сдачи студентом выполненной работы:	

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе *(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).*	Целью работы является создание системы искусственного интеллекта для управления объектами двумерной игровой сцены, созданной с помощью библиотеки Astrobox.

Перечень подлежащих исследованию, проектированию и разработке вопросов *(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).*	1. Исследование предметной области 2. Проектирование и реализация программных модулей 3. Демонстрация результатов разработки 4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение 5. Социальная ответственность
Перечень графического материала *(с точным указанием обязательных чертежей)*	1. Диаграмма процесса проектирования и реализации модулей 2. Карта сегментирования рынка 3. Рейтинг дронов среди разработчиков 4. Оценочная карта для сравнения конкурентных технических решений 5. Матрица SWOT анализа 6. Скриншоты работы системы 7. График Ганта 8. Расчет бюджета затрат НТИ 9. Динамика рейтинга DidenkoDrone

Консультанты по разделам выпускной квалификационной работы

(с указанием разделов)

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

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	

Задание выдал руководитель:

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

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К81	Диденко Назар Иванович		

TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

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

Школа: Инженерная школа информационных технологий и робототехники

Направление подготовки: Программная инженерия

Уровень образования: Бакалавр

Отделение школы: Отделение информационных технологий

Период выполнения: осенний / весенний семестр 2021/2022 учебного года

Форма представления работы:

бакалаврская работа

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

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	16 июня 2022 г.

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
30.04.2022	Раздел 1. Исследование предметной области	15
7.05.2022	Раздел 2. Проектирование и реализация программных модулей	20
14.05.2022	Раздел 3. Демонстрация результатов разработки	25
21.05.2022	Раздел 4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	20
28.05.2022	Раздел 5. Социальная ответственность	20

Составил преподаватель:

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

СОГЛАСОВАНО:

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

ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
8К81	Диденко Назар Иванович

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

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:	
1. *Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих*	*Оклад руководителя – 60000 руб.* *Оклад инженера –35000 руб.*
2. *Нормы и нормативы расходования ресурсов*	*Премиальный коэффициент руководителя 30%;* *Премиальный коэффициент инженера 20%;* *Доплаты и надбавки руководителя 30%;* *Доплаты и надбавки руководителя 30%;* *Дополнительной заработной платы 12%;* *Накладные расходы 16%;* *Районный коэффициент 1,3%.*
3. *Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования*	*Коэффициент отчислений на уплату во внебюджетные фонды 30,2 %*
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. *Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения*	*Определение потенциального потребителя результатов исследования, SWOT-анализ разработанной стратегии*
2. *Планирование и формирование бюджета научных исследований*	*Определение структуры работы. Расчет трудоемкости выполнения работ. Подсчет бюджета исследования*
3. *Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования*	*Рассчитать показатели финансовой эффективности, ресурсоэффективности и эффективности исполнения*
Перечень графического материала *(с точным указанием обязательных чертежей)***:**	
1. *Оценка конкурентоспособности технических решений* 2. *Матрица SWOT* 3. *Альтернативы проведения НИ* 4. *График проведения и бюджет НИ* 5. *Оценка ресурсной, финансовой и экономической эффективности НИ*	

Дата выдачи задания для раздела по линейному графику	04.03.2022

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН	Гасанов Магеррам Али оглы	д.э.н., профессор		04.03.2022

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К81	Диденко Назар Иванович		04.03.2022

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО		
8К81	Диденко Назар Иванович		
Школа	**Инженерная школа информационных технологий и робототехники**	**Отделение (НОЦ)**	**ОИТ**
Уровень образования	Бакалавриат	**Направление/ специальность**	09.03.04 Программная инженерия

Тема ВКР:

Разработка системы искусственного интеллекта для управления объектами двумерной игровой сцены	
Исходные данные к разделу «Социальная ответственность»:	
Введение − Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика) и области его применения. − Описание рабочей зоны (рабочего места) при разработке проектного решения/при эксплуатации	*Объект исследования:* класс “Drone” python библиотеки astrobox *Область применения:* искусственный интеллект, информационный системы *Рабочая зона:* офис *Размеры помещения:* 5 м * 3 м *Количество и наименование оборудования рабочей зоны:* Персональный компьютер 1 шт. *Рабочие процессы, связанные с объектом исследования, осуществляемые в рабочей зоне:* исследование возможностей модуля, корректировка модели поведения объектов на игровой сцене, создание искусственного интеллекта, соответствующего требованиям
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Правовые и организационные вопросы обеспечения безопасности при разработке проектного решения: − специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; − организационные мероприятия при компоновке рабочей зоны.	ГОСТ 12.2.032-78 ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования. ГОСТ 21889-76. Система «человек-машина». Кресло человека-оператора. Общие эргономические требования. ГОСТ Р ИСО 14738-2007. Безопасность машин. Антропометрические требования при проектировании рабочих мест машин. Рациональная организация труда в течение рабочего времени предусмотрена Трудовым Кодексом РФ ФЗ-197
2. Производственная безопасность при разработке проектного решения: − Анализ выявленных вредных и опасных производственных факторов	**Опасные факторы:** 1. Производственные факторы, связанные с повышенным уровнем ионизирующих излучений; 2. Производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которого попадает работающий; **Вредные факторы:** 1. Повышенный уровень общей вибрации; 2. Повышенный уровень локальной вибрации; 3. Повышенный уровень шума;

	4. Отсутствие или недостаток необходимого искусственного освещения; 5. Монотонность труда, вызывающая монотонию; 6. Длительное сосредоточенное наблюдение. **Требуемые средства коллективной и индивидуальной защиты от выявленных факторов:** Соблюдение техники безопасности и распорядка дня на рабочем месте. **Расчет:** расчет системы искусственного освещения
3. Экологическая безопасность при разработке проектного решения	Разработка проектного решения экологически безопасна, однако может косвенно влиять на атмосферу, так как работа компьютера связана с потреблением электроэнергии и нагревом аппаратных средств.
4. Безопасность в чрезвычайных ситуациях при разработке проектного решения	**Возможные ЧС:** Природные катастрофы (наводнения, цунами, ураган и т.д.); Геологические воздействия (землетрясения, оползни, обвалы, провалы территории и т.д.); Техногенные аварии (короткое замыкание в сети подключения компьютера, пожар) **Наиболее типичная ЧС:** Короткое замыкание, пожар
Дата выдачи задания для раздела по линейному графику	**01.03.2022**

Задание выдал консультант:

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

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К81	Диденко Назар Иванович		19.04.2022

Реферат

Выпускная квалификационная работа содержит 80 страниц, 8 рисунков, 22 таблиц и 8 литературных источников.

Ключевые слова: python, искусственный интеллект.

Цель работы: разработка системы искусственного интеллекта для управления объектами двумерной игровой сцены на языке Python, библиотеки Astrobox.

В процессе выполнения работы использовался Python.

Первая глава содержит ознакомительная информация о библиотеке astrobox и ее компонентах, описаны ее основные классы и подходы к работе с ними.

Вторая глава содержит процесс проектирования логики искусственного интеллекта игровых объектов.

Третья глава содержит результаты разработки, описаны основные поведенческие паттерны при нахождении на игровой сцене.

Четвертая глава содержит анализ ресурсоэффективности, оценку коммерческого потенциала, план выполнения работы.

Пятая глава содержит правовые и организационные вопросы обеспечения безопасности.

Список терминов и сокращений

1. Python – это высокоуровневый язык программирования общего назначения, который используется в том числе и для разработки веб-приложений.
2. BPMN (англ. Business Process Model and Notation, нотация и модель бизнес-процессов) — система условных обозначений для моделирования бизнес-процессов.
3. Player versus player или PvP — вариант интерактивного конфликта между несколькими пользователями, происходящий в игровом мире между двумя или более персонажами, управляемыми игроками-людьми.
4. Player versus environment или PvE - вариант интерактивного конфликта между пользователем/пользователями и игровыми событиями, в игровом мире.
5. Искусственный интеллект (ИИ) - свойство интеллектуальных систем выполнять творческие функции, которые традиционно считаются прерогативой человека; наука и технология создания интеллектуальных машин, особенно интеллектуальных компьютерных программ.
6. Элириум - внутриигровой ресурс. Основная цель в игре от Astrobox - его добыча.
7. Геймдев - разработка игр.
8. IDE - комплекс программных средств, используемый программистами для разработки программного обеспечения (ПО).

Оглавление

Реферат **9**

Список терминов и сокращений **10**

Введение **13**

Глава 1. Исследование предметной области **14**

1.1. Описание предметной области 14

1.2 Специфика Astrobox 15

1.3. Описание используемого средства разработки 16

1.4. Описание разрабатываемых модулей 17

1.5. Вывод по главе 17

Глава 2. Проектирование и реализация программных модулей **18**

2.1. Необходимый функционал 19

2.2. Описание процесса анализа 19

2.3. Данные, которыми оперирует ИИ 20

2.4. Проектирование логики класса 21

2.5. Вывод по главе 21

Глава 3. Демонстрация результатов разработки **22**

Глава 4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение **25**

4.1 Потребители результатов исследования 25

4.2 Анализ конкурентных технических решений 26

4.3 SWOT-анализ 29

4.4 Планирование работ по научно-техническому исследованию 32

4.4.1 Структура работ в рамках научного исследования 32

4.4.2 Определение трудоемкости выполнения работ 33

4.4.3 Разработка графика проведения научного исследования 34

4.5 Бюджет научно-технического исследования (НТИ) 39

4.5.1 Расчет материальных затрат НТИ 39

4.5.2 Расчет стоимости спецоборудования и амортизационных затрат 40

4.5.3 Основная заработная плата исполнителя темы 42

4.5.4 Расчет дополнительной заработной платы 45

4.5.5 Отчисления во внебюджетные фонды 46

4.5.6 Накладные расходы 46

4.5.7 Формирование бюджета затрат научно-исследовательского проекта 47

4.6. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования 48

Вывод по разделу 50

Глава 5. Социальная ответственность 52

Введение 52

5.1. Правовые и организационные вопросы обеспечения безопасности 53

5.2. Производственная безопасность 55

5.3. Анализ опасных и вредных факторов и обоснование мероприятий по снижению их воздействия 56

5.3.1. Производственные факторы, связанные с электромагнитными излучениями 56

5.3.2. Монотонность труда, вызывающая монотонию 60

5.3.3. Производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которого попадает работающий 61

5.3.4. Длительное сосредоточенное наблюдение 62

5.4. Экологическая безопасность 63

5.5. Безопасность в чрезвычайных ситуациях 63

Вывод по главе 65

Заключение 66

Список источников 69

ПРИЛОЖЕНИЕ А 70

ПРИЛОЖЕНИЕ Б 71

Введение

Данная работа выполнялась на обучающей площадке “Skillbox” в рамках курса “Python разработчик с нуля”. В этом курсе были изучены основные особенности языка Python и подходы к разработке программного кода. Завершающим заданием к аттестации по этому курсу было написание искусственного интеллекта для объектов двумерной игровой сцены. Задание было успешно выполнено и защищено перед экспертами. Было принято решение использовать эти результаты в данной выпускной квалификационной работе.

В разработке игр типа PvE большую роль в формировании игрового опыта играет искусственный интеллект (ИИ). В отличие от PvP игр, где игроки сражаются между собой и сами создают друг другу трудности в достижении цели, в играх PvE требуется качественно продумать поведение персонажей. Это необходимо для более глубокого погружения игрока в виртуальный мир. Создавая интересные и работающие модели поведения у игроков сохраняется игровой настрой и желание продолжить игру.

