

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Образовательная программа «Программная инженерия»

СОГЛАСОВАНО
Приглашенный преподаватель
департамента программной инженерии

_____ Д.П. Архаров
«___» _____ 2024 г.

УТВЕРЖДАЮ
Академический руководитель
образовательной программы
«Программная инженерия»
старший преподаватель департамента
программной инженерии

_____ Н.А. Павлочев
«___» _____ 2024 г.

**ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ ПРОТИВНИКОВ В КОМПЬЮТЕРНОЙ ИГРЕ
ЖАНРА СЛЭШЕР С ЭЛЕМЕНТАМИ РПГ НА UNREAL ENGINE 5**

Пояснительная записка

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.05.03-01 81 01-1-ЛУ

Исполнитель
студент группы БПИ212
_____ / К.А. Ганина /
«___» _____ 2024 г.

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

Москва 2024

УТВЕРЖДЕН
RU.17701729.05.03-01 81 01-1-ЛУ

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ ПРОТИВНИКОВ В КОМПЬЮТЕРНОЙ ИГРЕ
ЖАНРА СЛЭШЕР С ЭЛЕМЕНТАМИ РПГ НА UNREAL ENGINE 5

Пояснительная записка

RU.17701729.05.03-01 81 01-1

Листов 27

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

Москва 2024

АННОТАЦИЯ

Данный программный документ представляет собой пояснительную записку к разработке подсистемы «Искусственный интеллект противников в компьютерной игре жанра слэшер с элементами РППГ на Unreal Engine 5».

В разделе «Введение» содержится наименование разработки, условное обозначение темы разработки, документ, на основании которого ведется разработка, и организация, утвердившая данный документ.

В разделе «Назначение и область применения» указано функциональное и эксплуатационное назначение подсистемы и краткая характеристика области её применения.

В разделе «Технические характеристики» содержатся следующие подразделы: постановка задачи на разработку подсистемы, описание функционирования подсистемы, описание алгоритма работы подсистемы, описание и обоснование выбора метода организации входных и выходных данных, описание и обоснование выбора состава технических и программных средств.

В разделе «Ожидаемые технико-экономические показатели» указана предполагаемая потребность и экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами.

Программный документ разработан в соответствии с требованиями:

1. ГОСТ 19.101-77 Виды программ и программных документов [1].
2. ГОСТ 19.102-77 Стадии разработки [2].
3. ГОСТ 19.103-77 Обозначения программ и программных документов [3].
4. ГОСТ 19.104-78 Основные надписи [4].
5. ГОСТ 19.105-78 Общие требования к программным документам [5].
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [6].
7. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению [8].

Изменения к Пояснительной записке оформляются согласно ГОСТ 19.603-78 [9], ГОСТ 19.604-78 [10].

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

СОДЕРЖАНИЕ

ГЛОССАРИЙ	5
1. ВВЕДЕНИЕ	7
1.1. Наименование разработки	7
1.2. Документы, на основании которых ведется разработка.....	7
2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ	8
2.1. Функциональное назначение.....	8
2.2. Эксплуатационное назначение.....	8
2.3. Краткая характеристика области применения.....	8
3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ	9
3.1. Постановка задачи на разработку подсистемы	9
3.2. Описание алгоритма и функционирования подсистемы	9
3.2.1. Обоснование внедрения Component Pattern	9
3.2.2. Выбор варианта взаимодействия объектов подсистемы.....	10
3.2.3. Выбор оптимизированного подхода считывания информации из окружающей среды	12
3.2.4. Выбор системы иерархии и распределение деревьев поведения у ИИ	12
3.2.5. Разделение обязанностей сущности и контроллера у игрока.....	13
3.3. Описание и обоснование выбора метода организации входных и выходных данных.....	13
3.4. Описание и обоснование выбора состава технических и программных средств	13
4. ОЖИДАЕМЫЕ ТЕХНИКО--ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ	15
4.1. Ориентировочная экономическая эффективность	15
4.2. Предполагаемая потребность.....	15
4.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами	15

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	17
ПРИЛОЖЕНИЕ 1	20
ПРИЛОЖЕНИЕ 2	21
ПРИЛОЖЕНИЕ 3	23
ПРИЛОЖЕНИЕ 4.....	25
ПРИЛОЖЕНИЕ 5.....	27

