

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

Инженерная школа информационных технологий и робототехники
 Направление подготовки 09.04.02 «Информационные системы и технологии»
 Отделение школы (НОЦ) Отделение информационных технологий

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема работы
Разработка программного обеспечения в виртуальной реальности для развития эвристического мышления

УДК 004.415.2:004.946:37.025.7

Студент

Группа	ФИО	Подпись	Дата
8ИМ02	Рогозин Руслан Александрович		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Демин Антон Юрьевич	к.т.н., доцент		

Консультант

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ст. преп. ОИТ	Видман Виталий Викторович	–		

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент	Былкова Татьяна Васильевна	канд.экон.наук		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор	Федоренко Ольга Юрьевна	д.м.н.		

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Савельев Алексей Олегович	к.т.н., доцент		

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ООП
по направлению 09.04.02 Информационные системы и технологии

Код компетенции	Наименование компетенции
Универсальные компетенции	
УК(У)-1	Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий
УК(У)-2	Способен управлять проектом на всех этапах его жизненного цикла
УК(У)-3	Способен организовывать и руководить работой команды, вырабатывая командную стратегию для достижения поставленной цели
УК(У)-4	Способен применять современные коммуникативные технологии, в том числе на иностранном(-ых) языке(-ах), для академического и профессионального взаимодействия
УК(У)-5	Способен анализировать и учитывать разнообразие культур в процессе межкультурного взаимодействия
УК(У)-6	Способен определять и реализовывать приоритеты собственной деятельности и способы её совершенствования на основе самооценки
Общепрофессиональные компетенции	
ОПК(У)-1	Способен самостоятельно приобретать, развивать и применять математические, естественно-научные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте
ОПК(У)-2	Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач
ОПК(У)-3	Способен анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями
ОПК(У)-4	Способен применять на практике новые научные принципы и методы исследований
ОПК(У)-5	Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем
ОПК(У)-6	Способен использовать методы и средства системной инженерии в области получения, передачи, хранения, переработки и представления информации посредством информационных технологий
ОПК(У)-7	Способен разрабатывать и применять математические модели процессов и объектов при решении задач анализа и синтеза распределённых информационных систем и систем поддержки принятия решений
ОПК(У)-8	Способен осуществлять эффективное управление разработкой программных средств и проектов
Профессиональные компетенции	

ПК(У)-1	Способен управлять программно-техническими, технологическими и человеческими ресурсами
ПК(У)-2	Способен управлять развитием баз данных
ПК(У)-3	Способен управлять работами по сопровождению и проектами создания (модификации) информационных систем, автоматизирующих задачи организационного управления и бизнес-процессы.
ПК(У)-4	Способен проектировать и организовывать учебный процесс по образовательным программам с использованием современных образовательных технологий.
ПК(У)-5	Способен осуществлять руководство разработкой комплексных проектов на всех стадиях и этапах выполнения работ.

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

Инженерная школа информационных технологий и робототехники
 Направление подготовки 09.04.02 «Информационные системы и технологии»
 Отделение школы (НОЦ) Отделение информационных технологий

УТВЕРЖДАЮ:
 Руководитель ООП
 _____ Савельев А. О.
 (Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Магистерской диссертации

Студенту:

Группа	ФИО
8ИМ02	Рогозин Руслан Александрович

Тема работы:

Разработка программного обеспечения в виртуальной реальности для развития эвристического мышления	
Утверждена приказом директора (дата, номер)	14.03.2022 № 45-48/с

Срок сдачи студентом выполненной работы:	07.06.2022
--	------------

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Исходные данные к работе	Техническое задание к реализации ПО в виртуальной реальности для развития эвристического мышления
Перечень подлежащих исследованию, проектированию и разработке вопросов	1. Обзор используемых технологий 2. Проектирование ПО 3. Реализация ПО 4. Тестирование ПО 5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение 6. Социальная ответственность
Перечень графического материала	Рисунок 1 – Логотип Unity Рисунок 2 – Логотип Unreal engine Рисунок 3 – Логотип Godot Рисунок 4 – Логотип GameMaker Рисунок 5 – Диаграмма классов игровой механики

<p>Рисунок 6 – Диаграмма классов системы эвристического отбора</p> <p>Рисунок 7 – Код скрипта компонента InputControllerComponent</p> <p>Рисунок 8 – Код скрипта компонента InteractiveComponent</p> <p>Рисунок 9 – Код скрипта компонента JointGroup</p> <p>Рисунок 10 – Код скрипта компонента TagGhost</p> <p>Рисунок 11 – Код скрипта системы AutoInteractiveJointGroup</p> <p>Рисунок 12 – Код скрипта системы AutoInteractiveHandType</p> <p>Рисунок 13 – Код скрипта системы ResetDataSystem</p> <p>Рисунок 14 – Код скрипта системы InputControllerSystem</p> <p>Рисунок 15 – Код скрипта системы MoveHandSystem</p> <p>Рисунок 16 – Код скрипта системы TriggerEventHandWithInteractive</p> <p>Рисунок 17 – Код скрипта системы CollisionEventInteractive</p> <p>Рисунок 18 – Код скрипта системы ObjectSelectionSystem</p> <p>Рисунок 19 – Код скрипта системы GripSelectedObjectSystem</p> <p>Рисунок 20 – Код скрипта системы DisableJointObjects</p> <p>Рисунок 21 – Код скрипта системы CreateJointObjects</p> <p>Рисунок 22 – Код скрипта системы DisableGravityJointObjects</p> <p>Рисунок 23 – Структура данных Data</p> <p>Рисунок 24 – Структура данных BaseData</p> <p>Рисунок 25 – Структура данных Filters</p> <p>Рисунок 26 – Enum Amount</p> <p>Рисунок 27 – Enum Copies</p> <p>Рисунок 28 – Enum Get</p> <p>Рисунок 29 – Метод прямого отбора данных</p> <p>Рисунок 30 – Метод косвенного отбор данных</p> <p>Рисунок 31 – Реализация фильтров Amount, Copies и Get</p> <p>Рисунок 32 – Выделение объектов при приближении руки</p> <p>Рисунок 33 – Удерживание выделенных объектов в руках</p> <p>Рисунок 34 – Соединение объектов</p> <p>Рисунок 35 – Заполнение исходной базы</p> <p>Рисунок 36 – Создание фильтра и выполнение метода Get</p> <p>Рисунок 37 – Результат тестирования</p> <p>Рисунок 38 – Создание нового фильтра и выполнение метода Get</p> <p>Рисунок 39 – Результат тестирования</p> <p>Рисунок 40 – Заполнение базы</p> <p>Рисунок 41 – Фильтр системы эвристического отбора</p> <p>Рисунок 42 – Результат тестирования с одним предметом</p> <p>Рисунок 43 – Результат тестирования с тремя предметами</p> <p>Рисунок 44 – Результат тестирования со всеми предметами</p> <p>Рисунок 45 – План эвакуации при пожаре и других ЧС</p>

Консультанты по разделам выпускной квалификационной работы	
Раздел	Консультант
Финансовые менеджмент, ресурсоэффективность и ресурсосбережение	Былкова Татьяна Васильевна
Социальная ответственность	Федоренко Ольга Юрьевна
Названия разделов, которые должны быть написаны на русском и иностранном языках:	
Проблемы и перспективы эвристического развития	

Problems and prospects of heuristic development	Пичугова Инна Леонидовна
---	-----------------------------

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	
--	--

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Демин Антон Юрьевич	к.т.н., доцент		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИМ02	Рогозин Руслан Александрович		

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

Инженерная школа информационных технологий и робототехники
Направление подготовки 09.04.02 «Информационные системы и технологии»
Отделение школы (НОЦ) Отделение информационных технологий
Уровень образования магистратура
Период выполнения Осенний / весенний семестр 2021/2022 учебного года

Форма представления работы:

Магистерской диссертации

Разработка программного обеспечения в виртуальной реальности для развития эвристического мышления

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы: 06.06.2022

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела
28.12.2021– 31.05.2022	Основная часть	85
01.05.2022 – 07.05.2022	Написание раздела «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»	5
08.05.2022 – 10.05.2022	Написание раздела «Социальная ответственность»	5
11.05.2022 – 06.05.2022	Написание раздела на иностранном языке	5

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Демин Антон Юрьевич	к.т.н., доцент		

Принял студент:

ФИО	Подпись	Дата
Рогозин Руслан Александрович		

СОГЛАСОВАНО:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Савельев Алексей Олегович	к.т.н., доцент		

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

Студенту:

Группа	ФИО
8ИМ02	Рогозин Руслан Александрович

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	Магистратура	Направление/специальность	09.04.02 «Информационные системы и технологии»

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

<i>1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих</i>	Стоимость материальных ресурсов определялась по средней рыночной стоимости. Оклады в соответствии с окладами сотрудников организации.
<i>2. Нормы и нормативы расходования ресурсов</i>	30% районный коэффициент
<i>3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</i>	30% отчисления в социальные внебюджетные фонды)

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

<i>1. Оценка коммерческого и инновационного потенциала НТИ</i>	Провести предпроектный анализ
<i>2. Разработка устава научно-технического проекта</i>	Представить Устав научного проекта магистерской работы
<i>3. Планирование процесса управления НТИ: структура и график проведения, бюджет, риски и организация закупок</i>	Разработать план управления НТИ
<i>4. Определение ресурсной, финансовой, экономической эффективности</i>	Определение интегрального финансового показателя разработки Определение интегрального показателя ресурсоэффективности разработки Определение интегрального показателя эффективности

Перечень графического материала (с точным указанием обязательных чертежей):

<ol style="list-style-type: none"> <i>1. «Портрет» потребителя результатов НТИ</i> <i>2. Оценка конкурентоспособности технических решений</i> <i>3. Матрица SWOT</i> <i>4. График проведения и бюджет НТИ</i> <i>5. Оценка ресурсной, финансовой эффективности НТИ</i>

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

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент	Былкова Татьяна Васильевна	канд.экон.наук		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИМ02	Рогозин Руслан Александрович		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8ИИМ02	Рогозин Руслан Александрович

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	Магистратура	Направление/специальность	09.04.02 «Информационные системы и технологии»

Тема ВКР:

Разработка программного обеспечения в виртуальной реальности для развития эвристического мышления

Исходные данные к разделу «Социальная ответственность»:

<p>1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения</p>	<p>Объект исследования: компьютерный игровой проект, представляющий собой определённые игровые механики, вместе образующие игровой процесс, реализованный с помощью игровой среды Unity на персональном компьютере.</p> <p>Работа велась в учебной аудитории площадью 28 м² с водяным отоплением центрального теплоснабжения, где используется приточная вентиляция, естественное освещение, потолочные лампы для прямого освещения и 5 стационарных компьютеров.</p> <p>Область применения: обучающая и досугово-развлекательная деятельность.</p>
--	--

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

<p>2. Правовые и организационные вопросы обеспечения безопасности при разработке проектного решения:</p> <ul style="list-style-type: none"> — специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; — организационные мероприятия при компоновке рабочей зоны. 	<ul style="list-style-type: none"> — Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 09.03.2021); — ГОСТ 12.2.032-78 ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования; — ГОСТ 12.0.003-2015 Опасные и вредные производственные факторы; — СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания; — ГОСТ 12.1.005-88 Система стандартов безопасности труда (ССБТ). Общие санитарно-гигиенические требования к воздуху рабочей зоны — СП 52.133330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95; — МР 2.2.9.2311 – 07 «Профилактика стрессового состояния работников при различных видах профессиональной деятельности»; — ГОСТ 12.1.003-2014 ССБТ. Шум. Общие требования безопасности; — ГОСТ 12.1.045-84 ССБТ. Электростатические поля. Допустимые уровни на рабочих местах и требования к проведению контроля;
--	---

	<ul style="list-style-type: none"> — ГОСТ 12.1.019-2017 ССБТ. Электробезопасность. Общие требования и номенклатура видов защиты; — ГОСТ 12.1.004-91 ССБТ. Пожарная безопасность. Общие требования; — ГОСТ 12.1.030-81 ССБТ. Электробезопасность. Защитное заземление. Зануление; — ГОСТ 12.1.038-82 ССБТ. Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов.
<p>3. Производственная безопасность при разработке проектного решения</p> <ul style="list-style-type: none"> — Анализ выявленных вредных и опасных факторов; — Обоснование мероприятий по снижению воздействия. 	<p>Вредные факторы: электромагнитное излучение, повышенный уровень шума, слабая освещенность, отклонения показателей микроклимата, нервно-психические перегрузки.</p> <p>Опасные факторы: факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которого попадает работающий.</p> <p>Требуемые средства коллективной и индивидуальной защиты от выявленных факторов: беруши, экранирующая одежда, дополнительные источники света.</p> <p>Расчет: расчет системы искусственного освещения</p>
<p>4. Экологическая безопасность при разработке проектного решения:</p>	<p>Воздействие на селитебную зону, гидросферу и атмосферу отсутствует.</p> <p>Воздействие на литосферу: утилизация компьютера и периферийных устройств (принтеры, МФУ, веб-камеры, наушники, колонки, телефоны), люминесцентных ламп, канцелярских принадлежностей, макулатуры.</p>
<p>5. Безопасность в чрезвычайных ситуациях при разработке проектного решения:</p>	<ul style="list-style-type: none"> — Возможные ЧС: пожары, ураганы, оползни, наводнения. <p>Наиболее типичная ЧС: пожары.</p>

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

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор	Федоренко Ольга Юрьевна	д.м.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИМ02	Рогозин Руслан Александрович		

Содержание

Реферат	13
ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ, НОРМАТИВНЫЕ ССЫЛКИ	14
Введение	15
1 Обзор используемых технологий.....	17
1.1 Выбор среды разработки видеоигр	17
1.1.1 Unity.....	17
1.1.2 Unreal Engine	17
1.1.3 Godot	18
1.1.4 GameMaker: Studio 2.....	18
1.1.5 Сравнение сред разработки видеоигр	19
1.2 Среда разработки JetBrains Rider.....	20
1.3 Обзор технологии DOTS.....	20
2 Проектирование ПО	23
2.1 Соответствие ПО эвристическому подходу.....	23
2.2 Проектирование игровой механики	25
2.3 Проектирование системы эвристического отбора.....	27
3 Реализация ПО.....	28
3.1 Реализация игровой механики соединения/разъединения объектов	28
3.2 Реализация системы эвристического отбора	36
4 Тестирование ПО.....	40
4.1 Тестирование игровой механики соединения/разъединения объектов.....	40
4.2 Тестирование системы эвристического отбора.....	41
4.3 Тестирование системы эвристического отбора с игровой механикой.....	42
5 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение.....	44
5.1 Предпроектный анализ	44
5.1.1 Потенциальные потребители результатов исследования	44
5.1.2 Анализ конкурентных технических решений с позиции ресурсоэффективности и ресурсосбережения	45
5.1.3 SWOT-анализ	46
5.1.4 Оценка готовности проекта к коммерциализации.....	48
5.1.5 Методы коммерциализации результатов научно-технического исследования	49
5.2 Инициация проекта	50
5.3 Планирование управления научно-техническим проектом.....	51
5.3.1 План проекта	51
5.3.2 Потенциальные риски	54
5.4 Бюджет научно-технического исследования	54
5.5 Оценка сравнительной эффективности исследования	58

6 Социальная ответственность	60
6.1 Правовые и организационные вопросы обеспечения безопасности	60
6.1.1 Специальные правовые нормы трудового законодательства	60
6.1.2 Мероприятия по планировке рабочего места	60
6.2 Производственная безопасность	62
6.2.1 Отклонения показателей микроклимата	63
6.2.2 Недостаточная освещенность	64
6.2.3 Повышенный уровень электромагнитных излучений	66
6.2.4 Нервно-психические перегрузки	67
6.2.5 Повышенный уровень шума	67
6.2.6 Удар электрическим током	68
6.2.7 Определение мероприятий по уменьшению вреда от воздействия выявленных факторов	69
6.3 Экологическая безопасность	70
6.3.1 Производство и утилизация техники и комплектующих	71
6.3.2 Нормы экологической безопасности	71
6.3.3 Мероприятия по сохранению экологической безопасности	71
6.4 Безопасность в чрезвычайных ситуациях	72
6.4.1 Пожар	72
6.5 Вывод по разделу	74
Заключение	75
Список использованных источников	76
Приложение А	79

Реферат

Выпускная квалификационная работа содержит 90 страниц, 45 рисунков, 30 таблиц, 23 источника и 1 приложение.

Ключевые слова: Эвристика, программное обеспечение, Unity.

Объектами исследования является ПО под шлемы виртуальной реальности.

Цель работы – Разработка ПО под VR для развития эвристического мышления.

В процессе выполнения работы использовался игровой движок Unity и среда разработки Rider, программа реализована на языке C# с использованием внутренних библиотек Unity при помощи технологии DOTS.

В результате выполнения работы было разработано ПО для устройств виртуальной реальности развивающее эвристическое мышление.

Область применения: Развитие эвристического мышления среди людей возраста от 7 до 25 лет.

В будущем планируется использовать текущее разработанное ПО для разработки полноценной видеоигры.

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ, НОРМАТИВНЫЕ ССЫЛКИ

Скрипт – в Unity отдельный класс на языке C#, который взаимодействует с определенными объектами и их компонентами.

Игровой движок – Среда разработки игр, включающая возможности симуляции физики, работы со светом, работы с графикой, работы с игровой логикой, сборки под разные платформы.

ПО – Программное обеспечение.

Эвристика – в широком смысле — наука о творчестве. В узком, более современном — теория и практика организации избирательного поиска при решении сложных интеллектуальных задач. Эвристические компьютерные программы отличаются от программ систематического поиска, поскольку содержат критерии ограничения поиска в пространстве большого количества возможностей — эвристики — и не гарантируют обязательное решение задач.

Введение

Эвристика представляет собой специфическую научную отрасль, исследующую созидательную деятельность. Ввиду развития этой области знания на протяжении не менее двух тысячелетий существует комплексное представление о понятии «эвристика» как таковом. В общем эвристика – это творческая деятельность, направленная на решение нестандартных задач.

Проблема данного термина заключается в том, что для его понимания человеку необходимо вспомнить, когда он решал сложные задачи, путем творческой деятельности. Получается значение слова эвристика ссылается на воспоминания читающего. Есть вероятность, что воспоминания о творческом процессе окажутся на самом деле не творческими. Что делает текст не лучшим способ передачи информации для данного термина.

Поэтому было решено разработать ПО в виртуальной реальности для понимания эвристики и дальнейшего развития эвристического мышления. Данное ПО будет содержать в себе творческую игровую механику и систему эвристического отбора в зависимости от принятого решения.

Целями практики является формирование у обучающихся определенного состава компетенций для подготовки к профессиональной деятельности, в частности:

- Способен анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями
- Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем
- Способен управлять процессами и проектами по созданию (модификации) информационных ресурсов

Целью данной ВКР является разработка ПО в виртуальной реальности для понимания и развития эвристического мышления. Игра будет представлена на платформе Windows и выполнена при помощи среды разработки игр Unity используя технологию DOTS.

Для выполнения поставленной цели необходимо разрешить следующие

задачи:

1. Изучить источники по выбранной теме;
2. Произвести анализ предметной области;
3. Выбрать инструмент разработки;
4. Проектирование ПО;
5. Разработать ПО;
6. Тестирование ПО.