Создавая препятствия на пути к цели игрока важно сохранять баланс, между простой и сложностью. Если ИИ будет слишком сильным, а возможности компьютера в обработке информации значительно лучше человеческих, то у игрока могут появляться негативные эмоции, которые с большой вероятностью оттолкнут игрока от прохождения и испортят его опыт в игре. Если ИИ будет слабым, то игроку просто будет скучно и неинтересно проходить уровни, что тоже испортит впечатление об игре.

В данной работе не преследовалась цель создать ИИ, обладающий чувствами или способный обучаться, но велась разработка эффективного ИИ для дронов (игровых объектов), реализованных внутри библиотеки Python Astrobox, которые могут взаимодействовать с другими объектами на сцене.

Проблема: на данный момент разработчики Astrobox находятся в поиске эффективной стратегии сбора ресурсов и организации ведения огня, обеспечивающего защиту базы дронов, дронами.

Цель: создать жизнеспособную систему управления дронами в рамках библиотеки astrobox и сообщества разработчиков.

Задачи:

1. разработать искусственный интеллект для управления дронами
2. подтвердить его жизнеспособность и наметить планы для дальнейшего развития.

Глава 1. Исследование предметной области

1.1. Описание предметной области

В современном мире создание видеоигр является одним из наиболее крупных сегментов индустрии развлечений. Игровая индустрия по бюджету и охвату уже может с такими гигантами как киноиндустрия. А по скорости роста за последние пять лет рынок геймдева существенно опережал рынок кинематографа.

Высокая вовлеченность игрока в процесс является главным преимуществом игр, поэтому этот сегмент уже давно выделяется среди других видов развлечений.

Разработку игр невозможно рассматривать обособленно от индустрии компьютерных игр в целом. Непосредственно создание игр – это только часть комплексной «экосистемы», обеспечивающей полный жизненный цикл производства, распространения и потребления таких сложных продуктов, как компьютерные игры.

Разработка игр для разработчиков - это часть большого рынка создания игр.

Существует множество игр для разработчиков:

- Untrusted
- Robocode
- Elevator Saga
- Vim Adventures
- CodeCombat

Все эти игры имеют большой развивающий потенциал для разработчиков. Однако не все из них помогают развить способности к написанию рабочего кода на Python. Из представленного списка CodeCombat имеет такую возможность.

Однако все эти игры объединяет то, что у всех имеется графический интерфейс. И весь игровой процесс построен вокруг кода.

Сложно найти такой вариант развивающей игры, в которой бы мотивация играть была бы завязана не только в выборе правильного подбора строк кода.

Пример такой игры рассматривается в данной работе. Она поставляется и развертывается с помощью Astrobox - python библиотека с открытым исходным кодом.

1.2 Специфика Astrobox

Astrobox - python библиотека для развертывания окружения для обучения программированию разработчиков. Данная работа велась в версии Astrobox 1.6.0.

Astrobox позиционируется как игра для разработчиков. Предлагая свои варианты ИИ дронов, программисты начинают конкурировать с другими такими же разработчиками в эффективности ИИ. Разные люди, по-разному видят поведение дронов на сцене, кто-то ставит ставку на сбор ресурсов, кто-то больше продумывает тактики силового захвата ресурсов оппонента. А чей подход окажется более эффективным покажут еженедельные турниры, проводимые в компании “SkillBox”.

В этом подходе объединяются два типа игр PvP и PvE. Разработчики соревнуются между собой (PvP), проводят анализ своей работы, улучшают свои умения в разработке ИИ, но делают они это через свои запрограммированные объекты (EvE).

Игровой процесс представляет из себя сбор ресурсов космическими дронами в открытом космосе. У игрока есть определенное количество дронов , которые образуют команду. Каждой команде требуется собрать максимальное число ресурсов. Нужно разработать алгоритм, способный собирать больше ресурсов, чем представленные пользователями дроны.

Дроны могут, перемещаться в заданную точку, добывать ресурс, отгружать на свою базу, вести космический бой с противником. У дронов имеется определенный запас жизненных очков, участвуя в сражениях с

вражескими дронами и получая урон их значение уменьшается, но, когда дрон находится рядом со своей базой, это значение увеличивается

Все эти функции прописаны в абстрактном классе Drone. Интроспекция класса Drone приведена в приложении А.

Сцена представляет из себя замкнутое пространство заданной величины, в углах которой расположены базы противодействующих сторон. Между базами генерируются астероиды с особым космическим ресурсом “Элириумом” на каждом астероиде случайным образом генерируется его количество. У каждой базы имеется определенное количество жизненных очков. Уничтожение базы означает поражение команды.

Если происходит патовая ситуация, когда дроны не могут собирать ресурсы и вести боевые действия, действия на сцене прекращаются и подсчитываются результаты сбора “Элириума”

Все взаимодействие дронов на сцене прописано в классе SpaceField. Интроспекция класса SpaceField представлена в приложении А.

Используя данные классы требовалось настроить логику дрона на добычу максимально возможного числа ресурсов.

1.3. Описание используемого средства разработки

Так как Astrobox разработана на Python, то языком для реализации необходимых программных модулей стал Python. Python имеет в своем арсенале множество библиотек и готовых алгоритмов, предназначенных для анализа данных.

Для разработки на Python можно пользоваться простым блокнотом или любым другим редактором текста, однако использование IDE значительно упрощает разработку. Среди самых популярных IDE, поддерживающих Python можно выделить Jupiter, Visual Studio и PyCharm. Основываясь на личном опыте разработки, был выбран PyCharm, он по всем параметром подходил для работы.

PyCharm является интегрированной средой разработки для языка Python и имеет средства для анализа кода, графической отладки.

Разработка велась под операционной системой Windows, но PyCharm также работает с операционными системами Mac и Linux.

1.4. Описание разрабатываемых модулей

В рамках ВКР разрабатывался собственный модуль логики поведения дронов - класс DidenkoDrone. Данный класс разрабатывался на основе абстрактного класса Drone python библиотеки Astrobox.

Функции модуля:

- Отвечает за поведение каждого дрона в команде.
- Указывает как выполнять основные функции по сбору, доставке и защите ресурсов.
- Содержит методы, которые упрощают расчеты в выборе эффективной стратегии поведения.

На сцене каждый дрон под управлением разрабатываемого ИИ анализирует обстановку на сцене и принимает соответствующие решения.

Вывод по главе

В результате исследования предметной области стало ясно, что программа должна разрабатываться для Python библиотеки Astrobox, которая создана для проведения игрового обучения программированию всех желающих.

Основными задачами системы ИИ будет:

- анализ обстановки на сцене;
- выбор стратегии поведения;
- сбор, доставка и защита ресурсов.

Все эти задачи могут уместиться в рамках одного разрабатываемого модуля.

Для разработки программных модулей потребуется PyCharm, Python и Git.

Глава 2. Проектирование и реализация программных модулей

Весь процесс проектирования и реализации программных модулей можно разбить на 4 этапа:

- Изучение архитектуры библиотеки Astrobox
- Проектирование архитектуры класса DidenkoDrone
- Исследование поведения врагов
- Внедрение улучшений
- Тестирование улучшений

Изучение архитектуры библиотеки Astrobox заключалось в анализе кода классов Drone и SpaceField. Их взаимодействие, переменные и методы. При изучении были намечены основные подходы к разработке искусственного интеллекта дронов.

Последние 4 этапа производились итеративно. Для начала набрасывалась предварительная архитектура взаимодействия полей и методов класса, далее производилось исследование поведения врагов, затем на основе полученной информации внедрялись улучшения в программный код, после производилось симуляция поведения дронов и оценка качества внедренных улучшений.

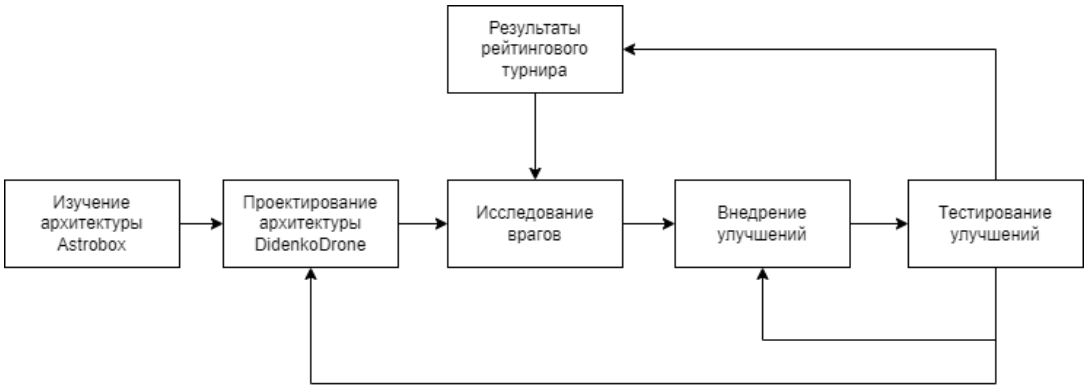

Рисунок 1 - диаграмма процесса проектирования и реализации модулей

2.1. Необходимый функционал

Для того чтобы дроны были жизнеспособны им необходимо обладать способностями к сбору, хранению и защите ресурсов. Основной сложностью при разработке являлось взаимодействие между несколькими дронами. Например для сбора ресурсов требуется грамотное распределение дронов по астероидом на которых хранится “Элириум”, так как противники собирают ресурсы с тех же астероидов, то требовалось определить самый эффективный подход к сбору ресурсов. При ведении огня по вражеским дронам и базам следовало предусмотреть эффективное расположение дронов на поле боя, таким образом чтобы они не мешали друг другу при стрельбе по противнику.

2.2. Описание процесса анализа

Существует определенная таблица рейтинга внутри разрабатываемой системы, которая показывает насколько представленное решение на текущий момент эффективно и жизнеспособно. Этот рейтинг высчитывается по формуле

$$\mathbb{E}_A = \frac{1}{1 + 10^{\frac{R_B - R_A}{400}}}, \tag{1}$$

где R_b и R_a - рейтинги игроков A и B.

Таким образом, чем сильнее противник, тем больше баллов получает разработчик за победу над ним.

От результата сражения будет зависеть коэффициент, с которым будут производиться дальнейшие расчёты:

- победа - 1x баллов,
- ничья - 0.5x баллов,
- поражение - 0x баллов.

Итоговая формула вычисления баллов выглядит так:

$$R'_A = R_A + K \cdot (S_A - \mathbb{E}_A),$$

(2)

Где

R'_a - новый рейтинг участника

R_a - старый рейтинг участника

K - koef_elo, коэффициент Эло

S_a - коэффициент результата сражения

E_a - коэффициент ожидания

Данная модель подсчета точно интерпретирует эффективность разрабатываемого ПО и ИИ в системе. И помогает постоянно отслеживать изменения в конкурентоспособности представляемого решения.

2.3. Данные, которыми оперирует ИИ

Классу Drone доступна информация о местоположении объектов сцены:

- своей базы
- вражеских дронов
- вражеских баз
- астероидов

Также дрон при выборе стратегии оперирует такими параметрами как:

- собственное местоположение
- количество очков жизни
- заполненность ресурсами

Вся информация собирается и анализируется дронами. Результаты анализа моментально влияют на принятие решений. Данные хранятся в оперативной памяти, сохраняются на диск только результаты рейтинговых боев ими разрабатываемый модуль не владеет, этот функционал встроен в библиотеку Astrobox.

2.4. Проектирование логики класса

Сам класс представляет из себя набор методов и полей.

Методы можно условно разделить на 3 группы:

- event_methods - методы срабатывающие на определенные события на сцене (входят в число абстрактных методов класса Drone)
- calc_methods - побочные вычислительные методы, помогающие произвести расчеты, помогающие при выборе стратегии
- on_heartbeat - метод срабатывающие каждый тик таймера

Поля можно условно разделить на 2 группы:

- Переменные состояния
- Переменные расчетов

Calc_methods, используя переменные расчетов, помогают event_methods изменять переменные состояния в зависимости от положения объектов на сцене. Метод on_heartbeat проверяет состояние переменных состояния и оперативно (в зависимости от частоты тика таймера) принимает стратегические решения по сбору и защите ресурсов команды.