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ГЛОССАРИЙ

1. **Персональный компьютер (ПК)** – однопользовательская ЭВМ, имеющая эксплуатационные характеристики бытового прибора и универсальные функциональные возможности.
2. **Компьютерная игра** – вид развлекательного программного обеспечения, предназначенного для запуска и использования на персональном компьютере. Обычно включает в себя визуальные и звуковые эффекты, а также интерактивное взаимодействие пользователя.
3. **Искусственный интеллект (ИИ, en-us: AI, Artificial Intelligence)** – технология, которая позволяет компьютерным программам осуществлять поведение, аналогичное человеческому, с целью выполнения определенных задач или симуляции различных ситуаций.
4. **Линейный одиночный слэшер (en-us: linear single slasher) с элементами РПГ** – тип компьютерной игры, в которой игрок управляет персонажем, сражаясь с врагами в режиме реального времени в линейном сюжете, а также включающей элементы развития персонажа, характерные для ролевых игр.
5. **Здоровье (Хп, en-us: HP, Health Points)** – количественный показатель текущего состояния здоровья персонажа в игре, определяющий его способность к продолжению игрового процесса.
6. **Урон (Дамаг, en-us: Damage)** – количество повреждений или потеря здоровья, нанесенных персонажу в результате атаки или других воздействий.
7. **Лечение (Хил, en-us: Heal)** – процесс восстановления здоровья персонажа, обычно с помощью специальных предметов или способностей.
8. **Цель (таргет, en-us: Target)** – объект или персонаж, на который направлено внимание или атака игрока в игровом процессе.
9. **Игровая сессия** – отрезок времени, в течение которого игрок участвует в игре, включая начало, продолжительность и завершение игрового процесса.
10. **Хит-лучи (en-us: Line Trace)** – в контексте Unreal Engine 5.2 представляют собой метод, используемый для определения столкновений или взаимодействий между объектами в виртуальном пространстве игры. Этот метод представляет собой лучевой взгляд из определенной точки в пространстве в определенном направлении и проверяет, пересекается ли он с каким-либо

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

объектом в его пути. Хит-лучи часто используются для определения попадания от оружия, обнаружения препятствий или взаимодействия с объектами окружения.

11. **Капсульная коллизия персонажа (en-us: Character Actor Capsule Collision)** – метод обнаружения столкновений для актеров в Unreal Engine 5.2. Капсульная коллизия представляет собой форму коллизии, которая приближенно соответствует форме актера и обеспечивает точные и эффективные расчеты столкновений в реальном времени. При использовании капсульной коллизии, актер представлен в виде капсулы, и система обнаруживает столкновения между ней и другими объектами в игровом мире.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

1. ВВЕДЕНИЕ

1.1. Наименование разработки

Наименование темы разработки – «Искусственный интеллект противников в компьютерной игре жанра слэшер с элементами РПГ на Unreal Engine 5»

Наименование темы разработки на английском языке – «Opponents' Artificial Intelligence in a Computer Game of the Slasher Genre with RPG Elements on Unreal Engine 5»

1.2. Документы, на основании которых ведется разработка

Основанием для разработки является учебный план подготовки бакалавров по направлению 09.03.04 «Программная инженерия» и утвержденная академическим руководителем тема курсового проекта «Искусственный интеллект противников в компьютерной игре жанра слэшер с элементами РПГ на Unreal Engine 5».

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

2.1. Функциональное назначение

Данная подсистема позволит симитировать интеллект противников в компьютерной игре жанра слэшер, что обеспечит динамическое взаимодействие с игроком, при этом разработка предоставит оптимальное использование алгоритмов, эффективное управление ресурсами и расширяемость кода, что соответствует жанру игры и обеспечивает высокую производительность.

2.2. Эксплуатационное назначение

Разработка предоставляет готовый набор инструментов для имитации поведения противников в компьютерной игре жанра слэшер, включающий обнаружение, преследование и атаку игрока, а также дополнительные взаимодействия в бою. При этом разработка использует component pattern, который унифицирует игровые сущности и позволяет переиспользовать фрагменты кода с общей логикой без дублирования.

В рамках проекта не рассматривается проработка наполнения игровых уровней – только маршруты противников и базовые препятствия для демонстрации реализованных механик. Также не рассматривается дизайн актеров, их анимации, звуковой дизайн и проработка системы заданий игрока.

ИИ будет эксплуатироваться инди-компанией «Gosling Entertainment» в игре «Послушник» и других проектах схожих жанров.