1 Обзор используемых технологий

В ходе выполнения выпускной квалификационной работы были использованы различные технологии, требующий описания.

1.1 Выбор среды разработки видеоигр

В начале разработки ПО необходимо выбрать среду разработки, так как нужно разработать игровую механику, то выбирать нужно из сред разработки видеоигр. Далее, для более точного придерживания к профессиональному лексикону, среда разработки игр будет называться игровым движком.

1.1.1 Unity

Unity [1] — межплатформенная среда разработки компьютерных игр, разработанная американской компанией Unity Technologies [2]. Unity представляет из себя один из самых популярных игровых движков на сегодняшний день. Логотип Unity представлен на рисунке 1.



Рисунок 1 – Логотип Unity

1.1.2 Unreal Engine

Unreal Engine 4 [3] — игровой движок, разрабатываемый и поддерживаемый компанией Epic Games [4]. Unreal Engine, наряду с Unity, является одним из самых популярных движков. В отличие от Unity, он имеет более высокий порог входа и специализирован больше для крупных проектов. Логотип Unreal engine представлен на рисунке 2.



Рисунок 2 – Логотип Unreal engine

1.1.3 Godot

Godot Engine [5] — открытый кроссплатформенный 2D и 3D игровой движок под лицензией MIT, который разрабатывается сообществом Godot Engine Community [6]. Godot — амбициозный новичок среди игровых движков, своими акцентами на небольшие 2D и редко 3D играми он очень похож на GameMaker. Логотип Godot представлен на рисунке 3.



Рисунок 3 – Логотип Godot

1.1.4 GameMaker: Studio 2

GameMaker: Studio [7] — Мощный инструмент для профессиональной разработки игр. Является одним из самых популярных игровых движков. Разрабатывается компанией YoYo Games [8]. GameMaker Studio отлично подходит для производства небольших 2D или простых 3D игр. GameMaker слабо используется среди игровых студий, а значит знания данного игрового движка не сильного помогут при дальнейшем поиске работы. Логотип GameMaker Studio представлен на рисунке 4.



Рисунок 4 – Логотип GameMaker Studio





1.1.5 Сравнение сред разработки видеоигр

Чтобы выбрать правильный игровой движок, будем сравнивать их по следующим критериям:

1. Доступность – стоимость использования игрового движка;
2. Кроссплатформенность – возможность запускать проекты на разных целевых платформах;
3. Работа с 3D – игровой движок должен хорошо работать в 3D пространстве, так как ВКР проект подразумевается под виртуальную реальность;
4. Магазин ресурсов – игровой движок должен иметь магазин ресурсов, который позволит получать или приобретать дополнительные полезные функции для разработки видеоигр;
5. Вакансии – важно использовать игровым движком, который в дальнейшем может дать оплачиваемое рабочее место.

В таблице 1 представлены результаты сравнения игровых движков по выбранным критериям.

Таблица 1 – Сравнение игровых движков по выбранным критериям

	 unity	 UNREAL ENGINE	 GODOT Game engine	 GameMaker Studio 2
Доступность	✓	✓	✓	✗
Кроссплатформенность	✓	✓	✓	✓
Работа с 3D	✓	✓	✓	✗
Магазин ресурсов	✓	✓	✗	✓
Вакансии	✓	✓	✗	✗

В результате, объективным критериям соответствуют два игровых движка Unity и Unreal Engine. Сделать выбор в сторону одного из них поможет язык программирования данных движков. Unity использует язык C#, а Unreal

Engine использует C++. В итоге выбор пал на игровой движок Unity так, как личные знания языка C# превосходят знания языка C++.

1.2 Среда разработки JetBrains Rider

JetBrains Rider — кроссплатформенная интегрированная среда разработки программного обеспечения для платформы .NET, разрабатываемая компанией JetBrains. В данном случае важно, что Rider может выступать редактором исходного кода для Unity с возможностью подсветки разных частей кода и автодополнением. Также в Rider уже встроена поддержка Unity [9].

1.3 Обзор технологии DOTS

Стек информационно-ориентированных технологий (DOTS) позволяет использовать все возможности новых многоядерных процессоров без создания дополнительных сложных программных алгоритмах. Пакет технологии DOTS находится все еще в разработке, который получает постоянные обновления [10]:

1. Entities (ECS) – это переход от объектно-ориентированного подхода к информационно-ориентированному. Entities успешно использует систему Job C# и Burst Compiler, раскрывая всю мощь современных многоядерных процессоров;
2. System Job C# - использует возможности современных многоядерных процессоров. Система разработана с целью упростить программистам C# разработку безопасного, быстрого и параллелизуемого кода.;
3. Компилятор Burst - новейшая на LLVM вычислительная технология бэкенд-компиляции превращает C# код в оптимизированный машинный код;
4. Unity Physics - технология DOTS, также предоставляет новый физический движок, который поможет моделировать физику без ущерба производительности, подходящая к современным требованиям для сетевого обмена данными;

5. Unity NetCode – инструмент для оптимизации сетевого кода, помогающий прогнозировать на стороне клиента, «авторитетный» сервер и с интерполяцией, упрощающий разработку шутеров и не только;
6. DSPGraph - новейшая звуковая подсистема низкого уровня, которая совместима с компилятором Burst, что позволяет программистам и разработчикам звукового оформления разрабатывать собственные звуковые системы;
7. Unity Animation – инструмент для создания анимаций, включая инверсную кинематику, смешивание, маски и слои.

ECS — это архитектурный шаблон для проектирования проекта состоящий из "Сущность Компонент Система". Где "Сущность" (Entity) — объекты-контейнеры, не имеющие свойств, но являющийся хранилищем для "Компонентов". "Компонент" — это блоки с данными, которые хранят всевозможные свойства любых игровых объектов и/или событий. И все эти данные, находящиеся в контейнерах, обрабатываются логикой, существующей только в виде "Систем" — это классы с определенными методами для изменения компонент. Данный архитектурный паттерн является независимым от любой среды разработки и имеет множество реализаций. Но каждый "движок" по-своему реализует инициализацию ECS элементов.

В стандартном Unity (без ECS) используются GameObject, как "Сущности", а наследниками класса MonoBehaviour являются "Компоненты" и "Системы". В ECS логика всегда должна быть отдельно от данных. Это позволяет очень гибко управлять логикой, не ломая данные. Другим плюсом является то, что данные обрабатываются поточно (за счет Job system) в каждой системе и независимо от "движка", в отличие от MonoBehaviour, который много взаимодействует с "Native"-частью, что делает проект менее производительным [11].

Архитектура Entity Component System (ECS) разделяет идентичность (сущности), данные (компоненты) и поведение (системы). Архитектура ориентирована на данные. Системы считывают потоки данных компонентов, а затем преобразуют данные из состояния ввода в состояние вывода, которые затем индексируются.

2 Проектирование ПО

2.1 Соответствие ПО эвристическому подходу

От стандартной учебной игры эвристическую игру отличает творческая составляющая, личностный характер и большая степень открытости. [24]

Чтобы добиться творческой составляющей при разработке ПО, нужно убедиться, что действия (геймплей) игрока являются уникальными. К примеру, если в игру добавить решение математического уравнения, то очевидно, что для людей, разбирающихся в математике, решение задачи не будет являться творческим. Из этого следует, что геймплей должен быть достаточно абстрактным.

Для того, чтобы геймплей был творческим, необходимо, чтобы задача не решалась с помощью комбинаторики. Например, замок с n -значным кодом можно открыть, перебирая значения. Конечно, получение кода от замка может и является творческой задачей, но само существование решения с помощью комбинаторики уже делает его шаткой творческой задачей. В случае, если существование комбинаторики в геймплее неизбежно, то необходимо акцентировать внимание игрока на творческих возможностях.

Добиться проявления личностного характера возможно с помощью нелинейности игрового процесса. Для реализации нелинейности игрового процесса не обязательно наличие нескольких концовок. Нелинейности можно добиться с помощью существования различных путей к одной и той же цели, что и делает путь игрока более личностным. Большая степень открытости в эвристической игре задается с помощью описанных выше требований к творческой составляющей и личностному росту.

Решение задачи в эвристической ПО должно быть:

- Новой для всех игроков (абстрактной);
- Нерешаемой с помощью комбинаторики (или заставить игрока не использовать комбинаторику);

— Открытой образовательной средой с отсутствием заранее известного результата.

Перейдем к описанию геймплея на основе выше упомянутых пунктов. Напомню, что видеоигра разрабатывается для VR устройств. Данный выбор обусловлен уникальными возможностями ввода данных с помощью VR шлема и контроллеров.

В основе ПО лежит игровая механика, позволяющая соединять объекты, которые можно взять в руки. Соединять их можно как угодно в любую произвольную форму. Также разъединение предметов работает противоположно соединению. Достаточно просто взять два соединённых объекта в разные руки, чтобы разъединить их.

И последнее, что остается сделать это систему эвристического отбора, которая выдает результаты по множеству возможных вариаций соединённых предметов.

Система эвристического отбора хранит в себе базу исходных данных о возможных комбинациях соединённых предметов. Отбор данных имеет два режима поиска: прямой отбор данных и косвенный отбор данных.

Прямой отбор данных подразумевает полное совпадение значений входного набора элементов с исходным набором элементов.

Косвенный отбор данных подразумевает совпадение значений входного набора элементов с исходным набором элементов, но без учета полного совпадения по количеству элементов. Особенностью косвенного отбора является настройка фильтров отбора, которая представлена в таблице 2.

Таблица 2 – Значения фильтров косвенного отбора

Название	Значение	Описание
Amount	Min	Выводит результат с минимальным количеством элементов в исходной базе
	Max	Выводит результат с максимальным количеством элементов в исходной базе
Copies	None	Выводит результат в котором нет одинаковых элементов
	Include	Выводит результат в котором, как есть, так и нет одинаковых элементов
	Only	Выводит результат в котором есть одинаковых элементов

Get	All	Если результатов несколько, то выводит все
	Random	Если результатов несколько, то выводит один случайный из них

Если нет ни одного совпадения входных данных с исходным набором данных, то система возвращает “Нет данных” в обоих случаях.

2.2 Проектирование игровой механики

На рисунке 5 представлена диаграмма классов игровой механики соединения/разъединения объектов. На которой в правой части расположены компоненты, хранящие данные, а в левой части системы, обрабатывающие данные.

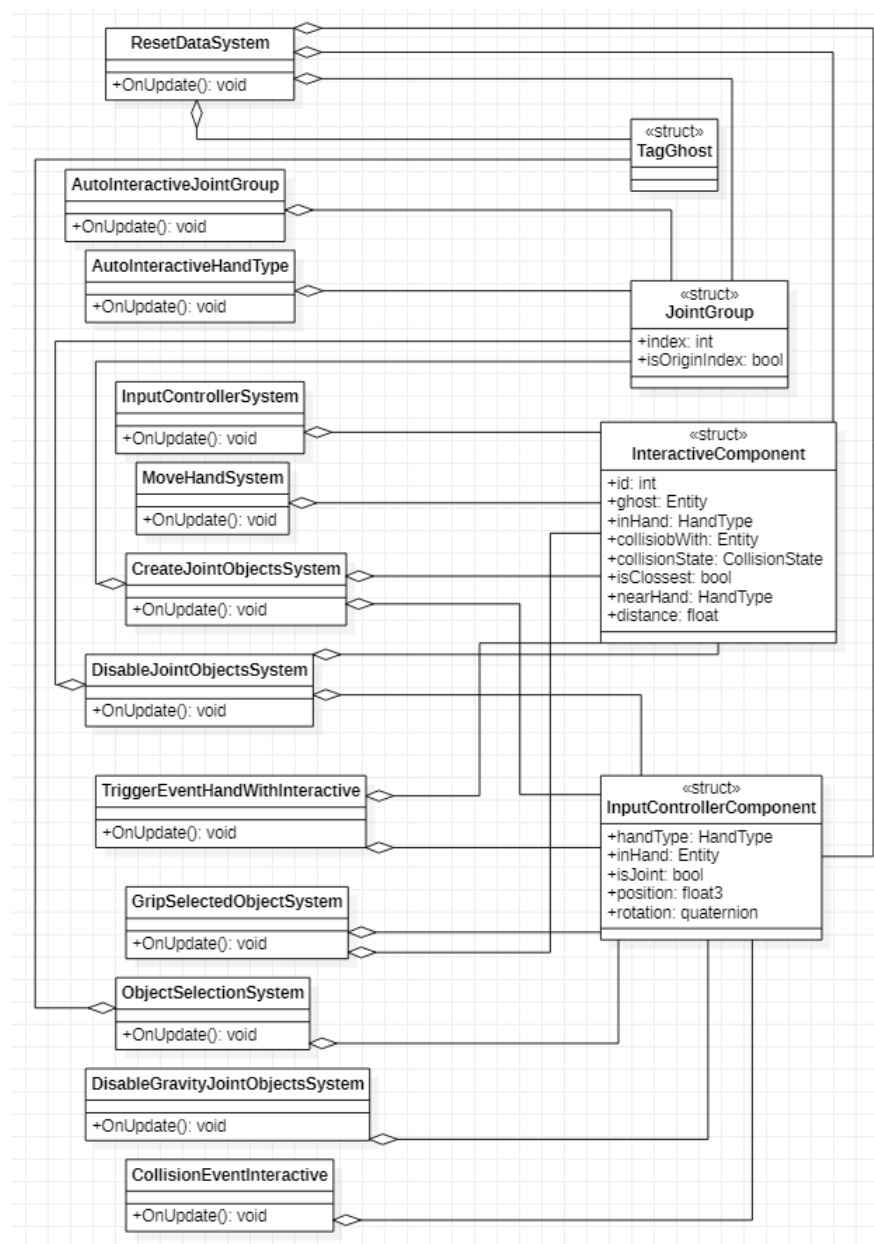


Рисунок 5 – Диаграмма классов игровой механики

В таблице 3 представлено описание компонентов для работы игровой механики соединения/ разъединения объектов.

Таблица 3 – Описание компонентов игровой механики.

Название	Описание
InputControllerComponent	Хранит данные о контролере
InteractiveComponent	Хранит данные о интерактивных объектах
JointGroup	Хранит данные индекса группы интерактивных объектов
TagGhost	Определяет выделенный объект интерактивного предмета

В таблице 4 представлено описание систем для работы игровой механики соединения/ разъединения объектов.

Таблица 4 – Описание систем игровой механики.

Название	Группа	Описание
AutoInteractiveJointGroup	InitializationSystemGroup	Смена индексов групп для интерактивных объектов
AutoInteractiveHandType	InitializationSystemGroup	Смена рук для интерактивных объектов
ResetDataSystem	InitializationSystemGroup	Сброс данных в начале каждого кадра
InputControllerSystem	InitializationSystemGroup	Регистрация ввода контролеров
MoveHandSystem	FixedStepSimulationSystemGroup	Движения рук
TriggerEventHandWithInteractive	FixedStepSimulationSystemGroup	Обработка касания рук и интерактивных объектов
CollisionEventInteractive	FixedStepSimulationSystemGroup	Обработка касания интерактивных объектов между собой
ObjectSelectionSystem	SimulationSystemGroup	Выделение объекта который можно схватить.
GripSelectedObjectSystem	SimulationSystemGroup	Обработка нажатия кнопки grip на выделенный объект
DisableJointObjectsSystem	SimulationSystemGroup	Разъединить соединённые объекты
CreateJointObjectsSystem	SimulationSystemGroup	Соединить объекты (руку с объектом, объект с объектом)

DisableGravityJointObjectsSystem	FixedStepSimulationSystemGroup	Отключение гравитации интерактивных объектов в руках
----------------------------------	--------------------------------	--

2.3 Проектирование системы эвристического отбора

На рисунке 6 представлена диаграмма классов системы эвристического прямого и косвенного отбора с фильтрами.

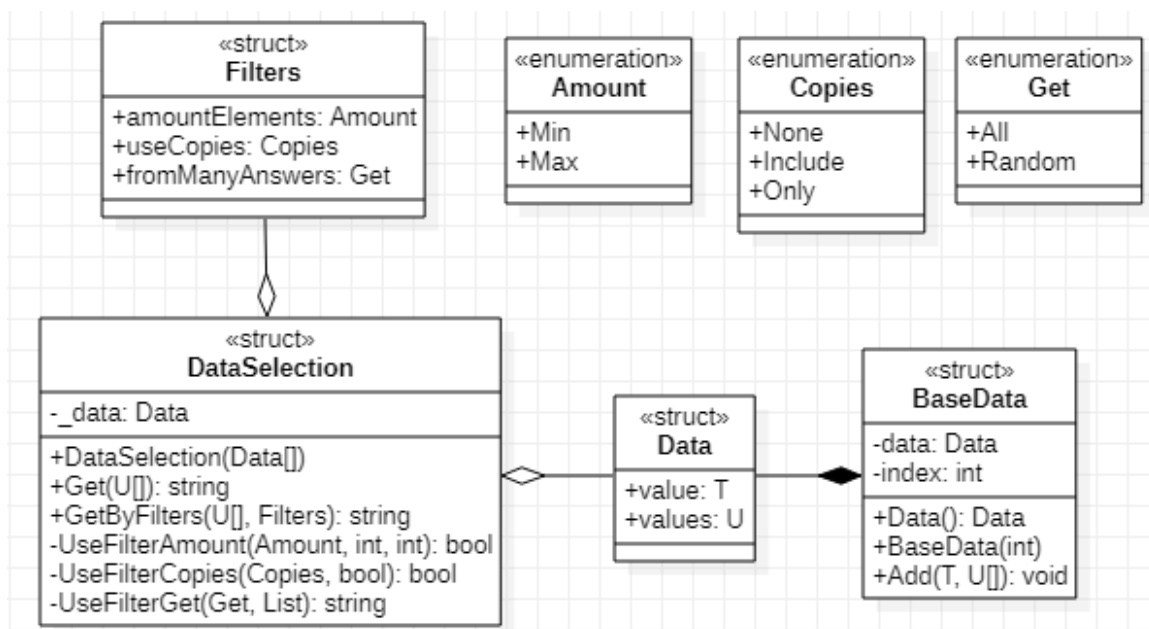


Рисунок 6 – Диаграмма классов системы эвристического отбора

В таблице 5 представлено описание классов для работы системы эвристического отбора.

Таблица 5 – Описание классов системы эвристического отбора.