Листинг кода приведен в приложении Б.

Вывод по главе

В данной главе были рассмотрены ключевые аспекты разработки ИИ дронов. Были обозначены требования к разрабатываемому модулю, критерии эффективности ИИ, основные данные которыми оперируют дроны, а также логика работы ИИ.

Глава 3. Демонстрация результатов разработки

Для демонстрации результатов работы требуется запустить симуляцию и загрузить с gitlab результаты работ других разработчиков. После работы в GitLab (создание веток, их слияния) нужно перейти в свою ветку, активировать виртуальное окружение выбрать противников, с которыми дроны будут состязаться и прописать команду запуска в терминале типа:

```
python -m battle -p folder1/module_1.py folder2/module_2.py –showscreen
```

После этого запустится симуляция и дроны начнут свою работу.

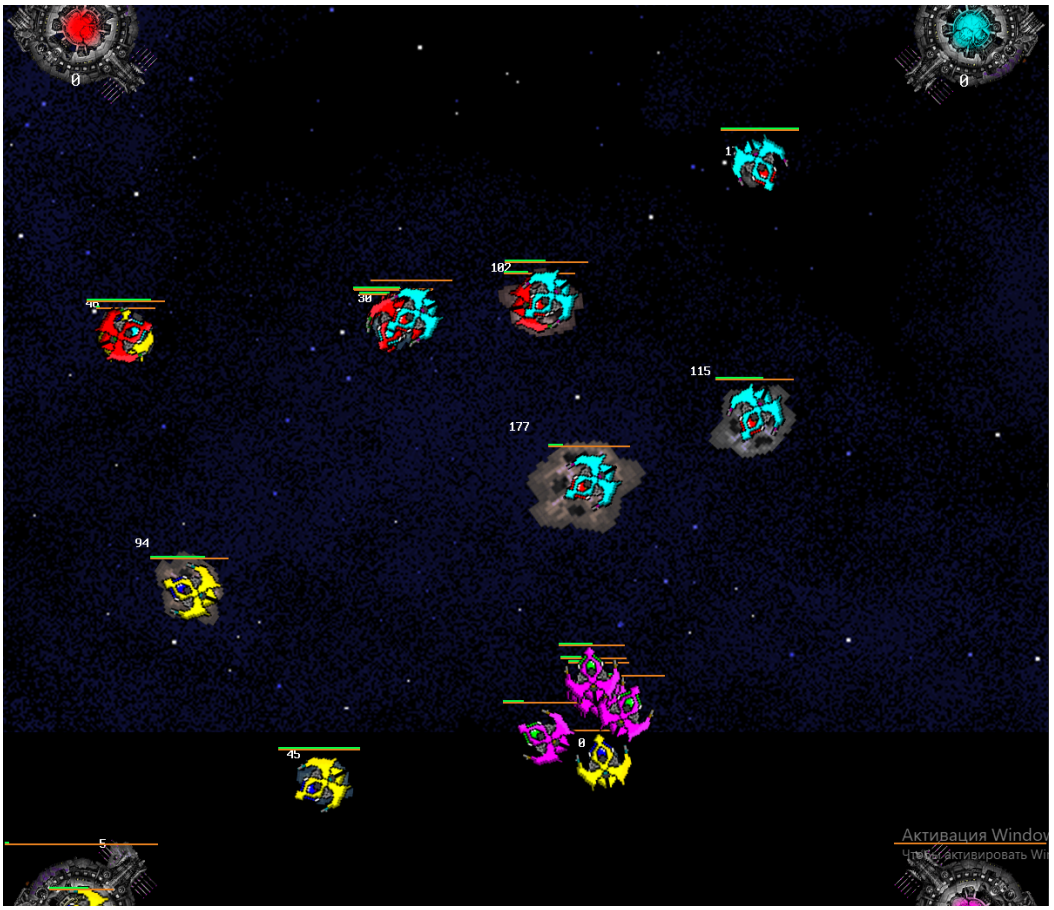

Рисунок 2 - Окно симуляции

Все элементы графического интерфейса встроены в Astrobox.

Основная стратегия дронов заключена в следующих принципах:

- На первом этапе выгоднее собирать ресурсы
- На втором этапе выгоднее защищать собранные ресурсы

Такой стратегией пользовалось большинство дронов разработчиков. Все упиралось в организацию оппозиции, когда все ресурсы собраны, а нападать не выгодно, потому что как только кто-то выйдет за пределы зоны лечения базы, то будет попадать под вражеский обстрел (а в бою участвует 4 команды). Основной строй при такой стратегии выглядит так как на рисунке N.

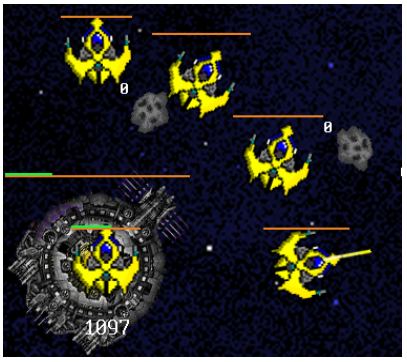

Рисунок 3 - Положение дронов при защите базы

Однако многократные пробы произведенные за время разработки показали, что дисциплинированный нападающий строй может обеспечить уничтожение базы соперника и добычу ресурсов выбывшего оппонента.

Поэтому при разработке логики переключения состояний было принято решение сконцентрироваться на атаке.

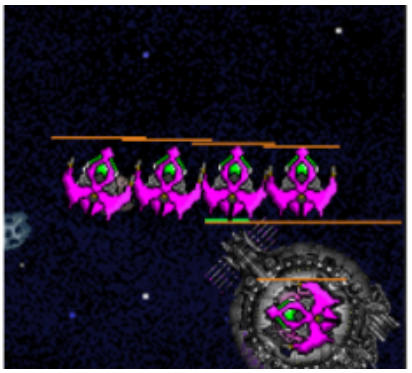

Рисунок 4 - Положение DidenkoDrone при атаке

Ведя концентрированный огонь по позициям соперника можно перестрелять восстановление жизненных очков у дронов и уничтожить их, а затем и саму базу команды.

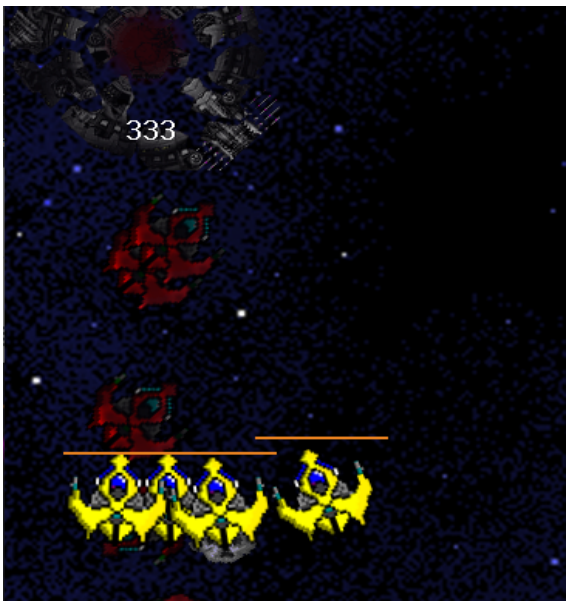

Рисунок 5 - Действия DidenkoDrone по сбору ресурсов с базы оппонента при его уничтожении

Данный подход и стал главной особенностью DidenkoDrone. К сожалению 100% вероятности победы достичь очень сложно. И проблемы озвученные ранее никуда не исчезли. Однако анализируя таблицы рейтинга можно убедиться, что данная модель поведения жизнеспособна и открывает новые возможности для разработки дронов.

Глава 4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

Разработка искусственного интеллекта (ИИ) производится группой работников, состоящей из двух человек – руководителя и студента.

Данная выпускная квалификационная работа заключается в разработке системы искусственного интеллекта для управления объектами двумерной игровой сцены.

Целью раздела «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» является определение перспективности и успешности ИИ, оценка его эффективности, разработка механизма управления и сопровождения конкретных проектных решений на этапе реализации.

Для достижения обозначенной цели необходимо решить следующие задачи:

1. Оценить коммерческий потенциал и перспективность разработки НИ;
2. Осуществить планирование этапов выполнения исследования;
3. Рассчитать бюджет затрат на исследования;

4.1 Потребители результатов исследования

Целевой аудиторией могут являться:

1. Образовательные платформы
2. Игровые компании
3. Логистические фирмы

Целевым рынком для данной разработки является рынок образования, предоставляющий услуги обучения студентов в игровом формате.

Исходя из вышеизложенного сегментацию рынка можно произвести по виду потребителей:

1) организации, предоставляющие образовательные услуги в классическом формате;

2) организации ищущие новые пути обучения студентов;
3) организации активно использующие элементы геймификации в образовании.

Таблица 1 - Карта сегментирования рынка

	вид организации			
размер компании		образование	геймдев	логистика
	крупные			
	средние			
	мелкие			

В качестве потребителя результатов проведенной разработки «Системы искусственного интеллекта для управления объектами двумерной игровой сцены» выступают разработчики python библиотеки “Astrobox”.

Компании занимающиеся образовательной деятельностью, использующие методы геймификации и искусственного интеллекта являются заинтересованными к данному исследованию. Это можно увидеть по их непосредственному участию в развитии системы, на которой происходила разработка.

Предприятием потребителем является компания ООО «Скилбокс». В этой компании развиты обучающие программы для разработчиков основанные в библиотеке python “Astrobox”.

4.2 Анализ конкурентных технических решений

Анализ конкурентных технических решений с позиции ресурсоэффективности и ресурсосбережения позволяет провести оценку сравнительной эффективности научной разработки и определить направления для ее будущего повышения.

Основными конкурентами являются разработчики, которые предоставляют свои решения для “Astrobox” в открытом виде. Разработчики представляют свои решения в большом количестве для проведения анализа было принято решение выбрать два самых востребованных решения на текущий момент (варианты искусственного интеллекта от разработчиков Пазухина и Захарова). Дроны под управлением искусственного интеллекта данных разработчиков на момент начала разработки являлись одними из самых эффективных и жизнеспособных.

Существует определенная таблица рейтинга внутри разрабатываемой системы, которая показывает насколько представленное решение на текущий момент эффективно и жизнеспособно. Этот рейтинг высчитывается по формуле

$$\mathbb{E}_A = \frac{1}{1 + 10^{\frac{R_B - R_A}{400}}},$$

где R_b и R_a - рейтинги игроков A и B.

Таким образом, чем сильнее противник, тем больше баллов получает разработчик за победу над ним.

От результата сражения будет зависеть коэффициент, с которым будут производиться дальнейшие расчёты:

- Победа - 1x баллов,
- ничья - 0.5x баллов,
- поражение - 0x баллов.

Итоговая формула вычисления баллов выглядит так:

$$R'_A = R_A + K \cdot (S_A - \mathbb{E}_A),$$

Где

R'_a - новый рейтинг участника

R_a - старый рейтинг участника

K - koef_elo, коэффициент Эло

S_a - коэффициент результата сражения

E_a - коэффициент ожидания

Данная модель подсчета точно интерпретирует эффективность разрабатываемого ПО и ИИ в системе. И помогает постоянно отслеживать изменения в конкурентоспособности представляемого решения.

На данный момент DidenkoDrone входит в топ 10 ИИ платформы.