2.3. Краткая характеристика области применения

Проект «Послушник» – линейный одиночный слэшер с элементами РПГ, разработанный для ПК, вдохновленный Souls-like играми и русской мифологией. Игра включает в себя ближний и дальний бой и сюжетных противников.

В контексте курсовой работы, участники проекта «Послушник» выступают в роли заказчиков, и основной задачей является разработка и интеграция искусственного интеллекта для противников в игре.

ИИ разработан для инди-компании «Gosling Entertainment» в рамках проекта игры «Послушник».

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

3.1. Постановка задачи на разработку подсистемы

Разрабатываемая подсистема должна соответствовать требованиям, описанным в настоящем Техническом задании «Искусственный интеллект противников в компьютерной игре жанра слэшер с элементами РПГ на Unreal Engine 5» [7]. Полученная подсистема должна предоставлять инструментарий для имитации игрового поведения противников, а именно:

1. Передвижение и патрулирование текущей локации (лобби) по заданному маршруту.
2. Восприятие объектов окружающей среды, в частности, персонажа-игрока (AI Perception).
3. Переходы между состояниями ИИ через конечные автоматы (Behavior Tree) [15].
4. Поддержка компонента системы нанесения и получения урона, компонента здоровья.
5. Взаимодействие с окружением: навигационная сетка на карте уровня (Navigation Mesh), принятие решений в зависимости от условий среды (Environment Query System) [16].
6. Обработка реакций на события игровой сессии (перемещение игрока между лобби, смерть игрока, атаки нескольких противников).

Также подсистема должна быть интегрирована в компьютерную игру, а именно:

1. Необходимо переработать шаблон игрового персонажа и встроить в него написанные компоненты.
2. Отделить ввод игрока от конкретной сущности персонажа в контроллер.
3. Создать кастомные объекты окружения (Game Instance, Game Mode, Game State), которые наследуют стандартные реализации.

3.2. Описание алгоритма и функционирования подсистемы

3.2.1. Обоснование внедрения Component Pattern

В системе нанесения и получения урона выделяются общие мотивы у игрока и у противников разных типов: факт нанесения урона, логика взаимодействия хит-лучей с актерами, потеря и восстановление очков здоровья, настройки базовых показателей. Чтобы не дублировать объемные части кода в нескольких местах одновременно и не усложнять дальнейшее расширение подсистемы

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

в случае внедрения новых механик, было принято решение придерживаться Component Pattern, который предполагает выделение общей логики внутри отдельного объекта. В инструментarii движка Unreal Engine 5 [11] с помощью Gameplay Framework [12] можно создать сущность типа Actor Component [17], что определяет поведение повторного использования, которое можно добавить к различным типам актеров.

Таким образом были выделены компоненты здоровья, выносливости и энергии, а также компонент системы урона. Помимо классов-компонент был добавлен интерфейс Damageable: каждый класс, что имплементирует данный интерфейс (базовый класс врага, игрок), внутри переопределяет методы в соответствии со своими особенностями. Независимые объекты могут обращаться к актеру через интерфейс Damageable, вне зависимости от того, игрок это или противник.

Рассматривался простой для понимания подход с использованием наследования классов для обработки урона и управления здоровьем, однако он не так гибок и может привести к проблемам с множественным наследованием и с нарушением инкапсуляции. Кроме того, такой подход усложняет внесение изменений и поддержку кода при интеграции в другие проекты.

3.2.2. Выбор варианта взаимодействия объектов подсистемы

В подсистеме ИИ множество сущностей взаимодействуют друг с другом: отдельные экземпляры противников с игроком, с объектами окружения. Необходимо учесть реакцию на события, происходящие во время игровой сессии (например, смерть противников или игрока). Одно из возможных вариантов решения – прописать взаимодействие между объектами, чтобы их синхронизировать. Однако это приводит к проблеме – сильная связность объектов, в таком случае для внесения исправления в логику одного объекта понадобится более комплексный подход, что учтет изменения согласно всем имеющимся связям.

Чтобы избежать проблемы сильной связности, было принято решение использовать Observer pattern и Mediator pattern.