Название	Описание
Filters	Хранит значения фильтров для косвенного отбора
Amount	Фильтр по количеству элементов в выходном наборе
Copies	Фильтр по использованию копий элементов в выходном наборе
Get	Фильтр для вывода выходного набора
Data	Описывает экземпляр элемента исходного набора данных
BaseData	Хранит исходные данные
DataSelection	Обработка исходных данных системы эвристического отбора

3 Реализация ПО

3.1 Реализация игровой механики соединения/разъединения объектов

На рисунке 7 представлен код скрипта компонента InputControllerComponent.

```
[GenerateAuthoringComponent]
11 usages Руслан Рогозин
public struct InputControllerComponent : IComponentData
{
    public HandType handType;
    public Entity inHand;
    public bool isJoint;
    [HideInInspector] public float3 position;
    [HideInInspector] public quaternion rotation;
    [HideInInspector] public int trackingState;
    [HideInInspector] public float select;
    [HideInInspector] public float selectValue;
    [HideInInspector] public float activate;
    [HideInInspector] public float activateValue;
    [HideInInspector] public float uiPress;
    [HideInInspector] public float uiPressValue;
    [HideInInspector] public float3 hapticDevice;
    [HideInInspector] public float teleportSelect;
    [HideInInspector] public float2 teleportModeActivate;
    [HideInInspector] public float teleportModeCancel;
    [HideInInspector] public float2 turn;
    [HideInInspector] public float2 move;
    [HideInInspector] public float2 rotateAnchor;
    [HideInInspector] public float2 translateAnchor;
    3 usages Руслан Рогозин
    public bool IsGripPressed => selectValue > 0.9f;
    Руслан Рогозин
    public bool IsTriggerPressed => activateValue > 0.9f;
}
```

Рисунок 7 – Код скрипта компонента InputControllerComponent

На рисунке 8 представлен код скрипта компонента InteractiveComponent.

```
[GenerateAuthoringComponent]
23 usages Руслан Рогозин
public struct InteractiveComponent : IComponentData
{
    public int id;
    public Entity ghost;
    public HandType inHand;
    public Entity CollisionWith;
    public CollisionState CollisionState;
    public bool isClosest;
    public HandType nearHand;
    public float distance;
}
```

Рисунок 8 – Код скрипта компонента InteractiveComponent

На рисунке 9 представлен код скрипта компонента JointGroup.

```
9 usages Руслан Рогозин
public struct JointGroup : IComponentData
{
    public int index;
    public bool isOriginIndex;
}
```

Рисунок 9 – Код скрипта компонента JointGroup

На рисунке 10 представлен код скрипта компонента тега TagGhost. Так как это тег, то он имеет пустую реализацию и нужен лишь для того, чтобы отделить интерактивный объект от точно такого же объекта, но который нужен лишь для выделения объекта.

```
[GenerateAuthoringComponent]
2 usages Руслан Рогозин More
public struct TagGhost : IComponentData
{
}
```

Рисунок 10 – Код скрипта компонента TagGhost

На рисунке 11 представлен код скрипта системы AutoInteractiveJointGroup.

```

[UpdateInGroup(typeof(InitializationSystemGroup))]
[UpdateBefore(typeof(ResetDataSystem))]
Unity ECS 1 usage Руслан Рогозин *
public partial class AutoInteractiveJointGroup : SystemBase
{
    Руслан Рогозин *
    protected override void OnUpdate()
    {
        var jointGroup = GetComponentDataFromEntity<JointGroup>();
        Entities.ForEach((Entity entity, ref JointGroup joint) =>
        {
            if (joint.isOriginIndex)
            {
                joint.index = entity.Index;
                joint.isOriginIndex = false;
            }
        }).Schedule();
        Entities.ForEach((ref PhysicsConstrainedBodyPair bodyPair) =>
        {
            //JointIndex for interactive with interactive
            if (jointGroup.HasComponent(bodyPair.EntityA) &&
                jointGroup.HasComponent(bodyPair.EntityB))
            {
                var jointA = jointGroup[bodyPair.EntityA];
                var jointB = jointGroup[bodyPair.EntityB];
                if (jointA.index == 0 && jointB.index == 0) return;
                if (jointA.index > jointB.index)
                    jointB.index = jointA.index;
                else if (jointB.index > jointA.index)
                    jointA.index = jointB.index;
                jointGroup[bodyPair.EntityA] = jointA;
                jointGroup[bodyPair.EntityB] = jointB;
            }
        }).Schedule();
    }
}

```

Рисунок 11 – Код скрипта системы AutoInteractiveJointGroup

На рисунке 12 представлен код скрипта системы AutoInteractiveHandType.

```

[UpdateInGroup(typeof(InitializationSystemGroup))]
[UpdateAfter(typeof(AutoInteractiveJointGroup))]
[UpdateBefore(typeof(ResetDataSystem))]
Unity ECS Руслан Рогозин *
public partial class AutoInteractiveHandType : SystemBase
{
    Руслан Рогозин *
    protected override void OnUpdate()
    {
        var jointGroup = GetComponentDataFromEntity<JointGroup>();
        NativeArray<int> hand = new NativeArray<int>(length:2, Allocator.TempJob);
        var createHandJointInteractive = Entities.ForEach(
            (Entity entity, in InputControllerComponent input) =>
            {
                if (input.inHand == Entity.Null) return;
                var index = (int)input.handType - 1;
                hand[index] = jointGroup[input.inHand].index;
            }).Schedule(Dependency);
        Entities.ForEach((Entity entity,
            ref InteractiveComponent interactive, in JointGroup jointGroup) =>
        {
            interactive.inHand = HandType.None;
            if (jointGroup.index == hand[0] && hand[0] != 0)
                interactive.inHand = HandType.Left;
            else if (jointGroup.index == hand[1] && hand[1] != 0)
                interactive.inHand = HandType.Right;
            else if (hand[0] == hand[1] && hand[0] != 0 && hand[1] != 0 &&
                (jointGroup.index == hand[0] || jointGroup.index == hand[1]))
                interactive.inHand = HandType.Both;
        }).Schedule(createHandJointInteractive).Complete();
        hand.Dispose();
    }
}

```

Рисунок 12 – Код скрипта системы AutoInteractiveHandType

На рисунке 13 представлен код скрипта системы ResetDataSystem.

```

[UpdateInGroup(typeof(InitializationSystemGroup))]
Unity ECS 2 usages  Руслан Рагзин +1
public partial class ResetDataSystem : SystemBase
{
    private EndInitializationEntityCommandBufferSystem
        _endInitializationEntityCommandBufferSystem;

    Руслан Рагзин
    protected override void OnCreate()
    {
        _endInitializationEntityCommandBufferSystem =
            World.GetOrCreateSystem<EndInitializationEntityCommandBufferSystem>();
    }

    Руслан Рагзин
    protected override void OnStartRunning()
    {
        var cbs = _endInitializationEntityCommandBufferSystem.CreateCommandBuffer();
        Entities.WithAll<InteractiveComponent>().WithNone<JointGroup>().ForEach(
            (Entity entity) =>
            {
                cbs.AddComponent(entity, component: new JointGroup {index = 0});
            }).WithoutBurst().Run();
    }

    Руслан Рагзин +1
    protected override void OnUpdate()
    {
        Entities.ForEach((ref InputControllerComponent input) =>
        {
            if (!input.IsGripPressed)
            {
                input.inHand = Entity.Null;
                input.isJoint = false;
            }
        }).Schedule();

        //Создаем дочернюю систему
        Entities.ForEach(
            (ref InteractiveComponent interactive) =>
            {
                interactive.isLosest = false;
                interactive.nearHand = HandType.None;
                interactive.distance = 0f;
            }).Schedule();

        //Создаем дочернюю систему
        Entities.WithAll<TagGhost>().ForEach(
            (ref Translation translation) => { translation.Value = float3.zero; }).Schedule();
    }
}

```

Рисунок 13 – Код скрипта системы ResetDataSystem

На рисунке 14 представлен код скрипта системы InputControllerSystem.

```

[UpdateInGroup(typeof(InitializationSystemGroup))]
Unity ECS  Руслан Рагзин
public partial class InputControllerSystem : SystemBase
{
    Руслан Рагзин
    protected override void OnUpdate()
    {
        var hands = new NativeArray<Input>(length: 2, Allocator.TempJob);
        hands[0] = GameController.Hands.input[0];
        hands[1] = GameController.Hands.input[1];
        JobHandle inputLeftHandJob = Entities.ForEach(
            (ref InputControllerComponent input) =>
            {
                if(input.handType == HandType.None)
                    return;
                var i = (int) input.handType - 1;
                input.position = hands[i].position;
                input.rotation = (quaternion)hands[i].rotation;
                input.trackingState = hands[i].trackingState;
                input.select = hands[i].select;
                input.selectValue = hands[i].selectValue;
                input.activate = hands[i].activate;
                input.activateValue = hands[i].activateValue;
                input.uiPress = hands[i].uiPress;
                input.uiPressValue = hands[i].uiPressValue;
                input.hapticDevice = hands[i].hapticDevice;
                input.teleportSelect = hands[i].teleportSelect;
                input.teleportModeActivate = hands[i].teleportModeActivate;
                input.teleportModeCancel = hands[i].teleportModeCancel;
                input.turn = hands[i].turn;
                input.move = hands[i].move;
                input.rotateAnchor = hands[i].rotateAnchor;
                input.translateAnchor = hands[i].translateAnchor;
            }).Schedule(Dependency);
        inputLeftHandJob.Complete();
        hands.Dispose();
    }
}

```

Рисунок 14 – Код скрипта системы InputControllerSystem

На рисунке 15 представлен код скрипта системы MoveHandSystem.

```

[UpdateInGroup(typeof(FixedStepSimulationSystemGroup))]
[UpdateBefore(typeof(TriggerEventHandWithInteractive))]
Unity ECS  Руслан Рогозин
public partial class MoveHandSystem : SystemBase
{
    Руслан Рогозин
    protected override void OnUpdate()
    {
        var deltaTime = Time.DeltaTime;
        Entities.ForEach(
            (ref PhysicsVelocity velocity, ref PhysicsMass mass, in Translation transform,
             in Rotation rotation, in InputControllerComponent input) =>
            {
                velocity.Linear = (input.position - transform.Value) / deltaTime;
                velocity.Angular =
                    math.mul(a: math.inverse(rotation.Value), b: input.rotation).value.xyz / deltaTime;
                mass.InverseInertia = new float3(x: 1, y: 1, z: 1);
            }).Schedule(Dependency).Complete();
    }
}

```

Рисунок 15 – Код скрипта системы MoveHandSystem

На рисунке 16 представлен код скрипта системы TriggerEventHandWithInteractive.

```

[UpdateInGroup(typeof(FixedStepSimulationSystemGroup))]
[UpdateAfter(typeof(MoveHandSystem))]
Unity ECS  2 usages  Руслан Рогозин *
public class TriggerEventHandWithInteractive : SystemBase
{
    private BuildPhysicsWorld _buildPhysicsWorld;
    private StepPhysicsWorld _stepPhysicsWorld;
    Руслан Рогозин
    protected override void OnCreate()
    {
        _buildPhysicsWorld = World.GetOrCreateSystem<BuildPhysicsWorld>();
        _stepPhysicsWorld = World.GetOrCreateSystem<StepPhysicsWorld>();
    }
    1 usage  Руслан Рогозин *
    private struct ApplicationJob : ITriggerEventsJob
    {
        [ReadOnly] public ComponentDataFromEntity<InputControllerComponent> handGroup;
        [ReadOnly] public ComponentDataFromEntity<Translation> translate;
        public ComponentDataFromEntity<Interactive> interactiveGroup;
        Руслан Рогозин *
        public void Execute(TriggerEvent triggerEvent)
        {
            Entity entityHand = Entity.Null;
            Entity entityInHand = Entity.Null;
            if (handGroup.HasComponent(triggerEvent.EntityA)
                && interactiveGroup.HasComponent(triggerEvent.EntityB))
            {
                entityHand = triggerEvent.EntityA;
                entityInHand = triggerEvent.EntityB;
            }
            else if (handGroup.HasComponent(triggerEvent.EntityB)
                && interactiveGroup.HasComponent(triggerEvent.EntityA))
            {
                entityHand = triggerEvent.EntityB;
                entityInHand = triggerEvent.EntityA;
            }
            else
            {
                return;
            }
            var interactive = interactiveGroup[entityInHand];
            interactive.nearHand = handGroup[entityHand].handType;
            interactive.distance = math.distance(x: translate[entityHand].Value,
                translate[entityInHand].Value);
            interactiveGroup[entityInHand] = interactive;
        }
    }
    Руслан Рогозин
    protected override void OnUpdate()
    {
        var applicationJob = new ApplicationJob
        {
            handGroup = GetComponentDataFromEntity<InputControllerComponent>(),
            interactiveGroup = GetComponentDataFromEntity<Interactive>(),
            translate = GetComponentDataFromEntity<Translation>(),
        };
        applicationJob.Schedule(_stepPhysicsWorld.Simulation,
            ref _buildPhysicsWorld.PhysicsWorld, Dependency).Complete();
    }
}

```

Рисунок 16 – Код скрипта системы TriggerEventHandWithInteractive

На рисунке 17 представлен код скрипта системы CollisionEventInteractive.

```
[UpdateInGroup(typeof(FixedStepSimulationSystemGroup))]
[UpdateAfter(typeof(TriggerEventHandlerWithInteractive))]
public partial class CollisionEventInteractive : SystemBase
{
    private StepPhysicsWorld _stepPhysicsWorld;
    private EndFixedStepSimulationEntityCommandBufferSystem _commandBufferSystem;
    protected override void OnCreate()
    {
        _stepPhysicsWorld = World.GetOrCreateSystem<StepPhysicsWorld>();
        _commandBufferSystem = World.GetOrCreateSystem<EndFixedStepSimulationEntityCommandBufferSystem>();
    }

    private struct CollisionSystem : ICollisionEventsJob
    {
        public ComponentDataFromEntity<InteractiveComponent> interactiveGroup;
        public void Execute(CollisionEvent collision)
        {
            if (!interactiveGroup.HasComponent(collision.EntityA) || !interactiveGroup.HasComponent(collision.EntityB))
                return;
            if (interactiveGroup[collision.EntityA].inHand == HandType.Left &&
                interactiveGroup[collision.EntityB].inHand == HandType.Right ||
                interactiveGroup[collision.EntityB].inHand == HandType.Left &&
                interactiveGroup[collision.EntityA].inHand == HandType.Right)
            {
                var interactiveA = interactiveGroup[collision.EntityA];
                var interactiveB = interactiveGroup[collision.EntityB];
                interactiveA.CollisionWith = collision.EntityB;
                interactiveB.CollisionWith = collision.EntityA;
                interactiveA.CollisionState = CollisionState.Yes;
                interactiveB.CollisionState = CollisionState.Yes;
                interactiveGroup[collision.EntityA] = interactiveA;
                interactiveGroup[collision.EntityB] = interactiveB;
            }
        }
    }

    protected override void OnUpdate()
    {
        var collisionSystem = new CollisionSystem
        {
            interactiveGroup = GetComponentDataFromEntity<InteractiveComponent>()
        };
        Dependency = collisionSystem.Schedule(_stepPhysicsWorld.Simulation, Dependency);
        _commandBufferSystem.AddJobHandleForProducer(Dependency);
    }
}
```

Рисунок 17 – Код скрипта системы CollisionEventInteractive

На рисунке 18 представлен код скрипта системы ObjectSelectionSystem.

```
public class ObjectSelectionSystem : SystemBase
{
    protected override void OnUpdate()
    {
        var minDistanceToObject = new NativeArray<float>(num, 2, Allocator.TempJob);
        var transformObject = new NativeArray<Translation>(num, 2, Allocator.TempJob);
        var rotateObject = new NativeArray<Rotation>(num, 2, Allocator.TempJob);
        var entityGhost = new NativeArray<Entity>(num, 2, Allocator.TempJob);
        var GetMinDistanceJob = Entities.ForEach(
            (Entity entity, ref Interactive interactive, in Translation translation, in Rotation rotate) =>
            {
                if (interactive.nearHand == HandType.None || interactive.withHand == JointState.On)
                    return;
                var index = (int) interactive.nearHand - 1;
                if (minDistanceToObject[index] == 0f ||
                    minDistanceToObject[index] > interactive.distance)
                {
                    minDistanceToObject[index] = interactive.distance;
                    transformObject[index] = translation;
                    rotateObject[index] = rotate;
                    entityGhost[index] = interactive.ghost;
                    entityObject[index] = entity;
                }
            }, WithName("GetMinDistanceJob"))
        .Schedule(Dependency);
        var setLossesForInteractive = Job.WithCode(() =>
        {
            var interactive = GetComponentDataFromEntity<Interactive>();
            foreach (var entity in entityObject)
            {
                if (entity != Entity.Null)
                {
                    var entityInteractive = interactive[entity];
                    entityInteractive.setLosses = true;
                    interactive[entity] = entityInteractive;
                }
            }
        }, WithName("setLossesForInteractive"), Schedule(GetMinDistanceJob));
        var selectedObject = Entities.WithAll<TagGhost>().ForEach(
            (Entity entity, ref Translation translation, ref Rotation rotate, ref NonUniformScale scale) =>
            {
                for (int i = 0; i < entityGhost.Length; i++)
                {
                    if (entity.Index == entityGhost[i].Index &&
                        entity.Version == entityGhost[i].Version)
                    {
                        translation = transformObject[i];
                        rotate = rotateObject[i];
                    }
                }
            }, WithName("selectedObject"), Schedule(setLossesForInteractive));
        selectedObject.Complete();
        minDistanceToObject.Dispose();
        transformObject.Dispose();
        rotateObject.Dispose();
        entityGhost.Dispose();
        entityObject.Dispose();
    }
}
```

Рисунок 18 – Код скрипта системы ObjectSelectionSystem

На рисунке 19 представлен код скрипта системы GripSelectedObjectSystem.

```
[UpdateInGroup(typeof(SimulationSystemGroup))]  
[UpdateAfter(typeof(ObjectSelectionSystem))]  
# Unity ECS 1 usage # Руслан Рогозин  
public class GripSelectedObjectSystem : SystemBase  
{  
    private EndSimulationEntityCommandBufferSystem _endSimulationEntityCommandBufferSystem;  
  
    # Руслан Рогозин  
    protected override void OnCreate()  
    {  
        _endSimulationEntityCommandBufferSystem =  
            World.GetOrCreateSystem<EndSimulationEntityCommandBufferSystem>();  
    }  
  
    # Руслан Рогозин  
    protected override void OnUpdate()  
    {  
        NativeArray<Entity> entityHands = new NativeArray<Entity>(length: 2, Allocator.TempJob);  
        NativeArray<Entity> entityObjects = new NativeArray<Entity>(length: 2, Allocator.TempJob);  
        var cbs = _endSimulationEntityCommandBufferSystem.CreateCommandBuffer()  
            .AsParallelUpdater();  
  
        var grabSelectedObjectJob = Entities.ForEach(  
            (Entity entity, int entityInQueryIndex, ref Interactive interactive) =>  
            {  
                if (interactive.nearHand == HandType.None ||  
                    !interactive.isClosest)  
                    return;  
  
                var index = (int) interactive.nearHand - 1;  
                entityObjects[index] = entity;  
                }.Schedule(Dependency);  
  
        var activeHandJob = Entities.ForEach(  
            (Entity entity, ref InputControllerComponent input) =>  
            {  
                if (input.inHand != Entity.Null || !input.isGripPressed)  
                    return;  
  
                var index = (int) input.handType - 1;  
                input.inHand = entityObjects[index];  
                }.Schedule(grabSelectedObjectJob);  
        activeHandJob.Complete();  
        entityHands.Dispose();  
        entityObjects.Dispose();  
    }  
}
```