Рейтинг по состоянию на 15.05.2022

Позиция	Имя команды	Рейтинг
1	VoychenkoDrones	1044
2	ZakharovDrone	925
3	ShapievDrone	901
4	KonovalovDrone	823
5	RochevDrone	799
6	ChashchinaDrone	794
7	DidenkoDrone	789
8	OlshannikovDron	785
9	PazukhinDrone	782
10	KodzaevDrone	779
11	UshkovDrone	772
12	IvanovDrone	769
13	KachanovDrone	756
14	EvteshinDrone	749
15	KazaryanDrone	749
16	BibikDrone	745
17	GoshkievDron	709
18	DevastatorDrone	693
19	RsadomskovDrone	667

Рисунок 6 - Рейтинг дронов среди разработчиков

Дополнительно можно представить оценочную карту, основные конкурентные технические решения представленные в ней это ИИ (искусственные интеллект) разработчиков (Пазухина и Захарова)

Таблица 2 - Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	Вес критерия	$ИИ_д$	$ИИ_з$	$ИИ_п$	$К_д$	$К_з$	$К_п$
1	2	3	4	5	6	7	8
Технические критерии оценки ресурсоэффективности							
1. Простота проведения	0,1	5	4	0,5	0,5	0,4	0,4
2. Стоимость услуги	0,2	5	4	1	1	0,8	0,6
3. Точность измерения	0,2	5	4	1	1	0,8	0,8
4. Универсальность метода	0,15	4	3	3	0,6	0,45	0,45
5. Безопасность метода	0,15	5	5	5	0,75	0,75	0,75
Экономические критерии оценки эффективности							
6. Цена	0,1	5	4	3	0,5	0,4	0,3
7. Конкурентоспособность	0,1	4	3	3	0,4	0,3	0,3
Итого	**1**	**33**	**27**	**25**	**4,75**	**3,9**	**3,6**

4.3 SWOT-анализ

Произведем также в данном разделе SWOT – анализ ИИ, позволяющий оценить факторы и явления, способствующие или препятствующие продвижению проекта на рынок.

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

Слабые стороны– это недостаток, упущение или ограниченность научно-исследовательского проекта, которые препятствуют достижению его

целей. Это то, что плохо получается в рамках проекта или где он располагает недостаточными возможностями или ресурсами по сравнению с конкурентами.

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

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

На первом этапе SWOT анализа в таблице 4.2 были описаны сильные и слабые стороны проекта, выявлены возможности и угрозы реализации НИ.

Таблица 3 – Матрица SWOT анализа

Сильные стороны	Возможности во внешней среде
С1. Простота разработки С2. Быстрая интеграция в систему С3. Широкий пул возможностей дронов С4. Наличие опытного руководителя.	В1. Развитие до уровня логистической системы В2. Большой потенциал применения ИИ в России и других странах;
Слабые стороны	Угрозы внешней среды
Сл1. Тяжелая масштабируемость кода Сл2. Большой уровень конкуренции Сл3. Значительные временные и интеллектуальные затраты	У1. Отказ от технической поддержки проекта после внедрения У2. Нехватка финансирования

Второй этап состоит в выявлении соответствия сильных и слабых сторон научно-исследовательского проекта внешним условиям окружающей среды. Это соответствие или несоответствие должны помочь выявить степень необходимости проведения стратегических изменений. В рамках данного этапа необходимо построить интерактивную матрицу проекта. Ее использование помогает разобраться с различными комбинациями взаимосвязей областей матрицы SWOT. Возможно использование этой матрицы в качестве одной из основ для оценки вариантов стратегического выбора. Каждый фактор помечается либо знаком «+» (означает сильное соответствие сильных сторон возможностям), либо знаком «-» (что означает слабое соответствие); «0» – если есть сомнения в том, что поставить «+» или «-». Интерактивная матрица проекта представлена в табл. 4.3.

Таблица 4 - Интерактивная матрица сильных и слабых сторон и возможностей

	Сильные стороны					Слабые стороны		
Возможности проекта		С1	С2	С3	С4	Сл1	Сл2	Сл3
	В1	0	0	+	+	-	0	0
	В2	+	+	+	+	+	+	+

Таблица 5 - Интерактивная матрица сильных сторон и слабых сторон и угроз

	Сильные стороны					Слабые стороны		
Угрозы проекта		С1	С2	С3	С4	Сл1	Сл2	Сл3
	У1	+	+	+	-	-	+	-
	У2	+	+	+	-	-	-	-

Самой большой угрозой для проекта является отсутствие финансовой поддержки из-за дороговизны и сложности разработки.

Что касаемо слабых сторон, то для данных методов требуется привлечение опытных и квалифицированных специалистов, обеспечение обучения нового персонала со знаниями методов, используемых в искусственном интеллекте.

В рамках третьего этапа составляется итоговая матрица SWOT-анализа, представленная в таблице 4.5.

Таблица 6 - Итоговая матрица SWOT-анализа

	Сильные стороны научно-исследовательского проекта: С1. Простота разработки С2. Быстрая интеграция в систему С3. Широкий пул возможностей дронов С4. Наличие опытного руководителя.	Слабые стороны научно-исследовательского проекта: Сл1. Тяжелая масштабируемость кода Сл2. Большой уровень конкуренции Сл3. Значительные временные и интеллектуальные затраты
Возможности: В1. Развитие до уровня логистической системы В2. Большой потенциал применения ИИ в России и других странах;	Большой потенциал применения метода в России и других странах способствует развитию и доработке искусственного интеллекта объектов двумерной игровой сцены	Для данного метода требуется привлечение опытных и квалифицированных специалистов, обеспечить обучение нового персонала со знаниями программирования на языке python
Угрозы: У1. Отказ от технической поддержки проекта после внедрения У2. Нехватка финансирования	Простота разработки и быстрая интеграция - это ключевые показатели, которые привлекают возможные источники финансирования	Тяжелая масштабируемость может негативно повлиять на жизненный цикл продукта сократив его.

4.4 Планирование работ по научно-техническому исследованию

4.4.1 Структура работ в рамках научного исследования

Планирование комплекса предполагаемых работ осуществляется в следующем порядке:

- определение структуры работ в рамках научного исследования;
- определение участников каждой работы;
- установление продолжительности работ;
- построение графика проведения научных исследований.

Для выполнения научных исследований формируется рабочая группа, в состав которой могут входить научные сотрудники и преподаватели, инженеры, техники и лаборанты, численность групп может варьироваться. По каждому виду запланированных работ устанавливается соответствующая должность исполнителей.

Перечень этапов и работ, распределение исполнителей по данным видам работ приведен в таблице 4.6.

Таблица 7 – Перечень этапов, работ и распределение исполнителей

Основные этапы	№ раб	Содержание работ	Должность исполнителя
Разработка технического задания	1	Составление и утверждение технического задания	Руководитель
Выбор направления исследований	2	Выбор направления исследований	Бакалавр
	3	Подбор и изучение материалов по теме	Бакалавр
	4	Календарное планирование работ	Руководитель Бакалавр
Теоретические и экспериментальные исследования	5	Создание архитектуры системы	Руководитель Бакалавр
	6	Разработка основной логики	Бакалавр
Обобщение и оценка результатов	7	Оценка эффективности полученных результатов	Бакалавр
Оформление отчета по НИР	8	Составление пояснительной записки	Бакалавр

4.4.2 Определение трудоемкости выполнения работ

Трудовые затраты в большинстве случаев образуют основную часть стоимости разработки, поэтому важным моментом является определение трудоемкости работ каждого из участников научного исследования.

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

$$t_{ожi} = \frac{3t_i + 2t_i}{5}, \quad (4.2)$$

где $t_{ожi}$ – ожидаемая трудоемкость выполнения i-ой работы чел.-дн.;

t_i – минимально возможная трудоемкость выполнения заданной i-ой работы, чел.-дн.;

t_i – максимально возможная трудоемкость выполнения заданной i-ой работы, чел.-дн.;

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

$$T_{pi} = \frac{t_{ожi}}{Ч_i}, \quad (4.3)$$

где T_{pi} – продолжительность одной работы, раб.дн.;

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

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

4.4.3 Разработка графика проведения научного исследования

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

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

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

$$T_{ki} = T_{pi} \cdot k_{\text{кал}}, \quad (4.4)$$

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

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

$k_{\text{кал}}$ – коэффициент календарности.

Коэффициент календарности определяется по следующей формуле:

$$k_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - (T_{\text{вых}} + T_{\text{пр}})}, \quad (4.5)$$

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

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

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

Расчет коэффициента календарности:

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

Таблица 8 – Временные показатели проведения научного исследования

Название работы	Трудоёмкость работ									Исполнители	Длительность работ в рабочих днях T_{pi}			Длительность работ в календарных днях T_{ki}		
	T_{min}, чел–дни			T_{max}, чел–дни			$T_{ожi}$, чел– дни									
	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3		Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
Выбор темы ВКР	1	1	1	2	2	2	1,4	1,4	1,4	Студент, научный руководитель	1	1	1	1	1	1
Составление и утверждение плана работ	1	1	1	2	2	2	1,4	1,4	1,4	Научный руководитель	1	1	2	1	1	2
Подбор и изучение материалов по теме	2	2	3	4	4	5	2,8	2,8	3,8	Студент	2	2	3	2	2	4
Выбор направления исследования	1	1	1	3	2	2	1,8	1,4	1,4	Студент, научный руководитель	2	2	2	2	2	2
Календарное планирование работ	1	1	1	3	4	4	1,8	2,2	2,2	Студент, научный руководитель	2	3	3	2	3	3
Подбор и изучение материалов по теме	9	10	13	12	16	17	10,2	12,4	14,6	Студент	12	13	16	16	17	22
Создание архитектуры системы	10	12	14	12	15	17	10,8	13,2	15,8	Студент, научный руководитель	11	14	15	13	18	19

Разработка методики	5	6	7	8	9	10	6,2	7,2	8,2	Студент	7	7	9	9	9	11
Оценка эффективности полученных результатов	4	4	5	6	6	8	4,8	4,8	6,2	Студент	4	5	5	4	7	7
Написание раздела «Финансовый менеджмент»	4	4	4	6	6	6	4,8	4,8	4,8	Студент	5	5	5	5	5	5
Написание раздела «Социальная ответственность»	1	2	2	3	4	4	1,8	2,8	2,8	Студент	1	2	2	1	2	2
Оформление ВКР	5	4	6	7	7	8	5,8	5,2	6,2	Студент	5	6	7	5	8	9

Составлен план научного исследования, в котором разработан календарный план выполнения работ. Для построения таблицы временных показателей проведения НИ был рассчитан коэффициент календарности. С помощью показателей в табл. 4.8 был разработан календарный план-график проведения НИ по теме. Для иллюстрации календарного плана была использована диаграмма Ганта, указывающая на целесообразность проведения данного исследования.

Таблица 9 - Календарный план-график проведения научного исследования

№ ра-бот	Вид работ	Испол-нители	T_{ki}, кал. дн.	Продолжительность выполнения работ										
				февраль		март			апрель			май		
				2	3	1	2	3	1	2	3	1	2	3
1	Выбор темы ВКР	Ст, НР	1											
2	Составление и утверждение плана работ	НР	1											
3	Подбор и изучение материалов по теме	Ст	2											
4	Выбор направления исследования	Ст, НР	2											
5	Календарное планирование работ	Ст, НР	2											
6	Подбор и изучение материалов по теме	Ст	16											

7	Создание архитектуры системы	Ст, НР	13												
8	Разработка методики	Ст	9												
9	Оценка эффективности полученных результатов	Ст	4												
10	Написание раздела «Финансовый менеджмент»	Ст	5												
11	Написание раздела «Социальная ответственность»	Ст	1												
12	Оформление ВКР	Ст	5												

– научный руководитель; – студент.