Observer pattern (паттерн Наблюдатель) позволяет объектам подписываться и наблюдать за изменениями в других объектах, неявно связываясь с ними. В этом паттерне выделяются наблюдаемые объекты, которые генерируют события, и наблюдатели, которые реагируют на эти события, обеспечивая слабую связанность между компонентами системы.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Mediator pattern (паттерн Посредник) позволяет уменьшить прямые связи между компонентами системы, вынося управление в отдельный объект-посредник. Компоненты взаимодействуют друг с другом через посредника, что позволяет уменьшить зависимости между ними и повысить гибкость системы. Посредник может координировать обмен сообщениями между объектами, управлять их жизненным циклом или реагировать на определенные события.

В рамках подсистемы для реализации паттернов используются технологии движка – Event Dispatchers [18]. Объекты могут вызывать события и подписываться на них, чтобы реализовать логику реакции.

3.2.2.1. Combat Manager для координации нескольких противников

Необходимо ограничить число противников, которые могут одновременно атаковать игрока. Для этого понадобится синхронизация актеров ИИ, которая была вынесена в отдельного актера, наблюдателя-посредника.

ИИ переходит в состояние атаки, когда получает урон от игрока (damage perception) или когда попадает в его поле зрения (sight perception). Перед вызовом непосредственно метода перехода в нужное состояние, контроллер ИИ обращается к Combat Manager с запросом, может ли текущий экземпляр конкретного типа противника атаковать игрока. Если число уже атакующих ниже заданного порога, то происходит переход в атакующее состояние, иначе – противник отправляется в лист ожидания и пока висит в состоянии исследования.

3.2.2.2. Combat Manager реагирует на телепортацию игрока через Game State

Демонстрация подсистемы происходит в рамках одной карты (уровня), которая содержит 3 комнаты с разным количеством противников, и перемещение между уровнями осуществляется с помощью актера-телепорта. На событие перемещения, которое инициирует Teleport Actor, реагирует Combat Manager: очищает имеющиеся списки активных противников, что могут атаковать персонажа-игрока. Таким образом, имитируется независимость групп противников из разных комнат. Сочетанием паттернов Observer и Mediator выступает кастомный класс Game State, через него происходит взаимодействие независимых объектов.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.2.3. Выбор оптимизированного подхода считывания информации из окружающей среды

Используются наследники классов Game Instance, Game Mode и Game State для эффективного управления логикой игрового процесса и взаимодействия с окружающей средой.

Game Instance обрабатывает логику, связанную с объектами окружения (например, настройкой параметров среды). Game Mode хранит логику перезапуска уровня и базовые значения характеристик (базовый урон игрока и противников разных типов, количество затрачиваемой выносливости на ускорение и пережат и другие параметры, влияющие на процесс игры). Game State предоставляет централизованный способ взаимодействия между независимыми объектами в игре, обеспечивая согласованность данных и состояний между различными компонентами игрового мира.

Основная стратегия оптимизации заключается в том, что не используются события Event Tick для перманентного считывания данных окружающей среды и не тратятся ресурсы на лишние вычислительные операции. Вместо этого применяются Event Dispatchers с имплементацией вышеописанных паттернов, что позволяет эффективно управлять обновлением информации при наступлении определенных событий, минимизируя нагрузку на систему и обеспечивая плавный игровой процесс.

3.2.4. Выбор системы иерархии и распределение деревьев поведения у ИИ

По выбранной иерархии базовый класс актера-противника реализует имплементированные интерфейсы по умолчанию, а классы-наследники могут как переопределить реализацию, так и оставить родительскую. Базовый класс необходим для того, чтобы вынести общий функционал наследников и избежать лишнего дублирования, добавив коду гибкости. Контроллер выполняет другие задачи, отвечающие за восприятие, принятие решений ИИ и переход состояний дерева.

Данные, передаваемые и используемые в деревьях поведения – Blackboard [15], – общие на все виды деревьев: решение обусловлено жанром одиночного слэшера, ведь все противники нападают только на одного игрока, соответственно, значения актера-цели и его локации будут одинаковы.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.2.5. Разделение обязанностей сущности и контроллера у игрока

У игрока – функционал смерти, настройка камеры, действий при нажатии кнопок – перенесено в контроллер. Это позволит проще адаптировать подсистему к игре, в которой можно будет менять активного персонажа: общий функционал описан отдельно, конкретные механики игровых персонажей будут прописаны уже внутри самих классов сущностей.

3.3. Описание и обоснование выбора метода организации входных и выходных данных

Для организации входных данных используется передача информации о состоянии игрового мира, действиях игрока и характеристиках противников в виде структур или ссылок на задействованные объекты. Входные данные предоставляются непосредственно в соответствующие методы подсистемы, обеспечивая точность и своевременное обновление информации. Для организации выходных данных используется информация о состоянии ИИ, реакции на действия игрока и состоянии игрового мира на основе внутренней логики подсистемы.