Рисунок 19 – Код скрипта системы GripSelectedObjectSystem

На рисунке 20 представлен код скрипта системы DisableJointObjects.

```
[UpdateInGroup(typeof(SimulationSystemGroup))]  
[UpdateAfter(typeof(GripSelectedObjectSystem))]  
# Unity ECS 1 usage # Руслан Рогозин  
public partial class DisableJointObjects : SystemBase  
{  
    private EndSimulationEntityCommandBufferSystem _endSimulationEntityCommandBufferSystem;  
  
    # Руслан Рогозин  
    protected override void OnCreate()  
    {  
        _endSimulationEntityCommandBufferSystem =  
            World.GetOrCreateSystem<EndSimulationEntityCommandBufferSystem>();  
    }  
  
    # Руслан Рогозин  
    protected override void OnUpdate()  
    {  
        var handInput = GetComponentDataFromEntity<InputControllerComponent>();  
        var interactiveGroup = GetComponentDataFromEntity<InteractiveComponent>();  
        var jointGroup = GetComponentDataFromEntity<JointGroup>();  
        var cbs = _endSimulationEntityCommandBufferSystem.CreateCommandBuffer();  
        var jointGroupQuery = GetEntityQuery(typeof(InteractiveComponent))  
            .ToArray(Allocator.TempJob);  
        var entityHand = GetEntityQuery(typeof(InputControllerComponent))  
            .ToComponentDataArray<InputControllerComponent>(Allocator.TempJob);  
  
        Entities.ForEach((Entity entity, ref PhysicsConstrainedBodyPair bodyPair) =>  
        {  
            // Hand -> object  
            if (handInput.HasComponent(bodyPair.EntityA))  
            {  
                if (!handInput[bodyPair.EntityA].isGripPressed)  
                    cbs.DestroyEntity(entity);  
                return;  
            }  
  
            // Object -> object  
            if (interactiveGroup.HasComponent(bodyPair.EntityA)  
                && interactiveGroup.HasComponent(bodyPair.EntityB))  
            {  
                if (entityHand[0].inHand == bodyPair.EntityA &&  
                    entityHand[1].inHand == bodyPair.EntityB ||  
                    entityHand[0].inHand == bodyPair.EntityB &&  
                    entityHand[1].inHand == bodyPair.EntityA)  
                {  
                    for (var i = 0; i < jointGroupQuery.Length; i++)  
                    {  
                        var joint = jointGroup[jointGroupQuery[i]];  
                        if (joint.index != 0)  
                            joint.isOriginIndex = true;  
                        jointGroup[jointGroupQuery[i]] = joint;  
                    }  
                    cbs.DestroyEntity(entity);  
                }  
            }  
        }).Schedule(Dependency).Complete();  
        jointGroupQuery.Dispose();  
        entityHand.Dispose();  
    }  
}
```

Рисунок 20 – Код скрипта системы DisableJointObjects

На рисунке 21 представлен код скрипта системы CreateJointObjects.

```
public class CreateJointObjects : SystemBase
{
    private EndSimulationEntityCommandBufferSystem _endSimulationEntityCommandBufferSystem;
    private BuildPhysicsWorld _buildPhysicsWorld;
    private PhysicsManager
    protected override void OnCreate()
    {
        _endSimulationEntityCommandBufferSystem = World.GetOrCreateSystem<EndSimulationEntityCommandBufferSystem>();
        _buildPhysicsWorld = World.GetOrCreateSystem<BuildPhysicsWorld>();
    }
    private PhysicsManager
    protected override void OnUpdate()
    {
        var interactiveGroup = GetComponentDataFromEntity<Interactive>();
        var cbs = _endSimulationEntityCommandBufferSystem.CreateCommandBuffer();
        var physicsWorld = _buildPhysicsWorld.PhysicsWorld;
        NativeArray<Entity> entityInteractive = new NativeArray<Entity>(Mathf.Ceil(Allocator.Temp32));
        //Find <- Interactive
        var createJointInteractive = Entities.WithHeadOnly<physicsWorld>.ForEach<(Entity entity, ref InputControllerComponent input) =>
        {
            if(input.inHand == Entity.Null || input.isJoint)
                return;
            input.isJoint = true;
            var interactive = interactiveGroup[input.inHand];
            interactive.withHand = JointState.On;
            interactiveGroup[input.inHand] = interactive;
            var rigidTransformHand = physicsWorld.Bodies[physicsWorld.GetRigidBodyIndex(entity)].WorldFromBody;
            var rigidTransform = physicsWorld.Bodies[physicsWorld.GetRigidBodyIndex(input.inHand)].WorldFromBody;
            var bodyFrame = new BodyFrame(new RigidTransform());
            RigidTransform bFrame = math.mul<math>(math.Inverse(rigidTransformObject), @rigidTransformHand);
            var bodyFrame = new BodyFrame(new RigidTransform()) { pos = bFrame.pos, rot = bFrame.rot };
            var entityHands = cbs.CreateEntity();
            cbs.AddComponent<entityHands>, @component new PhysicsConstrainedBodyPair(@input.entity, @input.inHand, @component true);
            cbs.AddComponent<entityHands>, @component new PhysicsJoint.CrateFixed(bodyFrame, bodyFrame);
        });
        //ScheduleDependency()
        //Interactive <- Interactive
        var getInteractiveForJoint = Entities.ForEach<(Entity entity, ref Interactive interactive) =>
        {
            if(interactive.withInteractive != JointState.InProgress)
                return;
            if (interactive.withHand == JointState.On) entity.interactive[0] = entity;
            else if (interactive.withHand == JointState.Off) entity.interactive[1] = entity;
            interactive.withInteractive = JointState.Off;
        });
        //Schedule<CreateJointInteractive>
        job.WithHeadOnly<physicsWorld>.WithCode<() =>
        {
            if(entityInteractive[0] == Entity.Null || entityInteractive[1] == Entity.Null)
                return;
            var rigidTransformHand = physicsWorld.Bodies[physicsWorld.GetRigidBodyIndex(entityInteractive[0])].WorldFromBody;
            var rigidTransformObject = physicsWorld.Bodies[physicsWorld.GetRigidBodyIndex(entityInteractive[1])].WorldFromBody;
            var bodyFrame = new BodyFrame(new RigidTransform());
            RigidTransform bFrame = math.mul<math>(math.Inverse(rigidTransformObject), @rigidTransformHand);
            var bodyFrame = new BodyFrame(new RigidTransform()) { pos = bFrame.pos, rot = bFrame.rot };
            var entityHands = cbs.CreateEntity();
            cbs.AddComponent<entityHands>, @component new PhysicsConstrainedBodyPair(@input.entityInteractive[0], @input.entityInteractive[1], @component true);
            cbs.AddComponent<entityHands>, @component new PhysicsJoint.CrateFixed(bodyFrame, bodyFrame);
        });
        entityInteractive.Dispose();
    }
}
```

Рисунок 21 – Код скрипта системы CreateJointObjects

На рисунке 22 представлен код скрипта системы DisableGravityJointObjects.

```
[UpdateInGroup(typeof(FixedStepSimulationSystemGroup))]
@ Unity ECS @ Pygame Pygame
public class DisableGravityJointObjects : ComponentSystem
{
    private EndFixedStepSimulationEntityCommandBufferSystem
    _endFixedStepSimulationEntityCommandBufferSystem;
    private PhysicsManager
    protected override void OnStartRunning()
    {
        _endFixedStepSimulationEntityCommandBufferSystem =
        World.GetOrCreateSystem<EndFixedStepSimulationEntityCommandBufferSystem>();
        var cbs = _endFixedStepSimulationEntityCommandBufferSystem.CreateCommandBuffer();
        Entities
        .WithAll<InteractiveComponent>()
        .WithNone<PhysicsGravityFactor>()
        .ForEach<(entity) => { cbs.AddComponent(entity, @component new PhysicsGravityFactor()); };
        Entities.ForEach<(Entity entity, ref PhysicsMass mass) =>
        {
            cbs.AddComponent(entity,
            @component new DefaultInverseInertia() { value = mass.InverseInertia });
        });
    }
    private PhysicsManager
    protected override void OnUpdate()
    {
        Entities
        .WithAll<PhysicsConstrainedBodyPair>()
        .WithNone<PhysicsWorldIndex>()
        .ForEach<(entity) =>
        {
            EntityManager.AddSharedComponentData(entity, @component new PhysicsWorldIndex());
        });
        Entities.ForEach<
        (ref PhysicsGravityFactor gravity,
        ref InteractiveComponent interactive, ref PhysicsMass mass,
        ref DefaultInverseInertia defaultInverseInertia) =>
        {
            if (interactive.inHand == HandType.None)
            {
                gravity.Value = 1;
                mass.InverseInertia = defaultInverseInertia.value;
            }
            else
            {
                gravity.Value = 0;
                mass.InverseInertia = new float(0.1, 0.1, 0.1);
            }
        });
    }
}
```

Рисунок 22 – Код скрипта системы DisableGravityJointObjects

3.2 Реализация системы эвристического отбора

Первым делом для системы эвристического отбора понадобится данные из которых и будет совершать отбор. Для хранения данных создадим структуру `Data`, как показано на рисунке 23. Структура `Data` обладает обобщёнными типами `T` и `U`, что позволяют создавать данные любого типа. Данная структура имеет поле `value` – название данных и поле `values`, которое хранит сами данные.

```
public struct Data<T, U>
{
    public T value;
    public U[] values;
}
```

Рисунок 23 – Структура данных `Data`

Структура `Data` описывает один экземпляр данных, для описания множеств данных создадим структуру `BaseData`, которая хранит массив структур `Data` и дает возможность заполнять данные с помощью метода `Add` и получать данные с помощью свойства `ResultBaseData`, как показано на рисунке 24. Структура `BaseData` имеет все необходимое для создания множеств данных

```

public struct BaseData<T, U>
{
    private Data<T, U>[] data;
    private int index;

    [2 usages] [new *]
    public Data<T, U>[] ResultBaseData => data;

    [1 usage] [new *]
    public BaseData(int length)
    {
        data = new Data<T, U>[length];
        index = 0;
    }

    [1 usage] [new *]
    public void Add(T value, U[] values)
    {
        data[index].value = value;
        data[index].values = values;
    }
}

```

Рисунок 24 – Структура данных BaseData

Реализуем возможность совершать отбор данных. Для этого создадим фильтр отбора данных, как показано на рисунке 25.

```

public struct Filters
{
    public Amount amountElements;
    public Copies useCopies;
    public Get fromManyAnswers;
}

```

Рисунок 25 – Структура данных Filters

Каждый элемент фильтра описывается типом enum. На рисунке 26 показан фильтр enum Amount.

```

public enum Amount
{
    Min,
    Max
}

```

Рисунок 26 – Enum Amount

На рисунке 27 показан фильтр enum Copies.

```
public enum Copies
{
    None,
    Include,
    Only
}
```

Рисунок 27 – Enum Copies

На рисунке 28 показан фильтр enum Get.

```
public enum Get
{
    All,
    Random
}
```

Рисунок 28 – Enum Get

На рисунке 29 показан метод прямого отбора данных.

```
public string Get(U[] values)
{
    for (var i = 0; i < _data.Length; i++)
    {
        var interactives = new Dictionary<U, byte>();

        for (var j = 0; j < _data[i].values.Length; j++)
            foreach (var value in values)
                if (_data[i].values[j].ToString() == value.ToString())
                {
                    if (!interactives.ContainsKey(value)) interactives.Add(value, 1);
                    else interactives[value] = (byte)(interactives[value] + 1);
                    break;
                }

        if (interactives.Count == values.Length)
            return _data[i].value.ToString();
    }
    return "Нет данных";
}
```

Рисунок 29 – Метод прямого отбора данных

На рисунке 30 показан метод косвенного отбора данных.

```

public string GetByFilters(NativeList<byte> inputData, Filters filters)
{
    var rooms = new Dictionary<byte, bool>();
    ref var value = ref _data.Value.data;
    for (var i = 0; i < value.Length; i++)
    {
        var interactives = new Dictionary<byte, byte>();
        ref var interactive = ref value[i].interactive;
        var numberCoincidences = 0;
        for (var j = 0; j < interactive.Length; j++)
            foreach (var data in inputData)
                if (interactive[j] == data)
                {
                    if (!interactives.ContainsKey(data)) interactives.Add(data, 1);
                    else interactives[data] = (byte)(interactives[data] + 1);
                    break;
                }
        for (var m = 0; m < interactives.Count; m++)
            numberCoincidences += interactives[interactives.Keys.ElementAt(m)];
        if (numberCoincidences == interactives.Count && interactives.Count != 0 &&
            inputData.Length == interactives.Count)
            rooms.Add((byte)i, false);
        else if (numberCoincidences > interactives.Count && interactives.Count != 0 &&
            inputData.Length == interactives.Count)
            rooms.Add((byte)i, true);
    }
    var count = (filters.amountElements == Amount.Min) ? 100 : 0;
    var listResult = new List<string>();
    foreach (var index in rooms.Keys)
    {
        ref var interactive = ref value[index].interactive;
        if (UseFilterAmount(filters.amountElements, currentCount: count, nextCount: interactive.Length))
        {
            count = interactive.Length;
            listResult.Clear();
        }
        if (count == interactive.Length)
        {
            if (UseFilterCopies(filters.useCopies, rooms[index]))
                listResult.Add(rooms.value[index].name.ToString());
        }
    }
    if (listResult.Count > 1)
    {
        return UseFilterGet(filters.fromManyAnswers, listResult);
    }
    return listResult.Count == 1 ? listResult[0] : "Нет данных";
}

```

Рисунок 30 – Метод косвенного отбор данных

На рисунке 31 показана реализация фильтров Amount, Copies и Get.

```

private bool UseFilterAmount(Amount filter, int currentCount, int nextCount)
{
    if (filter == Amount.Min && currentCount > nextCount)
        return true;
    if (filter == Amount.Max && currentCount < nextCount)
        return true;
    return false;
}

[1 usage] new *
private bool UseFilterCopies(Copies filter, bool isHaveCopies)
{
    return filter == Copies.Include ||
        filter == Copies.Only && isHaveCopies ||
        filter == Copies.None && !isHaveCopies;
}

[1 usage] Русская Погода *
private string UseFilterGet<T>(Get filter, List<T> objects)
{
    if (filter == global::Get.Random)
        return objects[new Random().Next(objects.Count)].ToString();
    if (filter == global::Get.All)
    {
        var resultAll = "";
        foreach (var result in objects)
            resultAll += result + "\n";
        return resultAll;
    }
    return "Нет данных";
}

```

Рисунок 31 – Реализация фильтров Amount, Copies и Get

4 Тестирование ПО

4.1 Тестирование игровой механики соединения/разъединения объектов

На рисунке 32 представлена работа выделения объектов при приближении руки к ним, которые в дальнейшем можно будет поднять.

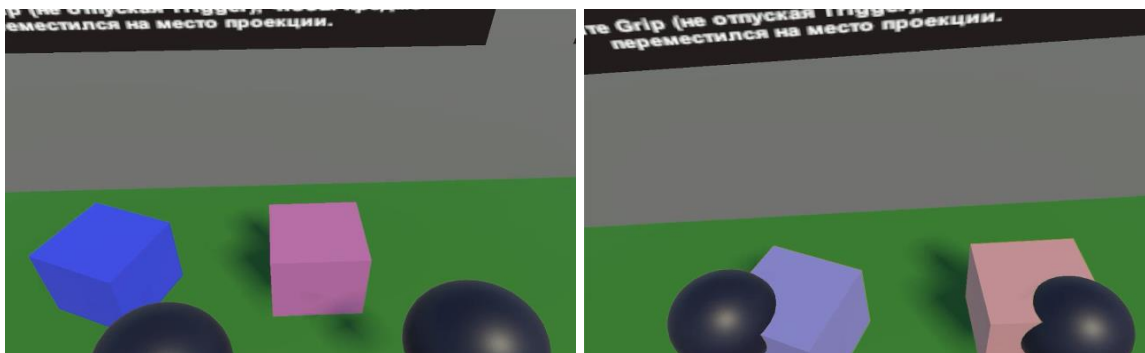


Рисунок 32 – Выделение объектов при приближении руки

На рисунке 33 представлена работа удерживания выделенных объектов в руках.

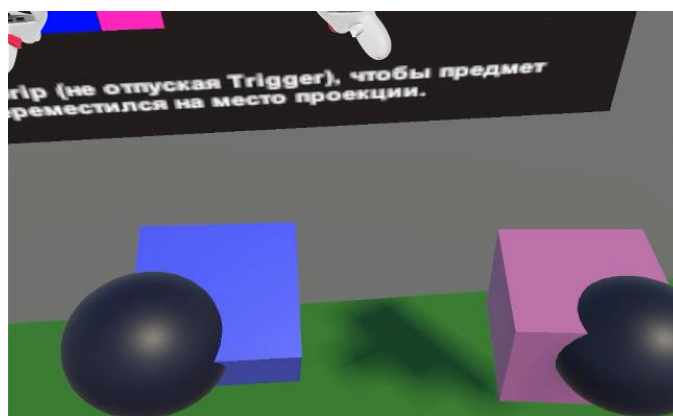


Рисунок 33 – Удерживание выделенных объектов в руках

На рисунке 34 представлена работа соединения объектов между собой.

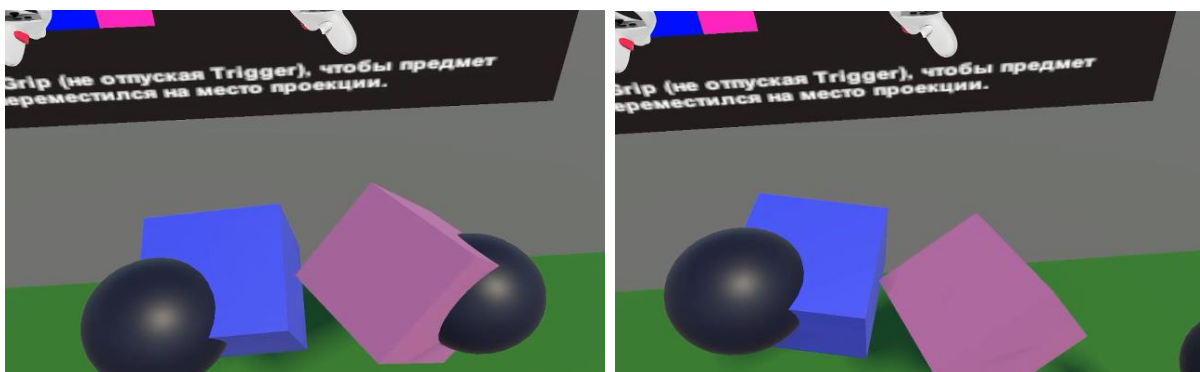


Рисунок 34 – Соединение объектов

4.2 Тестирование системы эвристического отбора

Для тестирования системы эвристического отбора нужна заполнить исходную базу, как показано на рисунке 35.

```
var baseData = new BaseData<string, string>(length: 5);
baseData.Add(value: "Тигр", values: new []{"4 лапы", "2 глаза"});
baseData.Add(value: "Лев", values: new []{"4 лапы", "2 глаза", "грива"});
baseData.Add(value: "Кот", values: new []{"4 лапы", "2 глаза"});
baseData.Add(value: "Слон", values: new []{"4 лапы", "2 глаза", "хобот"});
baseData.Add(value: "Голубь", values: new []{"2 лапы", "2 глаза"});
```

Рисунок 35 – Заполнение исходной базы

Далее, создадим фильтр и введем входные данные в метод Get, как показано на рисунке 36.

```
var filter = new Filters()
{
    amountElements = Amount.Min,
    useCopies = Copies.Include,
    fromManyAnswers = Get.All
};
var result = baseData.Get(values: new[] { "4 лапы", "2 глаза"}, filter);
Debug.Log(result);
```

Рисунок 36 – Создание фильтра и выполнение метода Get

Запустив код выше, получим результат, который показан на рисунке 37.