4.5 Бюджет научно-технического исследования (НТИ)

Для подробного планирования бюджета НТИ, необходимо отразить все расходы, связанные с его выполнением. Далее каждая из статей расходов будет рассмотрена подробно.

4.5.1 Расчет материальных затрат НТИ

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

В материальных затратах были учтены канцелярский принадлежности и расходы на электроэнергию, другие ресурсы на начало разработки у исполнителей имелись.

Тетрадь и ручка использовались для оперативного создания заметок, рисунков, вычислений. Бумага А4 и чернила для принтера использовались для печати документов. Электроэнергия использовалась для функционирования техники (персонального компьютера).

Расчет материальных затрат осуществляется по формуле:

$$З_{М} = \sum_{i=1}^{m} Ц_{i} \cdot N_{расхi}, \quad (4.6)$$

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

$N_{расхi}$ – количество материальных ресурсов i-го вида, планируемых к использованию при выполнении научного исследования;

$Ц_{i}$ – цена приобретения единицы i-го вида потребляемых материальных ресурсов.

Таблица 10 – Материальные затраты

Наименование	Единица измерения	Количество	Цена за ед., руб.	Затраты на материалы, ($З_{М}$), руб.
Тетрадь для записей	Шт.	1	45	45
Ручка	Шт.	1	15	15
Бумага А4	Уп	1	300	300
Черная краска для принтера	Уп	1	1700	1700
Электроэнергия	кВт*ч	250	3,5	875
Итого, руб.				2935

Общие материальные затраты составили 2935 руб.

4.5.2 Расчет стоимости спецоборудования и амортизационных затрат

Затраты на специальное оборудование приведены в таблице 4.10. В специальное оборудование входят оборудование для рабочего места и ПО, необходимое для реализации искусственного интеллекта.

Таблица 11 – Расчет стоимости спецоборудования для научных работ

Наименование	Единица измерения	Количество	Цена за ед., тыс. руб.	Затраты на материалы, тыс. руб.
Персональный компьютер	Шт.	1	40	40
Программное обеспечение (PyCharm)	Шт.	1	15	15

Комплект периферийных устройств	Шт.	1	6	6
Монитор	Шт.	2	15	30
Итого:				91

Амортизационные отчисления для рассматриваемого проекта включают в себя амортизацию используемого оборудования за время выполнения работы. Амортизационные отчисления рассчитываются по времени использования компьютера по формуле:

$$С_{ам} = Н_а * Ц_{об} / F_д * t_{рф} * n \quad (7)$$

где $Н_а$ – годовая норма амортизации;

$Ц_{об}$ – цена оборудования;

$F_д$ – действительный годовой фонд рабочего времени;

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

n – число задействованных единиц оборудования, n = 1.

Амортизация будет рассчитываться из списка спецоборудования, исключая программное обеспечение. Годовая амортизация НА определяется как величина, обратная сроку амортизации оборудования СА, который определяется согласно постановлению правительства РФ «О классификации основных средств, включенных в амортизационные группы». Для компьютера и периферийного оборудования, использующегося с ним, примем СА = 3 года, тогда НА = 0,33.

Расчет затрат на амортизационные отчисления представлен в таблице 4.10.2.

Таблица 12 - Расчет затрат на амортизационные отчисления

Наименование	Норма аморт. Оборуд., $Н_а$	Стоим. Оборуд., $Ц_{об}$, руб.	Факт. р/вр. Оборуд., $t_{рф}$, ч	Действ. Год. Фонд р/вр., $F_д$, ч.	Аморт. Отчисл., $С_{ам}$, руб.
Персональный компьютер	0,33	40000	240	1720	1841,86
Комплект периферийных	0,33	8000	240	1720	368,37

устройств					
Мониторы	0,33	30000	240	1720	1381,40
ИТОГО					3591,63

4.5.3 Основная заработная плата исполнителя темы

В настоящую статью включается основная заработная плата научных и инженерно-технических работников, рабочих макетных мастерских и опытных производств, непосредственно участвующих в выполнении работ по данной теме. Величина расходов по заработной плате определяется исходя из трудоемкости выполняемых работ и действующей системы окладов и тарифных ставок. В состав основной заработной платы включается премия, выплачиваемая ежемесячно из фонда заработной платы в размере 20-30 % от тарифа или оклада. Расчет основной заработной платы приводится в таблице 4.11.

Таблица 13 – Расчет основной заработной платы

№ п/п	Наименование этапов	Исполнители по категориям	Трудоемкость, чел.-дн.	Заработная плата, приходящаяся на один чел-дн.	Всего заработная плата по тарифу (окладам), тыс. руб.
1.	Выбор темы ВКР	Ст, НР	1	5,1	5,1
2.	Составление и утверждение плана работ	НР	1	3,1	3,1
3.	Подбор и изучение материалов по теме	Ст	2	2	4
4.	Выбор направления исследования	Ст, НР	2	5,1	10,2
5.	Календарное планирование работ	Ст, НР	2	5,1	10,2
6.	Подбор и изучение материалов по теме	Ст	12	2	24
7.	Создание архитектуры системы	Ст, НР	11	5,1	56,1

8.	Разработка методики	Ст	7	2	14
9.	Оценка эффективности полученных результатов	Ст	4	2	8
10.	Написание раздела «Финансовый менеджмент»	Ст	5	2	10
11	Написание раздела «Социальная ответственность»	Ст	1	2	2
12	Оформление ВКР	Ст	5	2	10
Итого					156,7

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

$$З_{зп} = З_{осн} + З_{доп} \quad (4.7)$$

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

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

Основная заработная плата руководителя рассчитывается по следующей формуле:

$$З_{осн} = З_{дн} \cdot Т_{р} \quad (4.8)$$

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

$Т_{р}$ – продолжительность работ, выполняемых научно-техническим работником, раб. дн.;

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

Среднедневная заработная плата рассчитывается по формуле:

$$З_{дн} = \frac{З_{м} \cdot М}{F_{д}} \quad (4.9)$$

где $З_{м}$ – месячный должностной оклад работника, руб.;

М – количество месяцев работы без отпуска в течение года:

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

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

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

Таблица 14 – Баланс рабочего времени

Показатели рабочего времени	Руководитель	Студент
Календарное число дней	365	365
Количество нерабочих дней - выходные дни - праздничные дни	118	118
Потери рабочего времени - отпуск - невыходы по болезни	 48 0	 72 0
Действительный годовой фонд рабочего времени	199	175

Месячный должностной оклад работника (руководителя):

$$З_м = З_{тс} \cdot (1 + k_{пр} + k_д) \cdot k_р \quad (4.10)$$

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

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

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

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

Тарифная заработная плата $З_{тс}$ находится из произведения тарифной ставки работника 1-го разряда Тсi = 600 руб. на тарифный коэффициент $k_т$ и учитывается по единой для бюджетной организации тарифной сетке.

Тарифный коэффициент для НР = 1,866; для С = 1,407.

Расчет основной заработной платы представлен в таблице 4.13

Таблица 15 – Расчет основной заработной платы

Исполнители	Разряд	$k_Т$	$З_{тс}$, руб.	$k_{пр}$	$k_д$	$k_р$	$З_м$, руб.	$З_{дн}$, руб.	$Т_р$, раб. дн.	$З_{осн}$, руб.
Научный руководитель	Кандидат тех. наук	1,866	30000	0,3	0,4	1,3	66300	3731,45	17	38505
Студент	Инженер	1,407	15000	0,3	0,2	1,3	29250	1872	52	50700

Итого	89205

4.5.4 Расчет дополнительной заработной платы

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

Расчет дополнительной заработной платы рассчитывается по формуле:

$$З_{доп} = k_{доп} \cdot З_{осн}, \quad (4.11)$$

где $k_{доп}$ – коэффициент дополнительной заработной платы, принятый на стадии проектирования за 0,15.

Соответственно:

Научный руководитель: 38505*0,15=5775,75

Студент: 50700*0,15=7605

4.5.5 Отчисления во внебюджетные фонды

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

Величина отчислений во внебюджетные фонды определяется исходя из формулы:

$$З_{внеб} = k_{внеб} \cdot (З_{осн} + З_{доп}) \quad (4.12)$$

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

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

Отчисления во внебюджетные фонды представлены в таблице 4.14.

Таблица 16 – Отчисления во внебюджетные фонды

Исполнитель	Основная заработная плата, руб.	Дополнительная заработная плата, руб.
Руководитель проекта	38505	5775,75
Студент	50700	7605
Коэффициент отчислений во внебюджетные фонды	0,302	
ИТОГО: 30980,90		

4.5.6 Накладные расходы

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

$$З_{накл} = \left(\sum статей\right) \cdot k_{нр} \qquad (4.13)$$

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

Величину коэффициента накладных расходов можно взять в размере 16%.

Накладные расходы составили:

$$З_{накл} = (2935 + 91000 + 3591,63 + 89205 + 13380,75 + 30980,90) \cdot 0,16 = 36\,974,92 \text{ руб.}$$

При разработке программного обеспечения накладные расходы были направлены в создание нормальных условий труда к этому относится:

1. Питание разработчиков;
2. Организация совместных мероприятий;

3. Прибытие к месту работы (убытие);
4. Посещение тематический встреч, митапов, конференций.

4.5.7 Формирование бюджета затрат научно-исследовательского проекта

Рассчитанная величина затрат научно–исследовательской работы является основой для формирования бюджета затрат проекта. Определение бюджета затрат на научно–исследовательский проект приведено в таблице 4.15.

Таблица 17 –Расчет бюджета затрат НТИ

Наименование статьи	Сумма, руб.	Примечание
Материальные затраты НТИ	2935	Пункт 4.5.1
Затраты на специальное оборудование для научных (экспериментальных) работ	91000	Пункт 4.5.2
Амортизационные затраты	3591,63	Пункт 4.5.2
Затраты по основной заработной плате исполнителей темы	89205	Пункт 4.5.3
Затраты по дополнительной заработной плате исполнителей темы	13380,75	Пункт 4.5.4
Отчисления во внебюджетные фонды	30980,90	Пункт 4.5.5
Накладные расходы	36 974,92	Пункт 4.5.6
9. Бюджет затрат НТИ	266068,20	

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

Определение эффективности происходит на основе расчета интегрального показателя эффективности научного исследования. Его нахождение связано с определением двух средневзвешенных величин: финансовой эффективности и ресурсоэффективности.

Интегральный показатель финансовой эффективности научного исследования определяется как:

$$I_{\text{фин.р}}^{\text{исп.}i} = \frac{\Phi_{pi}}{\Phi_{max}} \qquad (4.14)$$

где $I_{\text{фин.р}}^{\text{исп.}i}$ – интегральный финансовый показатель разработки;

$Ф_{pi}$ – стоимость i-го варианта исполнения;

$Ф_{max}$ – максимальная стоимость выполнения научно-исследовательского проекта.

$$I_{\text{фин.р}}^{\text{исп1}} = \frac{360393,75}{458685,30} = 0,785;$$

$$I_{\text{фин.р}}^{\text{исп2}} = \frac{414270,20}{458685,30} = 0,903;$$

$$I_{\text{фин.р}}^{\text{исп3}} = \frac{458685,30}{458685,30} = 1$$

Интегральный показатель ресурсоэффективности вариантов исполнения объекта исследования можно определить следующим образом:

$$I_{pi} = \sum_{i=1}^{n} a_i \times b_i \qquad (4.15)$$

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

a_i – весовой коэффициент i-го варианта исполнения разработки;

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

n – число параметров сравнения.