В контексте разработки игровой подсистемы, данные организованы таким образом, чтобы быть независимыми от конкретного запуска игровой симуляции или игровой сессии в рамках движка. Однако, это не означает, что данные не зависят от симуляции вообще: на многие игровые события будут влиять действия игрока и его нажатия клавиш.

3.4. Описание и обоснование выбора состава технических и программных средств

Для работы необходим следующий состав технических средств.

Требуется устройство на платформе Windows со следующими минимальными характеристиками:

1. 8 GB RAM. Необходимость обусловлена требованиями к производительности и эффективной работе с большим объемом данных в процессе разработки и запуска игрового движка.
2. HDD/SSD с объемом свободной памяти 25 ГБ и выше. Игровой движок Unreal Engine 5.2.1 занимает 20 ГБ – необходим для использования ИИ подсистемы. Минимальный объем обеспечивает достаточное пространство для установки и работы с подсистемой.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Для работы необходима операционная система Windows 10 64-bit и выше – обусловлена совместимостью с большинством современных программных продуктов и игровых движков, включая Unreal Engine 5.2.

Минимальный состав программных средств:

1. DirectX 11-12: Latest drivers – обеспечивает оптимальную поддержку графической обработки и совместимость с современными видеокартами.
2. Epic Games Launcher – необходим для установки игрового движка.
3. Unreal Engine 5.2.1 – необходим для разработки проектов и использования подсистемы ИИ.

Разработка должна быть реализована на игровом движке Unreal Engine 5.2.1 с использованием Gameplay Framework, языком визуального программирования Blueprints [13] и предполагать изменения кода через UE C++ [14].

Gameplay Framework необходим для структурирования проекта, обеспечения готовых игровых сущностей и инструментов, таких как Behavior Tree и Environment Query System (EQS), которые являются ключевыми элементами подсистемы ИИ. Blueprints обеспечивает быструю и гибкую разработку игровой логики и взаимодействия между игровыми объектами. Возможность использования UE C++ предоставляет расширяемость для оптимизации и дополнительной разработки сложных компонентов или функций – может быть полезно в дальнейшем развитии проекта и при необходимости улучшения производительности определенных частей игры.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4. ОЖИДАЕМЫЕ ТЕХНИКО--ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

4.1. Ориентировочная экономическая эффективность

Рынок видеоигр на момент 2021-2022 годов составил 158 млрд. рублей, а рынок разработчиков игр в России в 2022 году сократился на 38 %. Примерно можно оценить, что состояния рынка на конец 2023 и начало 2024 оценивается в 97 млрд рублей – примерно 0.62 от прошлого рынка. С учетом прогнозируемого роста в 7.7 % получаем, что общий рынок, на который может рассчитывать компания «Gosling Entertainment» в будущем, равен 105.5 млрд рублей. Рынок, который доступен сейчас 97 — млрд рублей [19].

Далее были оценены конкуренты проекта «Послушник», а именно студия «Morteshka», которая в 2017 выпустила свою первую игру «The Mooseman» в аналогичной фолк стилистике. С учетом того, что сейчас их игра продается примерно по 200 рублей, при этом в Google Play количество скачиваний перевалило за 500 тыс, рынок, который доступен для игры наших конкурентов, составляет более 100 млн рублей.

Монетизация: buy to play. Компания «Gosling Entertainment» рассчитывает продать 5000 копий игры «Послушник», и если выставить ценник для игры в 200 рублей, то ожидает выручку в 1 000 000 рублей.

4.2. Предполагаемая потребность

Компания «Gosling Entertainment» будет использовать разработку с ИИ системой для создания проектов компьютерных игр в жанрах слэшер или смежных жанрах.

4.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами

Поиск в сети Интернет на момент анализа выявил несколько аналогов как среди игр схожих жанров и сеттингов, так и среди открытого кода с имплементацией ИИ:

1. *The Mooseman* [20] – приключенческая компьютерная игра, разработанная пермской студией «Мортёшка» и выпущенная в 2017 году. Игра основана на мифологии коми, а также других финно-угорских народов, и является попыткой художественной реконструкции мифологических сюжетов.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2. *Yaga* [21] – игра в жанрах экшен и ролевые, разработанная “Breadcrumbs Interactive” и изданная “Versus Evil” в 2021 года. Игра основана на славянском фольклоре и древних языческих верованиях.