A screenshot of a Unity console log. It shows a message with a yellow warning icon: "[20:10:32] Тигр Кот UnityEngine.Debug:Log (object)".

Рисунок 37 – Результат тестирования

Теперь, оставим входные данные неизменными, но поменяем фильтр, как показано на рисунке 38.

```
var filter = new Filters()
{
    amountElements = Amount.Max,
    useCopies = Copies.None,
    fromManyAnswers = Get.All
};
var result = baseData.Get(values: new[] { "4 лапы", "2 глаза"}, filter);
Debug.Log(result);
```

Рисунок 38 – Создание нового фильтра и выполнение метода Get

Запустив код выше, получим результат, который показан на рисунке 39.

A screenshot of a Unity console log. It shows a message with a yellow warning icon: "[20:12:16] Лев Слон UnityEngine.Debug:Log (object)".

Рисунок 39 – Результат тестирования

4.3 Тестирование системы эвристического отбора с игровой механикой

Для тестирования системы эвристического отбора с игровой механикой, необходимо заполнить базу, как показано на рисунке 40.

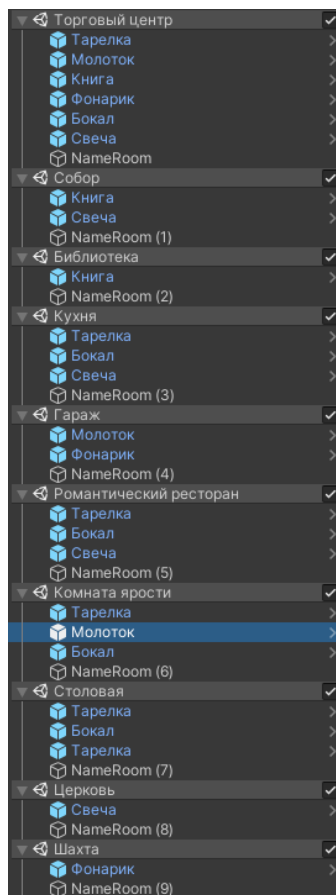


Рисунок 40 – Заполнение базы

На рисунке 41 представлен используемый фильтр для системы эвристического отбора.

```
var filter = new Filters()
{
    amountElements = Amount.Min,
    useCopies = Copies.Include,
    fromManyAnswers = Get.All
};
```

Рисунок 41 – Фильтр системы эвристического отбора

На рисунке 42 представлен результат тестирования с одним предметом в руке.

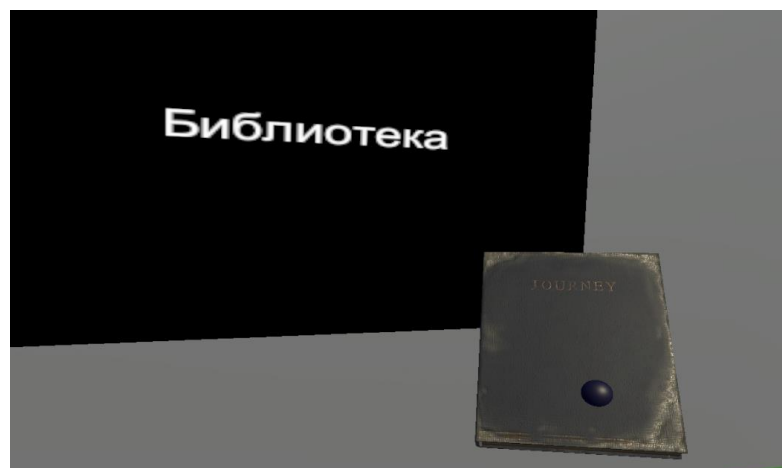


Рисунок 42 – Результат тестирования с одним предметом

На рисунке 43 представлен результат тестирования с тремя предметами в руке.



Рисунок 43 – Результат тестирования с тремя предметами

На рисунке 44 представлен результат тестирования со всеми предметами в руке.

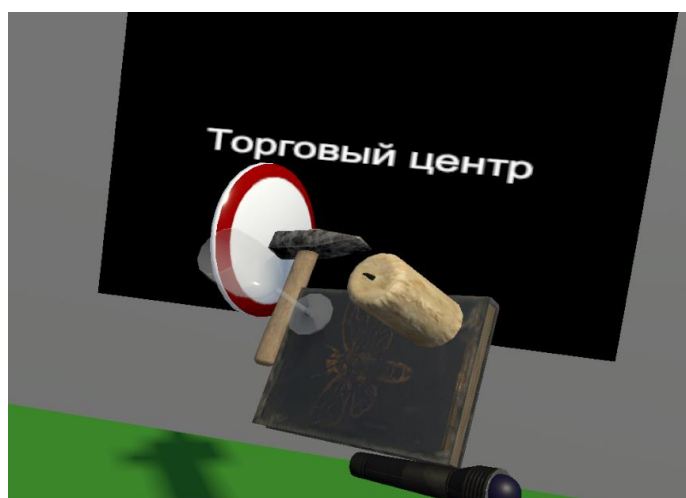


Рисунок 44 – Результат тестирования со всеми предметами

5 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

Целью данной работы является разработка ПО в виртуальной реальности для понимания и развития эвристического мышления. Игра будет представлена на платформе Windows и выполнена при помощи среды разработки игр Unity используя технологию DOTS.

Целью раздела является определение эффективности научно-исследовательского проекта.

Данный раздел, предусматривает рассмотрение следующих задач:

- Предпроектный анализ;
- Определение возможных альтернатив проведения научных исследований;
- Планирование научно-исследовательской работы;
- Определение ресурсной (ресурсосберегающей), финансовой, эффективности исследования.

5.1 Предпроектный анализ

5.1.1 Потенциальные потребители результатов исследования

Технологии шагают вперед и важно развивать в людях стремление к творчеству новыми способами. С этим может помочь разрабатываемая в рамках ВКР ПО, развивающие эвристическое мышление.

Целевая аудитория проекта — люди от 7 до 25 лет, т.к. именно в этом возрасте формируется личность, в котором наиболее важно развивать эвристическое мышление.

Основной причиной, почему был выбран этот возраст, является период формирования личности. В данный период, важно развиваться в творческих деятельности, поскольку именно творческие люди создают произведения искусства мирового масштаба. Разрабатываемое ПО раскрывает и/или развивает творческий потенциал человека эвристическим игровым опытом.

5.1.2 Анализ конкурентных технических решений с позиции ресурсоэффективности и ресурсосбережения

С конкурентно-технической стороны был проведен анализ проекта, который позволяет оценить эффективность будущей разработки и определить направления для ее будущего повышения.

Позиция технического решения и конкурентов оценивается по каждому показателю в пятибалльной шкале, где 5 является сильной позицией, а 1 наиболее слабая. Веса показателей, определяемые экспертным путем, в сумме должны составлять 1.

Анализ конкурентных технических решений определяется по формуле:

$$K = \sum B_i \cdot B_i \quad (1)$$

где K – конкурентоспособность научной разработки или конкурента;

B_i – вес показателя (в долях единицы);

B_i – балл i -го показателя.

В таблице 6 представлена оценочная карта для сравнения конкурентов, развивающих и пропагандирующих схожие области в ИТ. Конкуренты:

- The Legend of Zelda: Breath of the wild,
- Deus Ex,
- The Witness.

Таблица 6 – Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	Вес критерия	Баллы				Конкурентоспособность			
		Б _ф	Б _{к1}	Б _{к2}	Б _{к3}	К _ф	К _{к1}	К _{к2}	К _{к3}
1	2	3	4	5	6	7	8	9	10
Технические критерии оценки ресурсоэффективности									
Энергоэкономичность	0,2	5	3	4	4	1	0,6	0,8	0,8
Удобство эксплуатации	0,15	4	5	4	4	0,6	0,75	0,6	0,6
Быстродействие	0,1	3	3	4	4	0,3	0,3	0,4	0,4
Потребность в ресурсах	0,2	5	2	3	3	1	0,4	0,6	0,6

Экономические критерии оценки эффективности									
Конкурентоспособность продукта	0,1	3	5	3	4	0,3	0,5	0,3	0,4
Послепродажное сопровождение	0,15	4	5	4	3	0,6	0,75	0,6	0,45
Предполагаемый срок эксплуатации	0,1	5	5	5	5	0,5	0,5	0,5	0,5
Итого	1					4,3	3,8	3,8	3,75

Исходя из анализа технических решений и сравнения их с конкурентами, был сделан вывод, что разработка имеет как преимущества, так и недостатки перед конкурентами. Однако проект выигрывает в отношении эффективности к потребляемым ресурсам.

5.1.3 SWOT-анализ

В ходе проведения SWOT анализа были выявлены сильные и слабые стороны проекта, а также его возможности и угрозы. Выявим соответствия сильных и слабых сторон НТИ с помощью построения интерактивных матриц проекта, которая представлена в таблице 7 и в таблице 8.

Таблица 7 – Интерактивная матрица возможностей

	Сильные стороны проекта					
		С1	С2	С3	С4	С5
Возможности	В1	-	-	-	+	-
	В2	+	-	-	-	-
	В3	-	+	-	+	-
	В4	-	+	-	+	-
	В5	+	-	+	-	-
	Слабые стороны проекта					
		Сл1	Сл2	Сл3	Сл4	Сл5
	В1	-	+	-	-	-
	В2	+	-	-	+	-
	В3	-	-	-	-	+
В4	-	-	-	-	-	
В5	+	-	-	+	-	

При анализе интерактивной матрицы возможностей можно выявить следующие коррелирующие возможности: В1С4, В2С1, В3С2С4, В4С2С4, В5С1С3; В1Сл2, В2Сл1Сл4, В3Сл5, В5Сл1.

Таблица 8 – Интерактивная матрица угроз

Угрозы	Сильные стороны проекта					
		C1	C2	C3	C4	C5
	У1	+	-	-	-	-
	У2	-	-	-	-	+
	У3	-	-	-	-	-
	У4	-	-	-	-	-
	У5	-	-	-	-	-
	Слабые стороны проекта					
		Сл1	Сл2	Сл3	Сл4	Сл5
	У1	-	-	+	-	-
	У2	-	-	-	-	+
	У3	+	+	-	-	-
	У4	-	-	-	-	-
	У5	-	-	-	+	-

При анализе интерактивной матрицы угроз можно выявить следующие коррелирующие угрозы: У1С1, У2С5; У1Сл3, У2Сл5, У3Сл1Сл2, У5Сл4.

Результат SWOT-анализа представлен в Таблице 9.

Таблица 9 – Результат SWOT-анализа

	<p>Сильные стороны проекта: С1. Масштабируемость проекта С2. Игра направлена на развитие человеческих способностей, а не на заработок денег С3. Малая стоимость разработки проекта С4. Получение уникального игрового опыта С5. Низкий порог входа в игру</p>	<p>Слабые стороны проекта: Сл1. Необходимость большего человеческого ресурса для конкурентного развития проекта Сл2. Отсутствие пиар-компании проекта Сл3. Отсутствие финансовой поддержки проекта Сл4. Отсутствие специалистов в смежных областях разработки Сл5. Отсутствие игрового разнообразия</p>
<p>Возможности: В1. Малое количество прямых конкурентов на рынке В2. Использование технологии DOTS В3. Большая свобода действия на игровом поле В4. Улучшает понимание значения слова «Эвристика» В5. Использование технологии OpenXR для кроссплатформенности</p>	<p>С4 – В1. Из малого количества конкурентов, больше пользователей смогут получить уникальный игровой опыт. С1 – В2. Технология DOTS делает проект легко масштабируемым. С4 – В4. Объяснение термина «Эвристика» и делает игровой опыт уникальным. С1, С3 – В5. Бесплатная технология OpenXR делает создание VR продуктов под разные платформы дешевле.</p>	<p>Сл2-В1. Ввиду малого количества прямых конкурентов, менее необходима качественная пиар-компания. Сл5 – В3. Свобода действий делает монотонный игровой процесс более продолжительным. Сл1 – В2, Сл1 – В5, Сл4 – В2, Сл4 – В5. Простота использования технологии OpenXR и DOTS уменьшает необходимость особых</p>

	С4, С2-В3. Свобода действий в игре позволяет раскрывать потенциал пользователя С2 – В4. Развитие эвристического мышления у пользователя	специалистов и человеческих ресурсов для конкурентного развития.
Угрозы: У1. Малая заинтересованность игроков в проектах подобного жанра У2. Возможен возврат средств из-за несоответствия ожиданиям от игры У3. Создание подобных клонов проекта для вытеснения с рынка оригинала У4. Изменение критериев публикации на торговой площадке У5. Заблокируют доступ к зарубежным торговым площадкам	С1 – У1. После масштабирования можно расширить возможности жанра игры, чтобы заинтересовать больше игроков. С5 – У2. Низкий порог входа должен уменьшить количество возвратов средств.	Сл3-У1, ввиду малой целевой аудитории, готовой сразу опробовать игру, финансирование будет крайне сложно найти. Сл1, Сл2-У3. Отсутствие пиар компании и человеческих ресурсов увеличит вероятность вытеснения с рынка среди клонов. Сл5 – У2. Причиной возврата средств может служить отсутствие разнообразия в игре Сл4 – У5. Необходимость узких специалистов для обхода блокировки на торговых площадках

Таким образом, по результатам SWOT – анализа можно заключить, что данный проект имеет право на существование, но нужно быть готовым к критике и конкурентам, пускай и немногочисленным.

5.1.4 Оценка готовности проекта к коммерциализации

В таблице 10 представлен заполненный бланк оценки степени готовности проекта к коммерциализации.

Таблица 10 – Блан оценки степени готовности проекта к коммерциализации

№ п/п	Наименование	Степень проработанности научного проекта	Уровень имеющихся знаний у разработчика
1	Определен имеющийся научно-технический задел	3	3
2	Определены перспективные направления коммерциализации научно-технического задела	2	2
3	Определены отрасли и технологии (товары, услуги) для предложения на рынке	3	2

4	Определена товарная форма научно-технического задела для представления на рынок	1	1
5	Определены авторы и осуществлена охрана их прав	1	1
6	Проведена оценка стоимости интеллектуальной собственности	1	1
7	Проведены маркетинговые исследования рынков сбыта	1	1
8	Разработан бизнес-план коммерциализации научной разработки	1	1
9	Определены пути продвижения научной разработки на рынке	1	2
10	Разработана стратегия (форма) реализации научной разработки	3	3
11	Проработаны вопросы международного сотрудничества и выхода на зарубежный рынок	1	2
12	Проработаны вопросы использования услуг инфраструктуры поддержки, получения льгот	1	1
13	Проработаны вопросы финансирования коммерциализации научной разработки	1	1
14	Имеется команда для коммерциализации научной разработки	1	1
15	Проработан механизм реализации научного проекта	3	3
	ИТОГО БАЛЛОВ	24	25

В результате сумма баллов степени проработанности научного проекта равна 24 и сумма баллов по уровню имеющихся знаний равна 25, что определяет перспективность проекта ниже среднего. Для повышения перспективности проекта нужно больше направленных специалистов и развитие навыков текущих разработчиков.

5.1.5 Методы коммерциализации результатов научно-технического исследования

Проанализировав методы коммерциализации, выбран метод *организация собственного и совместных предприятий*. Так как, уже имеется соответствующий персонал для завершения разработки проекта до полного завершения. Также, учитывая, что проект создается не для коммерческих

целей, то найти финансирование на наших условиях для проекта очень маловероятно. И планируется разрабатывать проект в РФ, а распространять его на зарубежных торговых площадках, таких как Steam и Oculus.

5.2 Инициация проекта

В таблице 11 представлены заинтересованные стороны проекта.

Таблица 11 – Заинтересованные стороны проекта

Заинтересованные стороны проекта	Ожидания заинтересованных сторон
Научный руководитель	Планирование разработки проекта
Разработчик	Разработка проекта
Тестировщики	Получения результатов тестирования работы проекта

В таблице 12 представлена информация о иерархии целей проекта и критериях достижения целей. Цели проекта включают в себя цели в области ресурсоэффективности и ресурсосбережения.

Таблица 12 – Цели и результат проекта

Цели проекта:	Развить эвристические навыки пользователя
Ожидаемые результаты проекта:	Развитие эвристических навыков
Критерии приемки результата проекта:	Отсутствие багов
	Работоспособность всех запланированных функций ПО
Требования к результату проекта:	Требования:
	Соблюдение правил творческого игрового потока
	Соблюдение технических норм

В таблице 13 предоставлены участники рабочей группы проекта.

Таблица 13 – Рабочая группа проекта

№ п/п	ФИО, основное место работы, должность	Роле в проекте	Функции	Трудо-затраты, час
1	Демин Антон Юрьевич, ТПУ, Доцент ОИТ	Руководитель	Планирование проекта, консультирование персонала	10
2	Рогозин Руслан Александрович	Разработчик	Разработка всего проекта	504
ИТОГО:				514

5.3 Планирование управления научно-техническим проектом

5.3.1 План проекта

Важным этапом проведения научно-исследовательских работ является необходимость планирования работ, определение перечня работ, распределение времени работ между всеми исполнителями проекта. Исполнителями проекта являются студент и научный руководитель.

В Таблице 14 представлен перечень этапов и работ, а также распределение исполнителей по данным видам работ в рамках проводимого научно-исследовательского проекта.

Таблица 14 – Структура работ в рамках научного исследования

№	Наименование работы	Исполнители работы
1	Выбор научного руководителя магистерской работы	Рогозин Руслан Александрович
2	Составление и утверждение темы магистерской работы	Рогозин Руслан Александрович Демин Антон Юрьевич
3	Подбор и изучение литературы по теме магистерской работы	Рогозин Руслан Александрович
4	Анализ предметной области	Рогозин Руслан Александрович
5	Разработка ПО	Рогозин Руслан Александрович
	Тестирование ПО	Рогозин Руслан Александрович
6	Согласование выполненной работы с научным руководителем	Рогозин Руслан Александрович Демин Антон Юрьевич
7	Выполнение других частей работы (финансовый менеджмент, социальная ответственность)	Рогозин Руслан Александрович
8	Подведение итогов, оформление работы	Демин Антон Юрьевич Рогозин Руслан Александрович

Согласно производственному календарю (для 6-дневной рабочей недели) в 2022 году 365 календарных дней, 299 рабочих дней, 66 выходных/праздничных дней. Таким образом, коэффициент календарности на 2022 год равен:

$$T_{\text{кал}} = T_{\text{кал}} / (T_{\text{кал}} - T_{\text{вых}} - T_{\text{пр}}) = 1.22$$

После расчета коэффициента календарности можно составить таблицу временных показателей проведения научного исследования, приведенная в таблице 15 и диаграмму Ганта.

Таблица 15 – Временные показатели проведения научного исследования.