Таблица 18– Сравнительная оценка характеристик вариантов исполнения

Объект исследования Критерии	Весовой коэффициент параметра	ИИд	ИИп	ИИз
1. Потребность в ресурсах памяти	0,1	5	3	4
2. Функциональность	0,2	5	4	3
3. Простота эксплуатации	0,15	3	4	3
4. Скорость работы	0,15	5	3	4
5. Надежность	0,2	5	4	4
6. Точность	0,2	5	3	3
Итого	1	4,7	3,55	3,45

$$I_{\text{р}-\text{иид}} = 0,1 \cdot 5 + 0,2 \cdot 5 + 0,15 \cdot 3 + 0,15 \cdot 5 + 0,2 \cdot 5 + 0,2 \cdot 5 = 4,7;$$

$$I_{\text{р}-\text{иип}} = 0,1 \cdot 3 + 0,2 \cdot 4 + 0,15 \cdot 4 + 0,15 \cdot 3 + 0,2 \cdot 4 + 0,2 \cdot 3 = 3,55;$$

$$I_{\text{р}-\text{ииз}} = 0,1 \cdot 4 + 0,2 \cdot 3 + 0,15 \cdot 3 + 0,15 \cdot 4 + 0,2 \cdot 4 + 0,2 \cdot 3 = 3,45.$$

Интегральный показатель эффективности вариантов исполнения разработки ($I_{\text{испi}}$) определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формуле:

$$I_{\text{иид}} = \frac{I_{\text{р}-\text{иид}}}{I_{\text{фин.р}}^{\text{иид}}} = \frac{4,7}{0,785} = 5,98$$

$$I_{\text{иип}} = \frac{I_{\text{р}-\text{иип}}}{I_{\text{фин.р}}^{\text{иип}}} = \frac{3,55}{0,903} = 3,93;$$

$$I_{\text{ииз}} = \frac{I_{\text{р}-\text{ииз}}}{I_{\text{фин.р}}^{\text{ииз}}} = \frac{3,45}{1} = 3,45.$$

Сравнение интегрального показателя эффективности вариантов исполнения разработки позволит определить сравнительную эффективность проекта и выбрать наиболее целесообразный вариант из предложенных.

Сравнительная эффективность проекта ($Э_{ср}$):

$$Э_{\text{ср}} = \frac{I_{\text{исп2}}}{I_{\text{исп1}}} \qquad (4.16)$$

Таблица 19 – Сравнительная эффективность разработки

№	Показатели	ИИд	ИИп	ИИз
1	Интегральный финансовый показатель разработки	0,785	0,903	1
2	Интегральный показатель ресурсоэффективности разработки	4,85	3,55	3,45
3	Интегральный показатель эффективности	5,98	3,93	3,45
4	Сравнительная эффективность вариантов исполнения	1	0,657	0,576

Сравнив значения интегральных показателей эффективности можно сделать вывод, что реализация технологии в первом исполнении является более

эффективным вариантом решения задачи, поставленной в данной работе с позиции финансовой и ресурсной эффективности.

Вывод по разделу

В результате проделанной по разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» работы, был исследован проект, выполняемый в рамках научно-исследовательской работы, определены слабые и сильные стороны проекта, его конкуренты, потребители и эффективность, был построен календарный план-график для своевременного решения поставленных задач. Для проекта был подсчитан бюджет затрат, он составил 265 748,20 руб.

По результатам исследования можно сделать вывод, вариант искусственного интеллекта является жизнеспособным, он уверенно может конкурировать с решениями других разработчиков. У проектируемого ИИ есть ключевые конкурентоспособные особенности - это простота разработки и быстрая интеграция в систему. По данным параметрам решении опережает конкурентов и может обеспечить себе достаточную долю рынка.

Глава 5. Социальная ответственность

Введение

Разработка искусственного интеллекта (ИИ) производится группой работников, состоящей из двух человек – руководителя и студента. Выпускная квалификационная работа заключается в разработке системы искусственного интеллекта для управления объектами двумерной игровой сцены.

В качестве потребителя результатов проведенной разработки выступают разработчики python библиотеки “Astrobox”. Компании занимающиеся образовательной деятельностью, использующие методы геймификации и искусственного интеллекта являются заинтересованными к данному исследованию. Это можно увидеть по их непосредственному участию в развитии системы, на которой происходила разработка.

Разработка искусственного интеллекта велась в Томске.

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

Выпускная квалификационная работа по разработке искусственного интеллекта для объектов на двумерной игровой сцене выполнялась в ходе преддипломной практики в Кибернетическом центре. Проектируемое рабочее место представляет собой компьютерный класс, в котором будет работать разработчик.

Характеристика помещения:

- ширина – 5,5 м, длина – 6 м, высота – 3,0 м;
- площадь – 33 $м^2$;
- объем – 99 $м^3$;
- в помещении установлен кондиционер, имеется естественная вентиляция – вытяжное вентиляционное отверстие, дверь, окно, щели;

- в помещении установлено искусственное освещение, имеется естественное освещение.

В данном помещении максимальное количество сотрудников в одну смену – 4. В среднем на одного сотрудника приходится 8,25 $м^2$ площади и около 24,75 $м^3$ объема помещения. Данное размещение сотрудников удовлетворяет санитарным нормам, согласно которым на одного работника должно приходиться не менее 6 $м^2$ площади и 24 $м^3$.

5.1. Правовые и организационные вопросы обеспечения безопасности

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

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

В соответствии со статьей 162 ТК РФ «Введение, замена и пересмотр норм труда» о введении новых норм труда работники должны быть извещены не позднее чем за два месяца.

Согласно статье 163 ТК РФ «Обеспечение нормальных условий работы для выполнения норм выработки», работодатель обязан обеспечить:

• исправное состояние помещений и оборудования;

• надлежащее качество материалов и инструментов, необходимых для выполнения работы, их своевременное предоставление работнику;

• условия труда, соответствующие требованиям охраны труда и безопасности производства.

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

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

• создание и функционирование системы управления охраной труда;

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

• соответствующее требованиям охраны труда рабочее место;

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

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

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

Организация рабочего места при выполнении работы должна производиться в соответствии с требованиями ГОСТ 12.2.032-78 «Система стандартов безопасности труда. Рабочее место при выполнении работ сидя» и соблюдением трудовых норм, регулирующихся Трудовым кодексом РФ.

С учетом требований ГОСТ 12.2.032-78:

• конструкция рабочего места и взаимное расположение всех его элементов (сиденье, органы управления, средства отображения информации и т.д.) должны соответствовать физиологическим и психологическим требованиям, а также характеру работы.

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

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

• конструкцией оборудования и рабочего места должно быть обеспечено оптимальное положение работающего, которое достигается регулированием: высоты рабочей поверхности, сиденья, пространства для ног.

• при работе двумя руками органы управления размещают с таким расчетом, чтобы не было перекрещивания рук.

• очень часто используемые средства отображения информации, требующие точного и быстрого считывания показаний, следует располагать в вертикальной плоскости под углом ±15° от нормальной линии взгляда и в горизонтальной плоскости под углом ±15° от сагиттальной плоскости.

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

5.2. Производственная безопасность

В процессе работы химические и биологические факторы не оказывают влияния на состояние здоровья, поэтому рассмотрим только физические и психофизиологические факторы.

К вредным и опасным производственным факторам можно отнести:

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

Таблица 20 - Возможные опасные и вредные факторы производственные факторы на рабочем месте за персональным компьютером

Факторы (ГОСТ 12.0.003-2015)	Нормативные документы
1. Производственные факторы, связанные с электромагнитными излучениями;	СанПиН 1.2.3685-21

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

Более подробно каждый из этих факторов оценки были рассмотрены далее.

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

5.3.1. Производственные факторы, связанные с электромагнитными излучениями

Персональный компьютер имеет свойство подвергать работника вредному электромагнитному излучению. Но во время работы без него не обойтись. Электромагнитное излучение компьютера изменяется в диапазоне частот от 0 Гц до 1000 МГц. Такое излучение состоит из электрической и магнитной составляющих.

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

$$T = (50/E) – 2.$$

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

5.2.1.2. Отсутствие или недостаток необходимого искусственного освещения;

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

В СП 52.13330.2016 зрительная работа сотрудника, работающего с ПК охарактеризована как работу разряда Б – высокой точности (наименьший эквивалентный размер объекта различения - 0,3-0,5 мм), подразряда 1 (относительная продолжительность зрительной работы при направлении зрения на рабочую поверхность не менее 70%). В таблице 18 представлены требования к освещению рабочего помещения для вышеуказанного разряда.

Таблица 21 — Требования к освещению рабочего помещения для разряда Б1

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

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

5.2.1.3. Повышенный уровень шума;

Повышенный уровень шума на рабочем месте обусловлен использованием персональных компьютеров, наличием центральной системы вентиляции и кондиционирования воздуха. Становится сложнее разбирать речь, работоспособность снижается и повышается утомляемость сотрудников.

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

Таблица 22 - Предельно допустимые уровни звукового давления, уровни звука и эквивалентные уровни звука

Вид трудовой деятельности, рабочее место	Уровни звукового давления, дБ в октавных полосах со среднегеометрическими частотами, Гц									Уровни звука и эквивалентные уровни звука, дБА
	31,5	63	125	250	500	1000	2000	4000	8000	
Конструирование и проектирование, программирование. Рабочие места в помещениях дирекции, проектно-конструкторских бюро, расчетчиков, программистов вычислительных машин, в лабораториях для теоретических работ и обработки данных.	86	71	61	54	49	45	42	40	39	54

Существуют следующие пути уменьшения воздействий шума: экранирование рабочих мест (установка перегородок между рабочими местами); установка менее шумного оборудования; чистка оборудования от пыли, замена смазывающих веществ, т.к. любое оборудование при загрязнении увеличивает уровень шума.

5.3.2. Монотонность труда, вызывающая монотонию

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

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

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

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

Для снижения уровня монотонности можно проводить следующие мероприятия:

- внедрение режима труда и отдыха;
- чередование операций и темпа их выполнения;
- частые, но кратковременные перерывы во время работы;
- выполнять физических упражнений в течение перерывов;
- менять положение выполнения работ (стоя – сидя);

- посещение специальных помещений психологической разгрузки и отдыха.

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

Большое количество электрических приборов и вычислительных машин на рабочем месте объясняет важность электробезопасности на производстве.

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

Накопление статического электричества может привести к проблемам с вычислительными машинами. Однослойный линолеум для покрытия пола снизит величины зарядов статического электричества.

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

При повышенная влажность (относительная влажность воздуха выше 75 %) возрастает риск поражения электрическим током. Высокая температура воздуха и поверхностей (более 35 °С) тоже повышает вероятность распространения электрического тока. Даже наличие токопроводящей пыли и токопроводящих полов повысит риск поражения электрическим током.

Даже если работа будет проводиться исключительно в помещениях, исключающих повышенную опасность, есть риск возникновения следующих опасностей:

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

- при коротком замыкании в высоковольтных блоках.

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

Для предотвращения возникновения опасных ситуаций обязательны следующие меры предосторожности:

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

5.3.4. Длительное сосредоточенное наблюдение

Умственный труд разработчика заключается в приеме информации, ее переработке и выработке нестандартного решения. При такой работе большое значение имеет процесс мышления, выбор оптимального решения из ряда логических вариантов. Поиск решения связан с опытом, знаниями, особенностями нервной системы человека.

От напряженного умственного труда страдают зрительные и слуховые анализаторы, центральная нервная система, в особенности высшие психические функции (память, мышление и воображение).

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

Вследствие чего, у работника начинается снижение интереса к работе и работоспособность, проявление раздражительности и конфликтности, повышается количество ошибок в работе, психоэмоциональные сдвиги.

Для снижения эмоциональных перегрузок необходимы:

- умственные тренировки;
- повышение квалификации;
- умеренные и постоянные производственные нагрузки;
- выработка силы воли;
- правильное трудовое, психологическое и эстетическое воспитание;
- умение отходить от стрессовых состояний;
- повышение интереса к работе;
- создание положительных эмоций;
- оптимальное расписание отдыха.