3. *UE5 Hack & Slash* [22] – слэшер от третьего лица, разработанная на движке Unreal Engine 5 с ближними и дальними атаками игрока. Проект не включает несколько видов противников и реализацию ИИ через деревья поведения.

4. *HackAndSlashTemplate* [23] – шаблон для создания слэшера, включающий в себя функции боевой системы. Не использует AI инструменты, например, деревья поведения.

Киллер-фичей разработки является возможность переиспользовать отдельные фрагменты кода благодаря расширяемости и обобщенности компонент. Также в отличие от игр-аналогов разработка создается под шаблон слэшера от третьего лица в 3D.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.404-79 Пояснительная записка. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.603-78 Общие правила внесения изменений. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
11. Unreal Engine 5.2 Documentation [Электронный ресурс] / Epic Games Dev Community Unreal Engine 5.2 Documentation. Режим доступа: <https://docs.unrealengine.com/5.2/en-US/>, свободный (дата обращения: 19.12.2023).
12. Gameplay Framework [Электронный ресурс] / Epic Games Dev Community Unreal Engine 5.2 Documentation. Режим доступа: <https://docs.unrealengine.com/5.2/en-US/gameplay-framework-in-unreal-engine/>, свободный (дата обращения: 19.12.2023).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

13. Blueprints Visual Scripting [Электронный ресурс] / Epic Games Dev Community Unreal Engine 5.2 Documentation. Режим доступа: <https://docs.unrealengine.com/5.2/en-US/blueprints-visual-scripting-in-unreal-engine/>, свободный (дата обращения: 19.12.2023).
14. Programming with C++ [Электронный ресурс] / Epic Games Dev Community Unreal Engine 5.2 Documentation. Режим доступа: <https://docs.unrealengine.com/5.2/en-US/programming-with-cplusplus-in-unreal-engine/>, свободный (дата обращения: 19.12.2023).
15. Behavior Tree User Guide [Электронный ресурс] / Epic Games Dev Community Unreal Engine 5.2 Documentation. Режим доступа: <https://docs.unrealengine.com/5.2/en-US/behavior-tree-in-unreal-engine---user-guide/>, свободный (дата обращения: 19.12.2023).
16. Environment Query System [Электронный ресурс] / Epic Games Dev Community Unreal Engine 5.2 Documentation. Режим доступа: <https://docs.unrealengine.com/5.2/en-US/environment-query-system-in-unreal-engine/>, свободный (дата обращения: 19.12.2023).
17. Actor Components [Электронный ресурс] / Epic Games Dev Community Unreal Engine 5.2 Documentation. Режим доступа: https://dev.epicgames.com/documentation/en-us/unreal-engine/adding-components-to-an-actor-in-unreal-engine?application_version=5.2, свободный (дата обращения: 18.03.2024).
18. Event Dispatchers and Delegates [Электронный ресурс] / Epic Games Dev Community Unreal Engine 5.2 Documentation. Режим доступа: https://dev.epicgames.com/documentation/en-us/unreal-engine/event-dispatchers-and-delegates-quick-start-guide-in-unreal-engine?application_version=5.2, свободный (дата обращения: 18.03.2024).
19. Google for Games Reports| Google for Games [Электронный ресурс] / MadCap Software. Режим доступа: <https://games.withgoogle.com/reports/allreports/>, свободный (дата обращения: 03.10.2023).
20. The Mooseman [Электронный ресурс] / Steam. Режим доступа: https://store.steampowered.com/app/574310/The_Mooseman/, свободный (дата обращения: 03.10.2023).
21. Yaga [Электронный ресурс] / Steam. Режим доступа: <https://store.steampowered.com/app/888530/Yaga/>, свободный (дата обращения: 03.10.2023).
22. UE5 Hack & Slash [Электронный ресурс] / Github / Режим доступа: <https://github.com/willroberts/ue5-hack-and-slash?ysclid=lsn9r9dqro753316757>, свободный (дата обращения: 03.10.2023).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

23. HackAndSlashTemplate [Электронный ресурс] / Github. Режим доступа: <https://github.com/Naodo/HackAndSlashTemplate/tree/main>, свободный (дата обращения: 03.10.2023).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ОПИСАНИЕ И НАЗНАЧЕНИЕ КОНТРОЛЛЕРОВ