Наименование работы	Исполнители работы	Трудоёмкость работ, чел-дни			Длительность работ, дни	
		Tmin	Tmax	Toж	Tr	Tк
Выбор научного руководителя магистерской работы	Рогозин Р. А.	1	2	1,4	1	1
Составление и утверждение темы магистерской работы	Рогозин Р. А. Демин А. Ю.	1	2	1,4	1	1
Подбор и изучение литературы по теме магистерской работы	Рогозин Р. А.	1	2	1,2	1	1
Анализ предметной области	Рогозин Р. А.	5	10	7	7	9
Разработка ПО	Рогозин Р. А.	14	30	28	28	30
Тестирование ПО	Рогозин Р. А.	3	10	7	7	9
Согласование выполненной работы с научным руководителем	Рогозин Р. А. Демин А. Ю.	5	10	7	7	9
Выполнение других частей работы (финансовый менеджмент, социальная ответственность)	Рогозин Р. А.	3	4	3,4	2	2
Подведение итогов, оформление работы	Рогозин Р. А.	5	10	7	7	9

На основе таблицы 15 построен календарный план-график исследования (диаграмма Ганта), представленный в таблице 16.

Таблица 16 – Диаграмма Ганта.

Этап	Исполнители	Продолжительность	Продолжительность выполнения работ												
			Февраль		Март			Апрель			Май				
			2	3	1	2	3	1	2	3	1	2	3		
1	С	1д	■												
2	С, НР	1д	■ ■												
3	С	1д		■											
4	С	7д		■											
5	С	28д			■	■	■	■	■	■					
6	С	7д								■	■	■			
7	С, НР	7д										■	■		
8	С	2д												■	
9	С	9д													■

НР – ■ С – ■

5.3.2 Потенциальные риски

Идентифицированные риски проекта включают в себя возможные неопределенные события, которые могут возникнуть в проекте и вызвать последствия, которые повлекут за собой нежелательные эффекты. В таблице 17 представлен реестр рисков для текущего проекта.

Таблица 17 – Реестр рисков.

№	Риск	Потенциальное воздействие	Вероятность наступления	Влияние риска	Уровень риска	Способы смягчения риска	Условия наступления
1	Управление проектом	Планирование	4	5	Высокий	Разбиение пунктов планирования на более мелкие части	Запланированный этап не выполняется в срок
2	Внешний	Рынок	2	4	Средний	Выпустить продукт, как можно раньше	На рынок поступит подобный продукт
3	Технический	Технология, доступность	1	3	Средний	Использование нескольких разных видов VR гарнитур	Прекращение обслуживания текущих VR гарнитур в РФ

5.4 Бюджет научно-технического исследования

Расчет сметной стоимости ее выполнения производится по следующим статьям затрат:

- материальные затраты;
- амортизационные расходы;
- заработная плата исполнителей;
- отчисления во внебюджетные фонды;
- прочие расходы.

Существует 3 варианта выполнения исследования. Первый способ (Исп. 1) – использование бесплатных средств для проектирования. Второй способ (Исп. 2) – приобретение физических средств для проектирования. Третий способ (Исп. 3) – приобретение профессиональных цифровых средств проектирования.

С точки зрения финансовых вложений, варианты исполнения будут отличаться лишь материальными затратами. Для варианта Исп. 2 необходима покупка листов бумаги и канцелярских принадлежностей (470 рублей за пачку бумаги А4 500 листов + 92 рубля за набор ручек на весь срок написания ВКР). Для варианта Исп. 3 необходимо покупка приложения для проектирования XMind Pro за 8174 рубля на весь срок написания ВКР.

Материальные затраты включает в себя затраты на электроэнергию, потраченную в ходе выполнения проекта на работу используемого оборудования, рассчитываемые по формуле:

$$C_{\text{эл.об}} = P_{\text{об}} * t_{\text{об}} * \text{ЦЭ}$$

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

ЦЭ – тариф на 1 кВт·час;

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

Для Томского политехнического университета ЦЭ = 6,59 руб./квт·час (с учетом НДС). Время работы оборудования, определяется по формуле:

$$t_{\text{об}} = T_{\text{рд}} \cdot K_t$$

где $T_{\text{рд}}$ – трудоемкость работ, из расчета, что продолжительность рабочего дня равна 8 часов;

$K_t \leq 1$ – коэффициент использования оборудования по времени, равный отношению времени его работы в процессе выполнения проекта к ТРД, определяется исполнителем самостоятельно. Мощность, потребляемая оборудованием, определяется по формуле:

$$P_{\text{об}} = P_{\text{ном}} \cdot K_c$$

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

$K_c \leq 1$ – коэффициент загрузки, зависящий от средней степени использования номинальной мощности. Для технологического оборудования малой мощности $K_c = 1$.

Таблица 18 – Затраты на электроэнергию

Наименование оборудования	Время работы оборудования tОБ, час	Потребляемая мощность РОБ, кВт	Затраты ЭОБ, руб
Ноутбук	608 * 0,9	0,065	35,57
Итого:			35,57

В статье «Амортизационные отчисления» рассчитывается амортизация используемого оборудования за время выполнения проекта. Рассчитывается по формуле:

$$C_{ам} = (N_A \cdot C_{об} \cdot t_{рф} \cdot n) / F_D$$

где N_A – годовая норма амортизации единицы оборудования;

$C_{об}$ – балансовая стоимость единицы оборудования с учетом ТЗР;

$t_{рф}$ – фактическое время работы оборудования в ходе выполнения проекта, учитывается исполнителем проекта;

n – число задействованных однотипных единиц оборудования;

F_D – действительный годовой фонд времени работы соответствующего оборудования.

$$N_A = 1 / 2,5 = 0,4$$

При шестидневной рабочей неделе $F_D = 299 \cdot 8 = 2392$ часа, отсюда рассчитаем амортизация используемого оборудования за время выполнения проекта.

$$C_{ам} = (0,4 \cdot 65000 \cdot 608 \cdot 1) / 2392 = 6\,608,7 \text{ руб.}$$

Затраты на заработную плату:

$$Z_{п} = Z_{осн} + Z_{доп}$$

Здесь $Z_{осн}$ – основная заработная плата, руб;

$Z_{доп}$ – дополнительная заработная плата, руб.

Заработная плата основная:

$$Z_{осн} = Z_{дн} \cdot T_p \cdot (1 + K_{пр} + K_d) \cdot K_p$$

Здесь $Z_{дн}$ – средневзвешенная заработная плата, руб;

$K_{пр}$ – премиальный коэффициент (0,3);

K_d – коэффициент доплат и надбавок (0,2–0,5);

K_p – районный коэффициент (для Томска равен 1,3);

T_p – продолжительность работ, выполняемых работником, раб. дни.

Среднедневная заработная плата:

$$Z_{\text{дн}} = Z_{\text{м}} * M / F_{\text{д}}$$

Здесь $Z_{\text{м}}$ – месячный оклад работника, руб.;

M – количество месяцев работы без отпуска в течение года;

$F_{\text{д}}$ – действительный годовой фонд рабочего времени персонала, раб. дн.

M следует принять за 10,4 месяца, так как рабочая неделя длится 6 дней, а отпуск – 48 дней.

Таблица 19 – Баланс рабочего времени (для 6-дневной недели)

Показатели рабочего времени	Дни
Календарные дни	365
Нерабочие (праздники/выходные)	66
Потери рабочего времени (отпуск/невыходы по болезни)	56
Действительный годовой фонд рабочего времени	243

Таким образом, в 2022 году действительный годовой фонд рабочего времени составляет 243 дня. Исходя из всех найденных показателей можно составить таблицу расчета основной заработной платы. Стоит отметить, что зарплата студента в месяц равняется 4030, а руководителя – 33664 рублей.

Таблица 20 – Расчет основной заработной платы

Исполнители	Оклад, руб./мес.	Среднедневная ставка, руб./раб.день	Затраты времени, раб.дни	Коэффициент	Фонд з/платы, руб.
Демин Антон Юрьевич	33664	1346,56	8	1,699	18302,45
Рогозин Руслан Александрович	4030	161,2	63	1,699	17254,37
Итого:					35556,82
отчисления в социальные внебюджетные фонды 30 %					10667,05

В статью «Прочие расходы» включены расходы, которые не учтены в предыдущих статьях и составляют 10 % от суммы всех предыдущих расходов

$$(Исп 1) C_{\text{проч}} = (C_{\text{мат}} + C_{\text{зп}} + C_{\text{соц}} + C_{\text{эл.об}} + C_{\text{ам}}) \cdot 0,1 = (0 + 35556,82 + 10667,05 + 35,57 + 6\ 608,7) \cdot 0,1 = 5286,82 \text{ руб}$$

$$(Исп 2) C_{\text{проч}} = (C_{\text{мат}} + C_{\text{зп}} + C_{\text{соц}} + C_{\text{эл.об}} + C_{\text{ам}}) \cdot 0,1 = (470 + 35556,82 + 10667,05 + 35,57 + 6\ 608,7) \cdot 0,1 = 5333,82 \text{ руб}$$

$$(Исп 3) C_{\text{проч}} = (C_{\text{мат}} + C_{\text{зп}} + C_{\text{соц}} + C_{\text{эл.об}} + C_{\text{ам}}) \cdot 0,1 = (8174 + 35556,82 + 10667,05 + 35,57 + 6\ 608,7) \cdot 0,1 = 6104,22 \text{ руб}$$

Таблица 21 – Расчет бюджета затрат НТИ

Статья затрат	Условное обозначение	Сумма, руб		
		Исп. 1	Исп. 2	Исп. 3
Материалы и покупные изделия	$C_{\text{мат}}$	0	470	8174
Основная заработная плата	$C_{\text{зп}}$	35 556,82	35 556,82	35 556,82
Отчисления в социальные фонды	$C_{\text{соц}}$	10 667,05	10 667,05	10 667,05
Расходы на электроэнергию	$C_{\text{эл.об}}$	35,57	35,57	35,57
Амортизационные отчисления	$C_{\text{ам}}$	6 608,7	6 608,7	6 608,7
Прочие расходы	$C_{\text{проч}}$	5 286,82	5 333,82	6 104,22
Итого:		58 154,96	58 671,96	67 146,36

5.5 Оценка сравнительной эффективности исследования

В таблице 22 представлена сравнительная оценка характеристик вариантов использования проекта.

Таблица 22 – Сравнительная оценка характеристик вариантов использования проекта

Критерии \ ПО	Весовой коэффициент параметра	Текущий проект (Исп. 1)	Исп. 2	Исп. 3
1. Способствует росту производительности труда пользователя	0,1	4	3	5
2. Удобство в эксплуатации (соответствует требованиям потребителя)	0,15	5	5	5
3. Помехоустойчивость	0,15	5	4	5

4. Энергосбережения	0,2	4	5	3
5. Надежность	0,25	5	5	4
6. Материалоемкость	0,15	4	4	5
ИТОГО	1	27	26	27

$$T_{п(Исп.1)} = 4 * 0,1 + 5 * 0,15 + 5 * 0,15 + 4 * 0,2 + 5 * 0,25 + 4 * 0,15 = 4,55$$

$$Исп.2 = 3 * 0,1 + 5 * 0,15 + 4 * 0,15 + 5 * 0,2 + 5 * 0,25 + 4 * 0,15 = 4,5$$

$$Исп.3 = 5 * 0,1 + 5 * 0,15 + 5 * 0,15 + 3 * 0,2 + 4 * 0,25 + 5 * 0,15 = 4,35$$

В таблице 23 представлена сравнительная эффективность разработки.

Таблица 23 – Сравнительная эффективность разработки

№ п/п	показатели	Текущий проект (Исп. 1)	Исп. 2	Исп. 3
1	Интегральный финансовый показатель разработки	0,87	0,88	1
2	Интегральный показатель ресурсоэффективности разработки	4,55	4,5	4,35
3	Интегральный показатель эффективности	5,23	5,14	4,35
4	Сравнительная эффективность вариантов использования	1	1,02	1,21

Итак, распланирована структура работ проекта и определены ответственные должности для их выполнения. В соответствии с назначенными работами была рассчитана их трудоемкость и составлен график работ.

Общая длительность проектирования и разработки программного продукта составила 63 дня. Общий бюджет составил 58 154,96 рублей для текущего проекта (Исп. 1 = 4.55). Он включает в себя затраты на основную и дополнительную заработную плату работников, отчисления на внебюджетные фонды и накладные расходы.

6 Социальная ответственность

6.1 Правовые и организационные вопросы обеспечения безопасности

6.1.1 Специальные правовые нормы трудового законодательства

Трудовые отношения между работодателем и работником регулируются с помощью законодательного акта "Трудовой кодекс Российской Федерации" от 30.12.2001 N 197-ФЗ (ред. от 26.05.2022) [12]. Это регулирование помогает найти баланс в вопросах, связанных с организацией труда, управлением трудом, заработной платой, трудовыми спорами и т.д. Ниже приведены его наиболее важные для соблюдения фрагменты:

- Нормальная продолжительность рабочего времени не должна превышать 40 часов в неделю;
- В течение рабочего дня (смены) работнику должен быть предоставлен перерыв для отдыха и питания продолжительностью не более двух часов и не менее 30 минут, который в рабочее время не включается. Правилами внутреннего трудового распорядка может быть установлена отмена перерыва, если установленная для работника продолжительность ежедневной работы (смены) не превышает четырех часов;
- Всем работникам предоставляются выходные дни (еженедельный непрерывный отдых).

6.1.2 Мероприятия по планировке рабочего места

Разработка ПО от начала и до конца велась в одном помещении — аудитории общей площадью 28 квадратных метров. Днем работа проводилась с использованием солнца — естественного источника освещения, утром и вечером помещение освещалась 3 светодиодными лампами по 13 Вт мощности в каждой, что эквивалентно 100 Вт ламп накаливания. Помещение оборудовано большим компьютерным столом и компьютерным стулом стандартного размера. Кресло может вращаться на 360 градусов по

горизонтали и оборудовано 5 пластиковыми колесами диаметром по 55 мм каждое. С целью минимизации воздействия вредных факторов на автора ПО при ее создании, рабочее место должно быть организовано с учетом требований ГОСТ 12.2.032-78 «ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования» [13].

Ниже приведены наиболее важные для соблюдения фрагменты стандарта:

- При работе двумя руками органы управления размещают с таким расчетом, чтобы не было перекрещивания рук.
- Помещения, где размещаются рабочие места с ПЭВМ, должны быть оборудованы защитным заземлением (занулением) в соответствии с техническими требованиями по эксплуатации.
- В помещениях, оборудованных ПЭВМ, проводится ежедневная влажная уборка и систематическое проветривание после каждого часа работы на ПЭВМ.
- Подвижность кресла относительно пола или другой поверхности, на которой оно установлено, может не ограничиваться. В случае необходимости обеспечения строго определенного положения человека-оператора по отношению к средствам отображения информации и органам управления, а также в случае, если трудовая деятельность человека-оператора сопряжена с силовыми и резкими движениями, кресло должно быть фиксировано. При этом, в зависимости от характера трудовой деятельности оператора, должна быть обеспечена возможность изменения положения кресла или сиденья в горизонтальной плоскости с фиксацией его в нужном положении. При необходимости подвижность кресла должна задаваться также вращением кресла на 180-360° вокруг вертикальной оси опорной конструкции кресла с фиксацией в нужном положении.

- Рабочие столы следует размещать таким образом, чтобы видеодисплейные терминалы были ориентированы боковой стороной к световым проемам, чтобы естественный свет падал преимущественно слева.
- Экран видеомонитора должен находиться от глаз пользователя на расстоянии 600-700 мм, но не ближе 500 мм с учетом размеров алфавитноцифровых знаков и символов.

Автором были соблюдены все требования ГОСТ 12.2.032-78 [13]. Во время выполнения выпускной квалификационной работы не происходило случаев, несущих в себе угрозу для здоровья и жизни, а также предоставления опасности для окружающей среды.

6.2 Производственная безопасность

В данном подразделе производится анализ вредных и опасных факторов, которые могут возникнуть на одном из этапов выполнения работ по созданию проекта.

Отказ от рассмотрения химических факторов обусловлен незначительностью их наличия на рабочем месте или их полным отсутствием. В таблице 24 представлены возможные опасные и вредные факторы.

Таблица 24 – Возможные опасные и вредные факторы

Факторы (ГОСТ 12.0.003-2015)	Этапы работ			Нормативные документы
	Проектирование	Разработка	Тестирование	
1. Отклонение показателей микроклимата в закрытых помещениях	+	+	+	СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания [14] ГОСТ 12.1.005-88 Система стандартов безопасности труда (ССБТ). Общие санитарно-гигиенические требования к воздуху рабочей зоны [15]
2. Отсутствие или недостаток естественного света	+	+	+	СНиП 23-05-95* Естественное и искусственное освещение [16]

3. Недостаточная освещенность рабочей зоны	+	+	+	СНиП 23-05-95* Естественное и искусственное освещение [16]
4. Повышенный уровень электромагнитных излучений	+	+	+	СанПиН 2.1.8/2.2.4.1383-03 Гигиенические требования к размещению и эксплуатации передающих радиотехнических объектов [17]
5. Нервно-психические перегрузки, монотонность трудового процесса	+	+	+	Трудовой кодекс Российской Федерации от 30.12.2001 №197-ФЗ (ред. от 27.12.2018) [12]
6. Повышенный уровень шума	+	+	+	ГОСТ 12.1.003-2014 Система стандартов безопасности труда. Шум. Общие требования безопасности [18]
7. Удар электрическим током	+	+	+	ГОСТ 12.1.019-2017 ССБТ. Электробезопасность. Общие требования и номенклатура видов защиты [19]

Представленные выше производственные факторы оказывают негативное влияние на организм человека, поэтому необходимо знать о допустимых нормах их воздействия.

6.2.1 Отклонения показателей микроклимата

Микроклимат определяется действующими на организм человека показателями температуры, влажности и скорости движения воздуха. Длительное воздействие на человека неблагоприятных показателей микроклимата ухудшает его самочувствие, снижает производительность труда и приводит к заболеваниям, поэтому в организации должны обеспечиваться оптимальные параметры микроклимата. Оптимальные и допустимые величины показателей микроклимата на рабочих местах [14] представлены в таблице 25.

Таблица 25 – Оптимальные и допустимые величины показателей микроклимата на рабочих местах

Оптимальные значения характеристик микроклимата				
Период года	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	22 – 24	21 – 25	40 – 60	0,1

Теплый	23 – 25	22 – 26	40 – 60	0,1
Допустимые значения характеристик микроклимата				
Период года	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	20 – 25	19 – 26	15 – 75	0,1
Теплый	21 – 28	20 – 29	15 – 75	0,1 – 0,1

6.2.2 Недостаточная освещенность

Недостаточная освещенность рабочей зоны помещения, оборудованной ПК, также является одной из причин нарушения зрительной функции, а также влияет на общее самочувствие и эффективность труда.

Искусственное освещение в помещениях для эксплуатации ПК должно осуществляться системой общего равномерного освещения. В случаях преимущественной работы с документами, следует применять системы комбинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения, предназначенные для освещения зоны расположения документов). Окна в помещениях, где эксплуатируется вычислительная техника, преимущественно должны быть ориентированы на север и северо-восток. Нормируемые показатели естественного, искусственного и совмещенного освещения [16] указаны в таблице 26.