5.4. Экологическая безопасность

Разработка проектного решения экологически безопасна, однако может косвенно влиять на атмосферу, так как работа компьютера связана с потреблением электроэнергии и нагревом аппаратных средств.

Электроэнергия вырабатывается на тэц, где сжигаются углеродные соединения. Офисный ПК потребляет около 100 Ватт электроэнергии в час. Энергоблок 200 МВт потребляет 90 тонн угля в час. Благодаря нехитрым вычислениям можно сделать вывод, что за час работы над проектным решением сжигается 45г угля в час. Если работа над проектным решением заняла 30 дней при работе 4 часа в день, то за все время работы было сожжено 5 кг 400 г угля. Соответственно выброс CO2 в атмосферу составит приблизительно 10 кубических метров.

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

5.5. Безопасность в чрезвычайных ситуациях

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

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

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

Главные причины возникновения пожара:

- короткое замыкание;
- опасная перегрузка сетей, которая ведет за собой сильный нагрев токоведущих частей и загорание изоляции;
- пуск оборудования после некорректного и неквалифицированного ремонта.

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

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

Для предупреждения возникновения пожаров необходимо исключить образование горючей среды, следить за применением при строительстве и отделке зданий несгораемых или трудносгораемых материалов.

Пожарно-профилактические мероприятия:

- организационные мероприятия, касающиеся технического процесса с учетом пожарной безопасности объекта (инструктаж персонала, обучение правилам техники безопасности, издание инструкций, плакатов, планов эвакуации);
- эксплуатационные мероприятия, рассматривающие эксплуатацию используемого оборудования (соблюдение эксплуатационных норм оборудования, обеспечение свободного подхода к оборудованию, поддержание исправности изоляции проводников);

- технические и конструктивные мероприятия, связанные с правильным размещением и монтажом электрооборудования и отопительных приборов (соблюдение противопожарных мероприятий при устройстве электропроводок, оборудования, систем отопления, вентиляции и освещения).

Для повышения устойчивости рабочего помещения к ЧС необходимо произвести установку систем противопожарной сигнализации, реагирующих на дым и другие продукты горения, установку огнетушителей. Также, два раза в год проводить учебные тревоги для отработки действий при пожаре.

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

Вывод по главе

В результате проведенного анализа был рассмотрен процесс разработки системы с правовой, экологической, производственной точек зрения, а также обеспечения безопасности при чрезвычайных ситуациях. Рабочее место соответствует всем необходимым нормам, кроме норм освещенности. Показатель был ниже нормы в 3 раза, но компенсировался естественным дневным светом. Поэтому такое отклонение не доставляло неудобств и его можно считать несущественными. По электробезопасности помещение относится к категории безопасное. По тяжести труда в соответствие с СанПин 1.2.3685-21 работа при разработке проектного решения относится к категории I. Рабочее помещение, представленное для выполнения ВКР, согласно СанПиН 1.2.3685-21, можно отнести к категории В (пожароопасное).

Заключение

При выполнении выпускной квалификационной работы был разработан вариант искусственного интеллекта поведения игровых объектов на двумерной игровой сцене.

Перед началом разработки была изучена предметная область, выявлены проблемы, продуманы пути их решения. Проведена предварительная работа по выбору средств разработки, а также проанализированы модули библиотеки Astrobox.

В ходе разработки была реализована модель поведения игровых объектов по заданным исходным свойствам.

Основные сложности, с которыми пришлось столкнуться - это тестирование и отладка поведения объектов под управлением ИИ. Запуск симуляции и разбор причин неудач по ней крайне затруднял скорость разработки.

Была проведена оценка коммерческого потенциала и перспективности разработки с позиции ресурсоэффективности и ресурсосбережения. Произведен расчет бюджета разрабатываемой системы, определяется финансовая, бюджетная, социальная и экономическая эффективность разработки.

Рассмотрены правовые и организационные вопросы обеспечения производственной безопасности и безопасности в чрезвычайных ситуациях.

На данный момент DidenkoDrone входит в топ 10 ИИ платформы.

Рейтинг по состоянию на 15.05.2022

Позиция	Имя команды	Рейтинг
1	VoychenkoDrones	1044
2	ZakharovDrone	925
3	ShapievDrone	901
4	KonovalovDrone	823
5	RochevDrone	799
6	ChashchinaDrone	794
7	DidenkoDrone	789
8	OlshannikovDron	785
9	PazukhinDrone	782
10	KodzaevDrone	779
11	UshkovDrone	772
12	IvanovDrone	769
13	KachanovDrone	756
14	EvteshinDrone	749
15	KazaryanDrone	749
16	BibikDrone	745
17	GoshkievDron	709
18	DevastatorDrone	693
19	RsadomskovDrone	667

Рисунок 7 - Рейтинг дронов среди разработчиков

За все время проведения рейтинговых игр разрабатываемые дроны показывали положительную динамику своей эффективности, стабильно показывая себя лучше других вариантов ИИ разработчиков. Так с 1.04 по 15.05 за 16 турниров рейтинг дрона вырос на 257 пунктов, т.е возрос примерно в 1,5 раза.

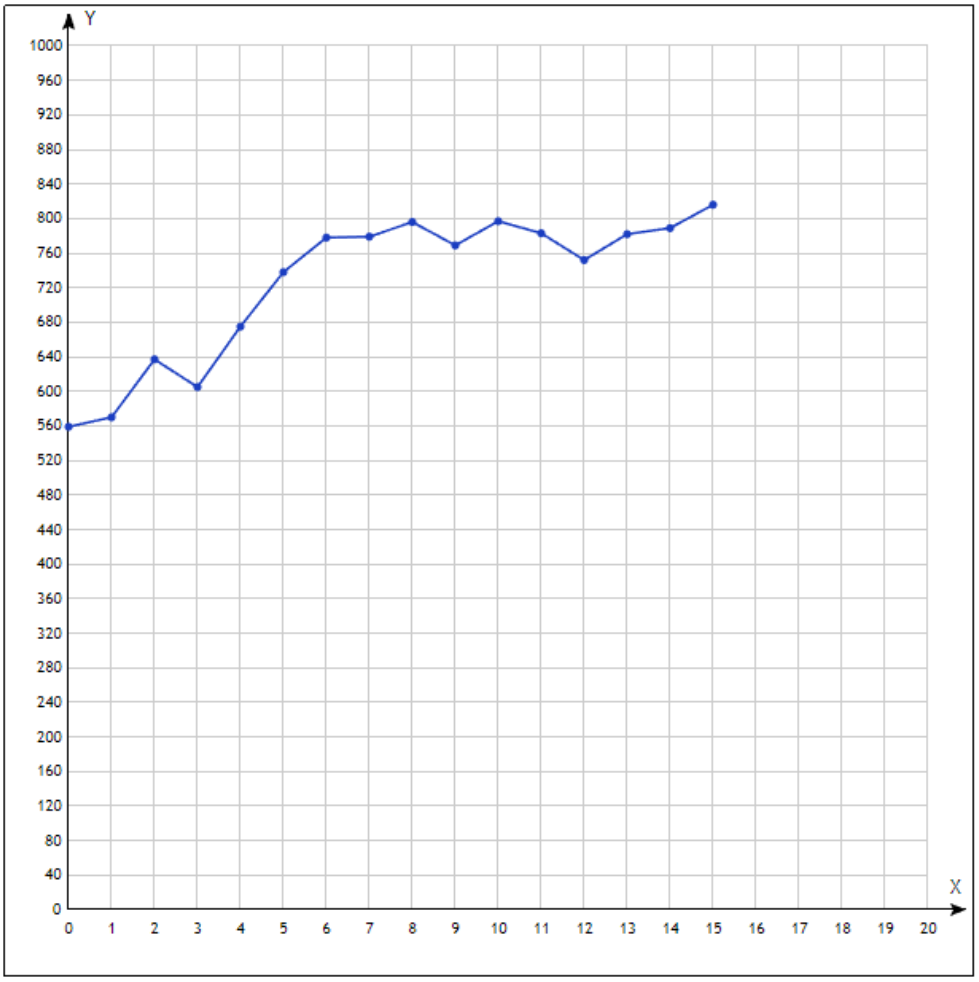

Рисунок 8 - Динамика рейтинга DidenkoDrone

По таким показателям можно с уверенностью говорить, что такой вариант ИИ для дронов Astrobox является жизнеспособным. Благодаря данной работе были освоены навыки разработки на языке Python, а также навыки в системе контроля версий Git. Отдельную благодарность хотелось объявить компании “Skillbox” за создание библиотеки Astrobox и помощи при разработке модулей.

Список источников

1. Python 3.7.3 documentation [Электронный ресурс]. URL: https://docs.python.org/3/ (Дата обращения 10.04.2022)
2. JSON в Python [Электронный ресурс]. URL: https://python-scripts.com/json (Дата обращения 10.04.2022)
3. Элементы нотации BPMN. [Электронный ресурс]. URL: https://www.elma-bpm.ru/bpmn2/7_2.html (Дата обращения 13.02.2022)
4. Методические указания по разработке раздела «Социальная ответственность» выпускной квалификационной работы магистра, специалиста и бакалавра всех направлений (специальностей) и форм обучения ТПУ/Сост. Е.Н. Пашков, И.Л. Мезенцева – Томск: Изд-во Томского политехнического университета, 2019. – 24 с.
5. SWOT анализ: основы метода и ключевые элементы [Электронный ресурс] URL: http://powerbranding.ru/biznes-analiz/swot/ (дата обращения: 12.05.2022).
6. Лутц М. Программирование на Python, том I, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
7. PyCharm: the Python IDE for Professional Developers by JetBrains [Электронный ресурс] URL: https://www.jetbrains.com/pycharm/(дата обращения: 5.02.2022).
8. Обучающий курс “Python-разработчик с нуля” [Электронный ресурс]. URL: https://go.skillbox.ru/profession/profession-dev-ops-pro/python-razrabotchik-s-nulya (дата обращения: 21.04.2022).

ПРИЛОЖЕНИЕ А

Интроспекция класса Drone:

['_CanLogging__logger', '_Drone__DeathAnimation', '_Drone__angle_of_death', '_Drone__dead_flight_speed', '_Drone__dead_game_step', '_Drone__death_animaion', '_Drone__heal_taken', '_Drone__health', '_GameObject__container', '_GameObject__objects_count', '_GameObject__scene', '_GameObject__team_name', '__class__', '__delattr__', '__dict__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__le__', '__lt__', '__module__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', '__unicode__', '__weakref__', '_cargo', '_check_runout', '_commands', '_events', '_gun', '_heartbeat', '_heartbeat_tics', '_log', '_mothership', '_move_target', '_runout', '_selected', '_sleep_countdown', '_sleep_state', '_sprite_filename', '_transition', 'add_command', 'add_event', 'animated', 'asteroids', 'auto_team', 'cargo', 'command', 'coord', 'counter', 'damage_taken', 'debug', 'direction', 'distance_to', 'error', 'free_space', 'fullness', 'game_step', 'gun', 'gun_cooldown', 'have_gun', 'health', 'id', 'info', 'is_alive', 'is_asleep', 'is_empty', 'is_full', 'is_moving', 'is_my_status_obj', 'layer', 'link_to_scene', 'load_from', 'logger', 'meter_1', 'meter_2', 'mothership', 'move_at', 'my_mothership', 'near', 'on_born', 'on_collide_with', 'on_heartbeat', 'on_load_complete', 'on_overlap_with', 'on_stop', 'on_stop_at_asteroid', 'on_stop_at_mothership', 'on_stop_at_point', 'on_stop_at_target', 'on_unload_complete', 'on_wake_up', 'payload', 'proceed_commands', 'proceed_events', 'radius', 'rotate_mode', 'scene', 'selectable', 'set_team', 'set_team_name', 'sleep_attrs', 'sprite_filename', 'stop', 'target', 'team', 'team_number', 'teammates', 'turn_to', 'unload_to', 'update_sleep_state', 'vector', 'warning', 'x', 'y', 'zoom']