Таблица 26 – Нормируемые показатели естественного, искусственного и совмещенного освещения

Помещения	Рабочая Поверхность и плоскость нормирования КЕО и освещенности (Горизонтальная, Вертикальная) и высота	Естественное освещение		Совмещенное освещение	
		КЕО е н, %		КЕО е н, %	
		При верхнем или комбинированном освещении	При боковом освещении	При верхнем или комбинированном освещении	При боковом освещении

	плоскости над полом, м				
Кабинеты, рабочие комнаты, офисы, представительства	Г – 0,8	3,0	1,0	1,8	0,6
Помещения	Искусственное освещение				
	Освещенность, лк				
	При комбинированном освещении		При общем освещении	Показатель дискомфорта, М, не более	Коэффициент пульсации освещенности, Кп, %, не более
	Всего	От общего			
Кабинеты, рабочие комнаты, офисы, представительства	400	200	300	40	15

На рабочем месте используется люминесцентные газоразрядные лампы (открытые двухламповые светильники типа ОД) белой цветности для общего равномерного освещения.

Дано помещение с размерами: длина $A = 4$ м, ширина $B = 7$ м, высота $H = 2,5$ м. Высота рабочей поверхности $h_{рп} = 0,8$ м. Требуется создать освещенность $E = 300$ лк.

Коэффициент отражения стен $R_c = 30$ %, потолка $R_n = 50$ %. Коэффициент запаса $k = 1,5$, коэффициент неравномерности $Z = 1,1$.

Рассчитываем систему общего люминесцентного освещения.

Выбираем светильники типа ОД, $\lambda = 1,4$.

Приняв $h_c = 0,5$ м, получаем $h = 2,5 - 0,5 - 0,8 = 1,2$ м;

$$L = 1,4 \cdot 1,2 = 1,68 \text{ м}; L/3 = 0,56 \text{ м}.$$

Размещены светильники в два ряда. В каждом ряду установлены по 2 светильника типа ОД мощностью 40 Вт (с длиной 1,23 м), при этом разрывы между светильниками в ряду составят 50 см. Учитывая, что в каждом светильнике установлено две лампы, общее число ламп в помещении $N = 8$.

Находим индекс помещения

$$i = 28 / 1,2 \cdot (4 + 7) = 2,12 .$$

Определяем коэффициент использования светового потока:

$$\eta = 0,61.$$

$$\Phi \cdot = (300 \cdot 28 \cdot 1,5 \cdot 1,1) / (8 \cdot 0,61) = 2840 \text{ Лм.}$$

Определяем электрическую мощность осветительной установки:

$$P = 8 \cdot 40 = 320 \text{ Вт.}$$

6.2.3 Повышенный уровень электромагнитных излучений

Увеличение количества и рост мощности различных искусственных источников (радиосвязь, радиолокация, расширение сетей высоковольтных линий электропередач, микроволновые печи, компьютеры, ноутбуки, телевизоры, транспорт) приводят к значительному росту уровня электромагнитных излучений в городах и населенных пунктах, которые создают дополнительное искусственное электромагнитное поле, которое неблагоприятно влияет на здоровье человека.

Электромагнитные волны приводят к неблагоприятным изменениям в организме. В таблице 27 представлены предельно допустимые уровни ЭМП диапазона частот 30 кГц-300 ГГц на рабочих местах персонала [17].

Таблица 27 – Предельно допустимые уровни ЭМП диапазона частот 30 кГц-300 ГГц на рабочих местах персонала

Параметр	Диапазонах частот (МГц)				
	0,03-3,0	3,0-30,0	30,0-50,0	50,0-300,0	300,0-300000
Предельно допустимое значение $\mathcal{E}\mathcal{E}_E$, (В/м) ² · ч	20000	7000	800	800	-
Предельно допустимое значение $\mathcal{E}\mathcal{E}_H$, (А/м) ² · ч	200	-	0,72	-	-
Предельно допустимое значение $\mathcal{E}\mathcal{E}_{ППЭ}$, (мкВт/см ²) · ч	-	-	-	-	200
Максимальный ПДУ E , В/м	500	296	80	80	-
Максимальный ПДУ H , А/м	50	-	3,0	-	-
Максимальный ПДУ $ППЭ$, мкВт/см ²	-	-	-	-	1000

6.2.4 Нервно-психические перегрузки

Нервно-психические перегрузки – совокупность таких сдвигов в психофизиологическом состоянии организма человека, которые развиваются после совершения работы и приводят к временному снижению эффективности труда. Состояние утомления (усталость) характеризуется определенными объективными показателями и субъективными ощущениями.

Нервно-психические перегрузки подразделяются на следующие:

- умственное перенапряжение;
- перенапряжение анализаторов;
- монотонность труда;
- эмоциональные перегрузки.

При первых симптомах психического перенапряжения необходимо:

- дать нервной системе расслабиться;
- рационально чередовать периоды отдыха и работы;
- начать заниматься спортом;
- ложиться спать в одно и то же время;
- в тяжелых случаях обратиться к врачу.

Естественно, что полностью исключить провоцирующие факторы из жизни вряд ли удастся, но можно уменьшить их негативное воздействие, давая нервной системе необходимый отдых [12].

6.2.5 Повышенный уровень шума

Влияние шума на слуховой анализатор проявляется в ауральных эффектах, которые, главным образом, заключаются в медленно прогрессирующем понижении слуха по типу неврита слухового нерва (кохлеарный неврит). В таблице 28 представлен предельно допустимые уровни звукового давления, уровни звука и эквивалентные уровни звука для основных наиболее типичных видов трудовой деятельности и рабочих мест [18].

Таблица 28 – Предельно допустимые уровни звукового давления, уровни звука и эквивалентные уровни звука для основных наиболее типичных видов трудовой деятельности и рабочих мест

Вид трудовой деятельности, рабочее место	Уровни звукового давления, дБ, в октавных полосах со среднегеометрическими частотами, Гц								
	31,5	63	125	250	500	1000	2000	4000	8000
Выполнение всех видов работ на постоянных рабочих местах в производственных помещениях и на территории	107	95	87	82	78	75	73	71	69

Подвергающиеся шумовому воздействию люди, чаще всего жалуются на головные боли, которые могут иметь разную интенсивность и локализацию, головокружение при перемене положения тела, снижение памяти, повышенную утомляемость, сонливость, нарушения сна, эмоциональную неустойчивость, снижение аппетита, потливость, боли в области сердца. Шум – это один из самых сильных стрессорных агентов. Влияние шума сказывается на функциях эндокринной и иммунной систем организма, в частности это может проявляться в виде трех главных биологических эффектов: снижение иммунитета к инфекционным болезням; снижение иммунитета, направленного против развития опухолевых процессов; появление благоприятных условий для возникновения и развития аллергических и аутоиммунных процессов.

6.2.6 Удар электрическим током

В таблице 29 представлены защитные меры при основной защите и защите при наличии неисправности [19].

Таблица 29 – Защитные меры при основной защите и защите при наличии неисправности

Защитные меры	Основная защита (защита при отсутствии неисправности)	Защита при наличии неисправности (защита при наличии простой неисправности)
Защита с использованием двойной или усиленной изоляции	Усиленная изоляция	
	Основная изоляция	Дополнительная изоляция

Защита с использованием выравнивания потенциалов	Основная изоляция, варианты: - (твердая) основная изоляция; - основная изоляция: внутри ограждений и оболочек; - за барьерами; - размещение вне зоны досягаемости	Выравнивание потенциалов, обеспечивающее защиту, варианты (одна мера защиты или подходящая комбинация следующих мер): - выравнивание потенциалов в электроустановке; - выравнивание потенциалов для электрооборудования; - РЕ-проводник; - PEN-проводник; - защитный экран
Защита с использованием автоматического отключения источника питания	-	Автоматическое отключение источника питания
Защита с использованием электрического разделения цепей	-	Простое разделение цепей
Защита с использованием нетокопроводящей среды	-	Нетокопроводящая среда
Защита с использованием других защитных мер	Другие меры	Другие меры
	Другие меры усиленной защиты	

Удар электрическим током моментально вызывает у пострадавшего сильный шок, который может сопровождаться галлюцинациями. При поражении больше всего страдают головной мозг и сердце. Последствия удара током могут быть явными: ожоги, боль, нарушения сердечного и дыхательного ритмов, головокружение, нарушение зрения, сознания, иногда возбуждение, ретроградная амнезия, разрывы мышц при их судорожном сокращении, также возможны компрессионные и отрывные переломы костей. Если удар был несильным, последствия могут быть не очевидными, но не менее опасными для жизни и здоровья человека.

6.2.7 Определение мероприятий по уменьшению вреда от воздействия выявленных факторов

Чтобы уменьшить воздействие вредных факторов необходимо обеспечивать определенные мероприятия как при планировании правильного обустройства рабочего помещения, так и в процессе работы. Таблица 30 описывает некоторые мероприятия, позволяющий уменьшить вред выявленных факторов.

Таблица 30 – Определение мероприятий по снижению воздействия вредных факторов

Факторы	Мероприятия по снижению воздействия
Поддержание микроклимата	Обеспечение помещения системами обогрева, вентилирования, увлажнения
	Защита от солнца: шторы, жалюзи, навесы
	Правильное размещения рабочих мест
	Влажная уборка
Плохая освещенность	Своевременная чистка стекол окон и осветительных приборов
	Ремонт помещения в светлых тонах
	Установка более мощных ламп в правильных местах
Повышенный уровень электромагнитный излучений	Использовать ЖК-экран
	Располагать монитор в углу помещения – стены поглощают излучение
	Выключать компьютер в перерывах между работой
	Сокращать время за компьютером
Нервно-психические перегрузки	Работа не более 8 часов в день
	Работа итерационно, прерываясь на небольшой отдых каждый час
	Выходные каждую неделю
Удар электрическим током	Ежегодная проверка компонентов ЭВМ на поломки
	Изменить влажность воздуха в соответствии с нормами микроклимата
	Проверить заземление розеток

6.3 Экологическая безопасность

Выполнение поставленного проекта требует использование ноутбука и малочисленной периферии — все это не выбрасывает в окружающую среду каких-либо вредных веществ.

Однако, производство данной электротехники, ее утилизация и утилизация канцелярских принадлежностей и бумаги — факторы, отрицательно влияющие на экологическую безопасность на планете Земля [21].

6.3.1 Производство и утилизация техники и комплектующих

Производство и утилизация современных вычислительных устройств составляют серьезную проблему: текстолит, используемый при производстве микросхем, имеющих срок разложения более тысячи лет [21].

Кроме того, при производстве вредные вещества не редко попадают в водоемы. Неочищенные стоки сливаются непосредственно в водоёмы и содержат ряд ядовитых химических соединений, тяжёлые металлы, а иногда и радиоактивные компоненты [21].

Бытовой мусор и канцелярские принадлежности при неверной утилизации затрудняет циркуляцию воды и разлагаясь отравляет водоёмы.

6.3.2 Нормы экологической безопасности

У офиса должен быть заключен договор на вывоз и утилизацию отходов.

Нельзя складировать и вывозить совместно твердые коммунальные отходы, организованные в процессе жизнедеятельности, и отходы оргтехники. Макулатура не подлежит захоронению, поскольку является вторичным материальным ресурсом, т.е. может быть использована повторно.

Вывоз отходов на утилизацию и захоронение должен быть подтвержден документально, иначе отходы будут рассматриваться проверяющими как несанкционированные [22].

В офисе должен быть организован учет образования и движения отходов.

6.3.3 Мероприятия по сохранению экологической безопасности

Мероприятия, позволяющие сохранять экологическую безопасность, находясь на своем рабочем месте:

- Правильная утилизация персональных компьютеров и ноутбуков, а также их комплектующих;
- Использование энергосберегающих ламп;
- Использование аккумуляторов вместо солевых батареек;
- Своевременное отключение неиспользуемых электроприборов;

— Проведение экологического аудита.

6.4 Безопасность в чрезвычайных ситуациях

Создание проекта проходило в учебной аудитории ТПУ. Возможен следующий список ЧС:

- Пожар;
- Наводнение;
- Землетрясение;
- Террористический акт;
- Пандемия.

Наиболее вероятное чрезвычайное происшествие — это пожар в здании.

6.4.1 Пожар

Основными причинами пожара являются: короткое замыкание, перегрузка сети, большое переходное сопротивление, искрение и т.д.

В связи с этим, согласно ГОСТ 12.1.004-91 «Пожарная безопасность. Общие требования», при работе с компьютером необходимо соблюдать следующие **нормы пожарной безопасности** [23]:

- для предохранения сети от перегрузок запрещается одновременно подключать к сети количество потребителей, превышающих допустимую нагрузку;
- работы за компьютером проводить только при исправном состоянии оборудования, электропроводки;
- иметь средства для тушения пожара (огнетушитель);
- установить количество, размеры и соответствующее конструктивное исполнение эвакуационных путей и выходов;
- обеспечить возможность беспрепятственного движения людей по эвакуационным путям. Прокладка всех видов кабелей в металлических газонаполненных трубах;
- отличный вариант для предотвращения возгорания.

Главное во время пожара – не поддаваться панике и действовать согласно **правилам поведения** в подобной чрезвычайной ситуации:

1. Вызовите пожарную охрану, сообщив свой адрес, фамилию и номер телефона.
2. Отключите все электрооборудование, бытовые приборы, газ.
3. Закройте (или не открывайте) двери и окна, свободный доступ кислорода только усилит огонь.
4. Если пламя еще небольшое, попытайтесь его погасить своими силами:
 - a. воспользовавшись огнетушителем;
 - b. накрыв очаг плотной тканью (шторы, одеяло);
 - c. засыпав очаг землей или песком;
 - d. залив очаг водой.
5. В случае неудавшегося тушения, срочно эвакуироваться из помещения (здания). При этом не пользоваться лифтом! По вентиляционным и лифтовым шахтам огонь распространяется с огромной скоростью – десятки метров в секунду. По возможности используйте запасные пожарные лестницы и выходы, которые показаны на рисунке 45.



Рисунок 45 – План эвакуации при пожаре и других ЧС

6.5 Вывод по разделу

В ходе исследования по социальной ответственности были рассмотрены правовые и организационные вопросы обеспечения безопасности, а также организационные мероприятия при компоновке рабочей зоны.

Выделены и проанализированы вредные и опасные факторы, влияющие на человека, находящегося в офисе. Были определены средства защиты, от выявленных вредных и опасных факторов.

Были выявлены бытовые отходы, негативно влияющие на окружающую среду. К ним относятся севшие аккумуляторные устройства, бумажные отходы, перегоревшие лампы. Данные отходы на месте работы утилизируются согласно ГОСТ.

Рассмотрена наиболее распространенная чрезвычайная ситуация – пожар. Пожар может быть следствием короткого замыкания или неверной эксплуатации электроприборов, а также несоблюдения техники безопасности. В разделе была приведена схема эвакуации при пожаре и других ЧС.

Правовые и организационные вопросы обеспечения безопасности регулируются государственными органами. Правила и нормы для обеспечения нормальных условий труда всех сотрудников устанавливаются на государственном уровне.

Заключение

В ходе выполнения выпускной квалификационной работы была разработано ПО в виртуальной реальности для понимания и развития эвристического мышления.

Изначально было необходимо изучить и проанализировать предметную область — эвристика. Это было сделано в рамках НИРС, где выбранные темы были тесно связаны с данной предметной областью.

Перед началом разработки было важно выбрать инструментарий — игровой движок. В ходе сравнения характеристик и особенностей выбор пал на Unity, как самый удобный и функциональный.

После выбора инструментария пришел черед проектирования — разработки ПО. Была продумана творческая игровая механика и система эвристического отбора.

В результате разработана и протестировано игровая механика соединения / разъединения объектов и система эвристического отбора, позволяющая получить данные из исходной базы данных, отсортированные напрямую или косвенно по начальным входным данным и фильтру.

Список использованных источников

1. Unity. Лого [Электронный ресурс]: Режим доступа: [https://ru.wikipedia.org/Unity_\(игровой_движок\)#/media/Файл:Unity_Technologies_logo.svg](https://ru.wikipedia.org/Unity_(игровой_движок)#/media/Файл:Unity_Technologies_logo.svg)
2. Игровой движок Unity [Электронный ресурс] // Режим доступа: <https://unity.com/ru>, свободный. – Загл. с экрана. (02.07.2021).
3. Unreal Engine. Лого [Электронный ресурс]: Режим доступа: https://ru.wikipedia.org/wiki/Unreal_Engine#/media/Файл:Unreal_Engine_logo_and_wordmark.png
4. Unreal Engine [Электронный ресурс]: Режим доступа: https://ru.wikipedia.org/wiki/Unreal_Engine
5. Godot. Лого [Электронный ресурс]: Режим доступа: https://ru.wikipedia.org/wiki/Godot#/media/Файл:Godot_logo.svg
6. Godot [Электронный ресурс]: Режим доступа: <https://ru.wikipedia.org/wiki/Godot>
7. GameMaker: Studio. Лого [Электронный ресурс]: Режим доступа: https://ru.wikipedia.org/wiki/GameMaker:_Studio#/media/Файл:GameMaker_Studio_2_logo.png
8. GameMaker: Studio [Электронный ресурс]: Режим доступа: https://ru.wikipedia.org/wiki/GameMaker:_Studio
9. GetBrain Rider [Электронный ресурс]: Режим доступа: <https://www.jetbrains.com/ru-ru/rider/>
10. Пакеты DOTS [Электронный ресурс] // Режим доступа: <https://unity.com/ru/dots/packages#unity-animation-experimental>, свободный. — Загл. с экрана (дата обращения: 10.02.2022).
11. Концепции ECS [Электронный ресурс] // Режим доступа: https://docs.unity3d.com/Packages/com.unity.entities@0.17/manual/ecs_core.html, свободный. — Загл. с экрана (дата обращения: 10.02.2022).

- 12.Трудовой кодекс РФ [Электронный ресурс]. – Режим доступа: https://stud.lms.tpu.ru/pluginfile.php/1246361/mod_resource/content/1/trudkod.pdf (дата обращения: 16.05.2022).
- 13.ГОСТ 12.2.032-78 «ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования» [Электронный ресурс]. – Режим доступа: https://stud.lms.tpu.ru/pluginfile.php/1246357/mod_resource/content/3/ГОСТ%2012.2.032-78.pdf (дата обращения: 16.05.2022).
- 14.СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания» [Электронный ресурс]. – Режим доступа: https://stud.lms.tpu.ru/pluginfile.php/1654987/mod_resource/content/1/san-pin-1-2-3685-21.pdf (дата обращения: 16.05.2022).
- 15.ГОСТ 12.1.005-88 «Система стандартов безопасности труда (ССБТ). Общие санитарно-гигиенические требования к воздуху рабочей зоны» [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200003608> (дата обращения: 16.05.2022).
- 16.СНиП 23-05-95* «Естественное и искусственное освещение» [Электронный ресурс]. – Режим доступа: https://stud.lms.tpu.ru/pluginfile.php/1246374/mod_resource/content/3/СП%2052.13330.2016_Освещение.pdf (дата обращения: 16.05.2022).
- 17.СанПиН 2.1.8/2.2.4.1383-03 «Гигиенические требования к размещению и эксплуатации передающих радиотехнических объектов» [Электронный ресурс]. – Режим доступа: <https://files.stroyinf.ru/Data1/41/41278/index.htm> (дата обращения: 16.05.2022).
- 18.ГОСТ 12.1.003-2014 «Система стандартов безопасности труда. Шум. Общие требования безопасности» [Электронный ресурс]. – Режим доступа: https://stud.lms.tpu.ru/pluginfile.php/1246373/mod_resource/content/1/ГОСТ%2012.1.003-2014_Шум.pdf (дата обращения: 16.05.2022).