Интроспекция класса SpaceField:

['_CELL_JITTER', '_CanLogging__logger', '_Scene__get_overlap_map', '_Scene__get_overlap_objects', '_Scene__overlap_map', '_Scene__teams', '_SpaceField__asteroids', '_SpaceField__drones', '_SpaceField__motherships', '__class__', '__delattr__', '__dict__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__le__', '__lt__', '__module__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', '__weakref__', '_check_collisions', '_detect_overlaps', '_fill_space', '_game_over_tics', '_game_statistics_printed', '_get_game_state', '_get_team_pos', '_log', '_make_game_result', '_prev_endgame_state', '_step', 'asteroids', 'check_collisions', 'debug', 'detect_overlaps', 'drones', 'error', 'field', 'game_speed', 'game_step', 'get_game_result', 'get_mothership', 'get_objects_by_type', 'get_objects_status',

'get_team_number', 'go', 'headless', 'hold_state', 'info', 'init_kwargs', 'logger', 'max_drones_at_team', 'motherships', 'name', 'objects', 'parent_conn', 'prepare', 'print_game_statistics', 'register_to_team', 'remove_object', 'teams', 'teams_count', 'time_sleep', 'ui', 'warning']

ПРИЛОЖЕНИЕ Б

Листинг кода класса DidenkoDrone:

```
class DidenkoDrone(Drone):
  all_drones = []

  def __init__(self, **kwargs):
    super(DidenkoDrone, self).__init__(**kwargs)
    self.target = None
    self.MIDDLE_LINE = 1.1
    self.fullness_statistics = []
    self.asteroids_to_mine = []
    self.defence_mode = False
    self.defence_point = None
    self.base_attack = False
    self.base_loot = False
    self.bases = None
    self.next_base = None
    self.direction_v = None
    DidenkoDrone.all_drones.append(self)

  def add_statistics(self, target):
    distance = self.distance_to(target)
    fullness = self.fullness
    self.fullness_statistics.append((distance, fullness))

  def move_at_with_statistics(self, target):
    self.move_at(target)
    self.add_statistics(target)

  def asteroids_are_empty(self):
    for asteroid in self.asteroids_to_mine:
      if asteroid.cargo.payload > 0:
```

```
            return False
        else:
            self.target = None
            return True

    def filter_asteroids_long_distance(self):
        asteroids_info = {}
        for asteroid in self.asteroids:
            asteroids_info[asteroid] = self.distance_to(asteroid)
        sorted_distance = sorted(asteroids_info.values(), reverse=True)
        middle = sum(sorted_distance) / len(sorted_distance) * self.MIDDLE_LINE
        sorted_asteroids = {}
        for i in sorted_distance:
            for k in asteroids_info.keys():
                if asteroids_info[k] == i:
                    sorted_asteroids[k] = asteroids_info[k]
                    break

        asteroids_info = sorted_asteroids
        for asteroid in asteroids_info:
            if middle >= asteroids_info[asteroid]:
                self.asteroids_to_mine.append(asteroid)

    def on_born(self):
        self.filter_asteroids_long_distance()
        self.target = self.id % 5 - 1
        self.move_at_with_statistics(self.asteroids_to_mine[self.target])
        pass

    def on_stop_at_asteroid(self, asteroid):
        if self.defence_mode is not True:
            self.load_from(asteroid)
            self.turn_to(self.my_mothership, 100)
```

```
def check_nearly_asteroids(self, current_aster):
    min_dist = 1000
    min_dist_asteroid = None
    for i, asteroid in enumerate(self.asteroids_to_mine):
        if asteroid is not current_aster and asteroid.cargo.payload != 0:
            if self.distance_to(asteroid) < min_dist:
                min_dist_asteroid = i
                min_dist = self.distance_to(asteroid)
    return min_dist_asteroid

def choose_target(self):
    self.target = self.get_full_asteroid()
    if self.target is None:
        self.target = self.get_non_empty_asteroid()

def check_destroyed_base(self):
    self.bases = self.scene.motherships
    for base in self.bases:
        if base == self.my_mothership:
            self.bases.remove(base)
    if self.bases is not None:
        for base in self.bases:
            if not base.is_alive and base.cargo.payload > 0:
                self.next_base = base
                return base
        else:
            return None
    else:
        return None

def on_heartbeat(self):
    if self.health < 2*MAX_HEALTH/3:
```

```
        self.move_at(self.my_mothership)
    else:
        if self.defence_mode and round(self.x) == round(self.defence_point.x) and \
                round(self.y) == round(self.defence_point.y) and not self.check_destroyed_base():
            near_enemy_drones = self.find_target(self.find_enemies())
            base = self.find_target(self.find_enemy_bases())
            if near_enemy_drones is not None:
                if self.distance_to(near_enemy_drones) > 500:
                    if base is not None:
                        if self.distance_to(base) > 620:
                            self.turn_to(base)
                            self.get_closer()
                            pass
                        else:
                            self.turn_to(base)
                            self.gun.shot(base)
                    else:
                        self.defence_mode = False
                        self.base_loot = True
                else:
                    self.turn_to(near_enemy_drones)
                    self.gun.shot(near_enemy_drones)
            else:
                if base is not None:
                    if self.distance_to(base) > 700:
                        self.turn_to(base)
                        self.get_closer()
                        pass
                    else:
                        self.turn_to(base)
                        self.gun.shot(base)
                else:
                    self.defence_mode = False
```

```
                self.base_loot = True

        elif self.check_destroyed_base() is not None and self.defence_mode:
            self.base_loot = True
            self.defence_mode = False
            self.bases = self.check_destroyed_base()
            self.move_at(self.bases)
        else:
            self.choose_target()

def on_load_complete(self):
    if self.is_full:
        self.move_at_with_statistics(self.my_mothership)
    else:
        if self.asteroids_are_empty():
            self.move_at_with_statistics(self.my_mothership)
        else:
            self.target = self.check_nearly_asteroids(self.asteroids_to_mine[self.target])
            if self.target is None:
                self.move_at_with_statistics(self.my_mothership)
            else:
                self.move_at_with_statistics(self.asteroids_to_mine[self.target])

def on_stop_at_mothership(self, mothership):
    if mothership == self.my_mothership:
        if self.fullness == 0 and self.asteroids_are_empty():
            if self.find_enemies():
                self.defence_base()
        else:
            if not self.defence_mode:
                self.unload_to(mothership)
            if self.target is not None:
                self.turn_to(self.asteroids_to_mine[self.target], 100)
```

```
    else:
        self.load_from(mothership)
        self.turn_to(self.my_mothership, 100)

def on_unload_complete(self):
    if self.asteroids_are_empty():
        if not self.base_loot:
            self.defence_base()
        else:
            if self.next_base.payload == 0:
                self.next_base = None
                self.defence_base()
            else:
                self.move_at(self.next_base)
    else:
        self.choose_target()
        if self.target is None:
            self.defence_base()
        else:
            self.move_at_with_statistics(self.asteroids_to_mine[self.target])

def get_full_asteroid(self):
    full_aster = []
    for i in range(len(self.asteroids_to_mine)):
        if self.asteroids_to_mine[i].cargo.payload >= 100:
            full_aster.append(i)
    if full_aster:
        rnd = random.randint(0, len(full_aster) - 1)
        return full_aster[rnd]
    else:
        return None
```

```
    def get_non_empty_asteroid(self):
        non_empty_aster = []
        for i in range(len(self.asteroids_to_mine)):
            if 0 < self.asteroids_to_mine[i].cargo.payload < 100:
                non_empty_aster.append(i)
        if non_empty_aster:
            rnd = random.randint(0, len(non_empty_aster) - 1)
            return non_empty_aster[rnd]
        else:
            return None


    def defence_base(self):
        defence_direction = self.find_near_alive_base()
        if defence_direction is not None:
            self.base_loot = False
            self.defence_mode = True
            self.direction_v = Point(round(defence_direction.x) - round(self.x),
round(defence_direction.y - self.y))
            try:
                direction_cos = (self.direction_v.x * 1 + defence_direction.y * 0)/(
                    math.sqrt(self.direction_v.x**2 + self.direction_v.y**2))
            except Exception:
                direction_cos = 0.5
            if -0.35 < direction_cos <= 0.35:
                # top/bot
                if self.direction_v.y > 0:
                    if self.x < 500:
                        x = (self.x - 30) + self.id % 5 * 60
                        y = self.y + 120
                    else:
                        x = (self.x + 30) - self.id % 5 * 60
                        y = self.y + 120
                else:
```

```
            if self.x < 500:
                x = (self.x - 30) + self.id % 5 * 60
                y = self.y - 120
            else:
                x = (self.x + 30) - self.id % 5 * 60
                y = self.y - 120
    elif 0.95 < direction_cos or -0.95 > direction_cos:
        # left/right
        if self.direction_v.x > 0:
            if self.y < 300:
                x = self.x + 120
                y = (self.y - 30) + self.id % 5 * 60
            else:
                x = self.x + 120
                y = (self.y + 30) - self.id % 5 * 60
        else:
            if self.y < 300:
                x = self.x - 120
                y = (self.y - 30) + self.id % 5 * 60
            else:
                x = self.x - 120
                y = (self.y + 30) - self.id % 5 * 60
    else:
        # diagonal
        if self.direction_v.y > 0 and self.direction_v.x > 0:
            # enemy top/right
            x = 375 - (self.id - 1) * 75
            y = 75 + (self.id - 1) * 75
        elif self.direction_v.y > 0 > self.direction_v.x:
            # enemy top/left
            x = 700
            y = 250 + (self.id % 5 - 1) * 75
        elif self.direction_v.y < 0 and self.direction_v.x < 0:
```

```
                # enemy bot/left
                x = self.x - 300 + (self.id % 5) * 75
                y = self.y - (self.id % 15) * 75
            else:
                # enemy bot/right
                x = 375 - (self.id % 10 - 1) * 75
                y = self.y - (self.id % 10 - 1) * 75
        self.defence_point = Point(x, y)
        self.move_at(self.defence_point)
    else:
        x = 600 + self.id % 5 * 60
        y = 400 + self.id % 2 * 60
        self.defence_point = Point(x, y)
        self.move_at(self.defence_point)
    self.base_loot = False
    self.defence_mode = True
    self.defence_point = Point(x, y)
    self.move_at(self.defence_point)

def find_near_alive_base(self):
    min_dist = 2000
    min_dist_base = None
    bases = self.find_enemy_bases()
    for base in bases:
        if min_dist > self.distance_to(base) and base.is_alive:
            min_dist = self.distance_to(base)
            min_dist_base = base
    return min_dist_base

def find_enemy_bases(self):
    bases = self.scene.motherships
    for base in bases:
        if base == self.my_mothership or not base.is_alive:
```

```
            bases.remove(base)
    return bases

def find_enemies(self):
    enemies = []
    for drone in self.scene.get_objects_by_type(Drone):
        if drone.team != self.team and drone.is_alive:
            enemies.append(drone)
    return enemies

def find_target(self, enemies):
    min_dist = 10000
    fire_target = None
    for enemy in enemies:
        if min_dist > self.distance_to(enemy):
            min_dist = self.distance_to(enemy)
            fire_target = enemy
    return fire_target

def get_closer(self):
    attack_x = self.x
    attack_y = self.y
    attack_x += round(self.direction_v.x / 50)
    attack_y += round(self.direction_v.y / 50)
    self.defence_point = Point(attack_x, attack_y)
    self.move_at(self.defence_point)
```