- 19.ГОСТ 12.1.019-2017 ССБТ. «Электробезопасность. Общие требования и номенклатура видов защиты» [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200161238> (дата обращения: 16.05.2022).
- 20.ГОСТ 17.4.3.04-85 «Охрана природы (ССОП). Почвы. Общие требования к контролю и охране от загрязнения» [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200020658> (дата обращения: 16.05.2022).
- 21.ГОСТ Р 55102-2012 «Ресурсосбережение. Обращение с отходами. Руководство по безопасному сбору, хранению, транспортированию и разборке отработавшего электротехнического и электронного оборудования, за исключением ртутисодержащих устройств и приборов» [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200104723> (дата обращения: 16.05.2022).
- 22.ГОСТ Р 57740-2017 «Ресурсосбережение. Обращение с отходами. Требования к приему, сортировке и упаковыванию опасных твердых коммунальных отходов» [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200020658> (дата обращения: 16.05.2022).
23. N 123-ФЗ от 22.07.2008 (ред. от 30.04.2021) «Технический регламент о требованиях пожарной безопасности» [Электронный ресурс]. – Режим доступа:
https://stud.lms.tpu.ru/pluginfile.php/1654991/mod_resource/content/1/ФЗ%20№123%20Технический%20регламент%20о%20требованиях%20пожарной%20безопасности..pdf (дата обращения: 16.05.2022).
24. Эвристическая игра, или как обрести собственный смысл обучения [Электронный ресурс]. – Режим доступа: <https://elib.bsu.by/bitstream/123456789/264774/1/12-14.pdf> (дата обращения: 16.05.2022).

Приложение А

(справочное)

Problems and prospects of heuristic development

Студент

Группа	ФИО	Подпись	Дата
8ИМ02	Рогозин Руслан Александрович		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Демин Антон Юрьевич	к.т.н., Доцент		

Консультант-лингвист отделения иностранных языков ШБИП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель ОИЯ	Пичугова Инна Леонидовна			

Problems and prospects of heuristic learning

Nowadays, heuristics is the science of productive thinking (the creation of the previously unknown), aimed at the creative solution of a problem (task). Consequently, heuristics is closely related to creative and project activities, where enlightenment and insight are the leading element. In addition, heuristic training includes non-standard mental operations and manifests itself in personality characteristics:

- the ability of a person to make a heuristic choice among numerous possible options for finding solutions;
- the ability of a person to deduce guesses that significantly reduce the time to find a solution to a problem;
- the ability of a person to intuitive thinking;
- the ability of a person to integrate new ideas;
- the ability of a person to understand the conditions for finding solutions to problems, developing hypotheses.

It should be noted that heuristic learning contributes to the disclosure of the internal reserves of the individual, which help him to create new knowledge, to construct it in his activities. The main features of heuristic methods can be distinguished:

- psychological strategic idea including the removal of inertia of thinking, the coverage of all possible approaches to solving the problem, the definition of the nearest areas of the task;
- adaptive component of the methods, i.e. purposeful solution of the problem taking into account the specifics and subjective-personal qualities of the problem solver;
- interface with massive and informational components of the problem solving process (study of structural features and specifics of the problem being solved);
- an informed choice based on the knowledge of the teacher and the student;

- narrowing the search for a solution to the problem by means of ARIZ (an algorithm for solving inventive, x tasks) that incorporates the experience of similar solutions.

Therefore, the distinctive features of heuristic learning are variable-probabilistic, unifying, and model character. Since we have defined heuristics by a certain general method of solving problems, we will proceed to the consideration of heuristic teaching methods.

The forecasting method is based on the process of studying phenomena, trends in the development of the forecast of the solution of the problem. Forecasting is a type of cognitive activity of a person aimed at designing forecasts of the development of an object, through retrospective analysis. It should be noted that prognostics has about 4 forecasting groups: on factual information, expert information, forecasting on analogues, forecasting systems.

The method of association involves the activation of semantic connections (metaphors, random concepts). The idea of using the method is based on establishing a connection between previously unrelated and unrelated concepts. The catalog method involves the transfer of knowledge (attribute, property, i.e. characteristic of an object) from one area to another. The method of focal objects involves the transfer of some features to a more improving object.

The method of heuristic questions involves the initiation and organization of activities to increase the productivity of intelligence, where ideas are generated and the most acceptable and effective solution to the problem is chosen.

The method of synectics involves intensive, psychological activation when solving a problem. The method works in a group of 6-8 people of various specialties and interests. Morphological analysis synthesis is characterized by the constructive implementation of a systematic approach to solving the problem. Finally, the algorithm for solving inventive tasks, the method is based on the sequential execution of actions by comparing the ideal and the real in the object.

ARIZ includes nine stages: object analysis, analysis, model development, definition of contradictions and selection of a standard, resource mobilization, use of the information fund elimination of contradictions, application of the received solution, reflection.

Thus, heuristic teaching methods make it possible to overcome the monologue of the pedagogical process and contribute to the disclosure of the student's personal, creative, potential, manifested in creative, cognitive, activity, personality characteristics.

Problems and prospects of heuristics in video games

Few approaches are currently linking the results of heuristic evaluation methods to user experience. Especially in the field of computer games, where the experience is the leading factor, different aspects can be evaluated using heuristics. This discussion of current advanced interaction games leads to the introduction of a set of specific evaluation heuristics applicable to the user experience of these games.

Regarding game heuristic evaluation, the data show to some degree that game heuristics evaluation is overrated in literature as compared to reality. A significantly large amount of sampled companies are not being involved in heuristic evaluation despite its stated benefits in the literature. Various reasons were exposed as to not using heuristics in game evaluation such as the lack of knowledge and experts, lack of time, and lack of suitable heuristics. These findings call for future research on promoting the heuristic evaluation to game companies and providing game heuristics that are clearer and apply in practice.

The majority of the companies which employed game heuristic evaluation developed and used their own heuristic lists. Moreover, the respondents involved in heuristic evaluation considered that specific lists should exist for specific genres and new games. These results indicate a natural practice and an expected attitude towards game heuristics, given that on the other hand, the majority of the practitioners found the existing heuristics unsuitable. However, ideally, heuristics is a means to provide some form of standardization among similar products, thus creating heuristics for

each game or gamification system is neither economically sustainable nor conceptually justified. Perhaps researchers and practitioners together should agree on what constitute good game heuristics, to avoid unnecessary efforts to be spent on creating tailored heuristics that are difficult to use outside their domain. Thus, there is a need for a collaborative effort between researchers and practitioners to develop heuristics that are comprehensive and useful to practitioners, and to increase the awareness, adoption and diffusion of heuristic evaluation practice in game companies. In addition, more case studies and surveys should be conducted to explore the benefits of game and gamification heuristics in development contexts.

Heuristic evaluation in game and gamification development

Game development business has become very competitive. Therefore it is very important to develop games that stand out from the vast amount of other competing game titles. Usability of the game is crucial to ensure game acceptance and subsequently positive players' reviews. Usability is important in gamification development as well as it optimizes player engagement and accomplishment of individual and organizational objectives. Usability is defined as being the extent to which a product, system, or service can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use.

One successful method to ensure good usability is heuristic or expert evaluation, which has been also introduced in the game context and proven to be useful in game development. However, the existing game heuristics are typically criticized for being too general to suit the specificity of different games genres. Thus, the literature has become abundant of various lists of game heuristics from which it is difficult to choose. Moreover, the available game heuristics need further development to improve its comprehensibility and clarity. Furthermore, the conceptualization of game usability represents also an issue in employing heuristic evaluation as the understanding and definition of game usability affects the way heuristic evaluation is conducted, its scope, and the categories in the heuristic list.

Some game usability heuristic lists include only issues of interface, controls, and methods of interaction, while others also add aspects of gameplay and mechanics.

This study investigates through a survey approach the views and practices of the game companies regarding game usability heuristics and the extent to which they utilize heuristic evaluation as a game usability method. The aim is to understand the practice of game heuristic evaluation given the current debate in the literature on what constitutes game usability and effective game usability heuristics. The purpose is also to understand how game heuristics adoption could contribute to the design and development of effective gamification systems that reach their goals to motivate users to utilize them and to change their behavior towards the desired target.

Heuristic evaluation (HE) is a widely used software usability evaluation method in industry because it is considered a discount usability method not requiring a lot of infrastructure, time, and money. It involves a group of experts evaluating a software product guided by a list of so-called heuristics that provide rules of best practice design, first individually and then as a group. A single person can also perform HE, although using more than one evaluators is recommended in order to find more problems.

Heuristics can be used in game development as early as in the concept design phase and prototype phase, where they can help to inspire a creative player experience as well as to identify usability problems before becoming too expensive and difficult to fix. Although most of the generic heuristic lists can be used in analyzing the user interface of a game, these heuristics still fail to address the fundamental game play issues.

Federoff created a set of 40 game usability heuristics that are categorized into game interface, game mechanics, and gameplay heuristics. This is the first set of heuristics targeted to games and many of the subsequent game heuristics build on it. Heuristic list has four categories (gameplay, story, mechanics, and usability heuristics) and has been found to be very helpful in early phases of game design for

evaluating general issues. Further developments of game heuristics targeted specific genres, mobile games, multi-player games, and social media platforms.

Several authors focused on developing general heuristics for games instead of targeting those to specific genres or types of games. Their game usability heuristics are built around a modular framework consisting of three categories: device specific heuristics, gameplay and game story heuristics, and virtual interface heuristics.

Some of the existing game heuristics are of particular relevance in the gamification context, where the educational or instructional aspect, the engagement, and the fact that the users are not necessarily experienced players or gamers are issues of primary concern. Gamification represents the use of game design elements in non-gaming systems to improve user experience and user engagement. Gamification is viewed as having a business or organizational orientation, namely gamification is considered beneficial to an organization or goal (such as education) by creating and maintaining intrinsic motivation of users to achieve desired target behaviors through the play of a well-designed game.

The design of gamification involves introducing game-design elements into the software development of the target system, and it is challenging. Thus, game heuristics can facilitate the development of the gamification system and ensure that the target objectives of the system are fulfilled. A relevant research in this respect, in particular applicable to the educational aspect of gamification, is that by Omar & Jaafar who developed the first set of playability heuristics targeted to educational games. This set of 34 heuristics focuses on five aspects: user interface, playability, content, education, and multimedia. Sweetser, Johnson, and Wyeth created a heuristic list of 165 heuristics for evaluating real-time strategy games based on the gameflow model by Sweetser & Wyeth to achieve detailed gameflow criteria for one game genre. These heuristics have eight categories: control, immersion, concentration, player skills, challenge, clear goals, and social interaction; these could be applicable to gamification design to ensure flow and engagement. Desurvire & Wiberg introduced a set of heuristics focusing on the experience of the

new players, particularly casual gamers. They addressed the game usability and game usability heuristic evaluation through the concept of game approachability, which they define as making games initially more friendly, fun, immersive, and accessible for those players who have the desire to play, yet do not always follow-through to actually playing the game.

In addition to the three sets of heuristics above that are relevant to gamification, Tondello, Kappen, Mekler, Ganaba, and Nacke developed a heuristic list specifically targeted to gamification design. This list distinguishes between three types of heuristics based on the type of motivation the system is targets: intrinsic, extrinsic, and context-dependent motivations.

To understand how practitioners perceive game usability and practice heuristic evaluation, a series of surveys among North American (Canada and the USA) and North European (Denmark, Finland, Ireland, Latvia, Lithuania, Sweden, the UK) companies were conducted. Data were collected as part of a larger, long-term research effort to understand the views of practitioners on usability and their utilization of usability methods in the game development. The present study focuses on data collected during 2012-2016 starting in Finland (2012), then extending to other Northern European countries and further to North America (2016). The survey reached in total 331 companies in North Europe (NE) and 802 companies in North America (NA), whose contact information was mainly available on gamedevmap.com.

The first survey, which was conducted in Finland in 2012, contained mainly open answers and the obtained data were coded and used to form categories that were utilized in the subsequent surveys as multiple-choice answers or multiple answers in close-ended questions.

The questionnaire consisted of 39 questions: 29 close-ended questions and 10 open questions. The questions were in English. Only questions regarding the concept of game usability and heuristic evaluation were analyzed. In addition, relevant background information about the participants was reported.

Usability was regarded as very important in games by both NA companies (average score 4.68 on a 5-point Likert scale) and NE companies (4.69). When asked to rate the importance of different aspects of game usability such as interface, controls, user experience, flow, game mechanics, the top three aspects associated with game usability were user interface, controls, and user experience.

Table 1 shows the ratings by region. No significant differences were found among the ratings in NA and NE by employing the Mann-Whitney test. Many of the respondents do not view game mechanics, fun and gameplay as well as flow and game challenge amongst the most important aspects of game usability.

Table 1 – Aspects of game usability

Aspect of usability	NE companies (N = 45)		NA companies (N = 59)		Total (N = 104)	
	n	%	n	%	n	%
User interface	42	93 %	49	83 %	91	88 %
Controls	41	91 %	51	86 %	92	88 %
User experience	38	84 %	53	90 %	91	88 %
Flow	32	71 %	39	66 %	71	68 %
Challenge	26	58 %	40	68 %	66	63 %
Gameplay	25	56 %	39	66 %	64	62 %
Fun	22	49 %	36	61 %	58	56 %
Mechanics	20	44 %	38	64 %	58	56 %

Table 2 shows the extent of employing heuristic evaluation in the sample of companies conducting usability activities (n = 34 and 50 in NE and NA, respectively). The proportion of NA companies performing HE (28%) is greater than in NE (18%). Nevertheless, the overall utilization of this approach seems overrated, in reality reaching only 24% of the sampled companies that perform usability activities.

Table 2 – Heuristic evaluation in game development

Game companies	Perform usability activities		Perform game HE	
	n	%	n	%
NE companies	34	76	6	18
NA companies	50	85	14	28
Total	84	81	20	24

Companies not conducting HE have given different reasons for not employing game HE (Table 3). Among them is lack of awareness, perceived time costs or other costs (too time consuming, no time to test new methods, expensive), lack of human resources to perform HE, lack of knowledge or expertise, not perceiving HE as

worthwhile, and perceived unsuitability of the heuristic lists (e.g., too general, not comprehensive, unsuited for own games).

Table 3 – Reasons given for not using heuristic evaluation in game development

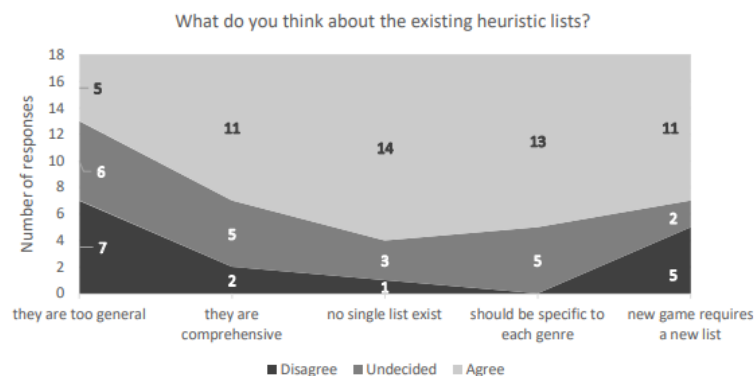
Reason	NE companies <i>N</i> = 34 (<i>n</i>)	NA companies <i>N</i> = 50 (<i>n</i>)	Total <i>N</i> = 84 (<i>n</i>)
Did not know about HE	11	20	31
Did not know how to use HE	3	10	13
Thought that HE is too time consuming	4	5	9
Did not think it is worthwhile	2	5	7
No suitable existing heuristic lists	1	4	5
Not enough people to conduct HE	3	2	5
Seen as too expensive	1	1	2
HE did not produce good results when used	0	1	1
Other	1	3	4

Among the reasons of employing HE, respondents noted that HE is cheap, effective, easy to use, fast to perform, and reduces the number and/or severity of usability problems. Table 4 shows the reasons given for using heuristic evaluation as part of usability evaluation.

Table 4 – Reasons given for using heuristic evaluation in game development

Reason	NE companies <i>N</i> = 6 (<i>n</i>)	NA companies <i>N</i> = 14 (<i>n</i>)	Total <i>N</i> = 20 (<i>n</i>)
HE is an effective way to find problems	3	11	14
Proactive use of HE reduces the number of problems found in later stages of development	4	9	13
HE is easy to use	3	6	9
HE is not too time consuming	4	4	8
HE is a cheap method	3	3	6
Other	1	2	3

When asked to characterize the existing heuristics lists, many respondents using HE characterized them in a positive light. Moreover, the majority of the respondents indicated that there are a lot of lists and that the heuristics should be specific to each game genre or each new game.



The way the game HE was employed varied with respect to company size, to whether the evaluation is done by own employees or by external experts, and to the kind of team involved in HE. The general profile of a team performing HE is one that functions within a large company, with own employees as usability experts, and typically consisting of a single member. In NA, teams of three members and external usability experts were also common. Different roles were involved in HE such as player champion, lead designer, user researcher, programmer, quality assurance specialist. Companies used as game heuristics mainly their own lists, as well as Nielsen's list of heuristics adapted or used together with own list.

This study described the state-of-the-art of heuristic evaluation in game development companies. The aim was to understand the practice of game heuristic evaluation given the current debate in the literature on what constitutes game usability and effective game usability heuristics. The purpose was also to understand how game heuristics adoption could contribute to the design and development of effective gamification systems that reach their goals to motivate users to utilize them and to change their behavior towards the desired target. Though the sample size of surveyed companies is relatively small, the findings provide interesting and valuable insight on how companies understand game usability and apply game heuristics in game development with implications to gamification development.

Game usability was seen mostly associated with user interface, controls, and user experience, which are general aspects of usability not specific to games. The more specific aspects such as flow, challenge and gameplay were also seen related to usability, but they did not reach consensus as to their importance among the developers. Moreover, fun and game mechanics were the least associated with game usability. However, a slight trend could be observed in the North American companies which recognize these attributes as part of game usability to a larger extent.

Regarding game heuristic evaluation, the data show to some degree that game heuristics evaluation is overrated in literature as compared to reality, a significantly

large amount of sampled companies not being involved in heuristic evaluation despite its stated benefits in the literature. Various reasons were exposed as to not using heuristics in game evaluation such as the lack of knowledge and experts, lack of time, and lack of suitable heuristics. These findings call for future research on promoting the heuristic evaluation to game companies and providing game heuristics that are clearer, easier to understand and apply in practice, and providing examples similarly to the ones developed by Nielsen.

The majority of the companies which employed game heuristic evaluation developed and used own heuristic lists, and/or used an adapted Nielsen's list. Moreover, respondents involved in heuristic evaluation considered that specific lists should exist for specific genres and new games. These results indicate a natural practice and an expected attitude towards game heuristics, given that on the other hand, the majority of the practitioners found the existing heuristics unsuitable. However, ideally, heuristics is a means to provide some form of standardization among similar products, thus creating heuristics for each game or gamification system is neither economically sustainable nor conceptually justified. Perhaps researchers and practitioners together should agree on what constitute good game heuristics, to avoid unnecessary efforts to be spent on creating tailored heuristics that are difficult to use outside their domain. Thus, there is a need for a collaborative effort between researchers and practitioners to develop heuristics that are comprehensive and useful to practitioners, and to increase the awareness, adoption and diffusion of heuristic evaluation practice in the game companies. More case studies and surveys should be conducted to explore also the benefits of game and gamification heuristics in development contexts.