Класс	Назначение
BP_PlayerController	Реализация функционала персонажа, которым управляет игрок
BP_AIController	Реализация принятия решений, переключения состояний используемых деревьев поведения и логики системы восприятия у ИИ

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ОПИСАНИЕ И НАЗНАЧЕНИЕ ОСНОВНЫХ МЕТОДОВ КОНТРОЛЛЕРОВ

BP_PlayerController

Метод	Назначение
Play Slash Punch	Воспроизведение анимации удара атакой в ближнем бою
Setup Sprint	Увеличение скорости до состояния Sprint
Setup Run	Возвращение скорости к состоянию Running по умолчанию
Roll	Воспроизведение анимации переката
Setup Target on Enemy	Фокус на противника (или отключение фокуса, если тот был активен)
Player Death	Имплементация смерти игрока
Respawn Player	Воскрешение игрока и перезапуск уровня
Cast Spell	Трата энергии на заклинание

BP_AIController

Класс	Назначение
On Possess	Запуск деревьев поведения, настройка значений внутри Blackboard
On Perception Updated	Имплементация восприятия (зрение, слух, реакция на получение урона)
Can Sense Actor	Проверка, может ли чувствовать актера каким-то способом восприятия
Handle Sensed Sight/Sound/Damage	Логика взаимодействия с целью, что удалось прочувствовать

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Класс	Назначение
Check Attack Possibility	Проверка, может ли текущий экземпляр ИИ атаковать цель
Get Current State	Получение текущего состояния экземпляра ИИ
Set State as Passive/ Attacking/ Investigating/ Frozen/Dead	Настройка перехода в нужное состояние

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ОСНОВНЫХ СУЩНОСТЕЙ

Сущности Game System

Сущность	Назначение
BP_GameInstance	Настройка объектов окружения
BP_GameMode	Правила игры, базовые характеристики параметров игрока и противников
BP_GameState	Посредник-наблюдатель при взаимодействии независимых объектов

Сущности Player System

Сущность	Назначение
BP_Player	Класс персонажа-игрока
WB_HealthBar	Виджет отображения очков здоровья, выносливости и энергии игрока
BP_Teleport	Класс для перемещения в следующее лобби
BP_TeleportManager	Класс для настроек разделов ПМИ

Сущности AI System

Сущность	Назначение
BP_PatrolRoute	Кривая линия маршрута патруля ИИ
WB_EnemyHealthBar	Виджет отображения очков здоровья ИИ
BP_CombatManager	Посредник для регулирования перехода в состояние атаки ИИ

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Сущность	Назначение
EQS_FindCover	Настройка среды под поиск укрытия
EQS_FindRangedLocation	Настройка среды под поиск подходящей области для атаки
BP_EnemyBase	Базовый класс противника с общей логикой
BP_Enemy_Melee	Класс противника-дуэлянта
BT_Enemy_Melee	Дерево поведения ИИ-дуэлянта
BP_Enemy_Ranged	Класс противника-метателя
BT_Enemy_Ranged	Дерево поведения ИИ-метателя
BB_EnemyBase	Общее хранилище данных деревьев поведения

Сущности-компоненты

Сущность	Назначение
BPC_HealthComp	Компонент здоровья – тратится при получении урона
BPC_StaminaComp	Компонент выносливости – тратится на перекат и ускорение
BPC_ManaComp	Компонент маны (энергии) – тратится на заклинания атакующие и восстанавливающие очки здоровья
BPC_DamageSystem	Компонент системы нанесения и получения урона

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ОСНОВНЫХ ИНТЕРФЕЙСОВ,
СТРУКТУР, ПЕРЕЧИСЛЕНИЙ**

Интерфейсы

Интерфейс	Назначение
BPI_EnemyAI	Общий функционал ИИ
BPI_Attacker	Общий функционал атакующего
BPI_AttackTarget	Общий функционал цели для всех атакующих
BPI_Damagable	Общий функционал для реализации компонента здоровья и системы урона
BPI_StaminaCapable	Общий функционал для реализации компонента выносливости
BPI_ManaCapable	Общий функционал для реализации компонента энергии

Структуры

Структура	Назначение
S_DamageInfo	Список данных об уроне: значение урона, тип, ответ на урон, флаги для различных условий

Перечисления

Перечисление	Назначение
E_DamageResponse	Ответ на урон (эффект после урона): удар, оглушение, опрокидывание

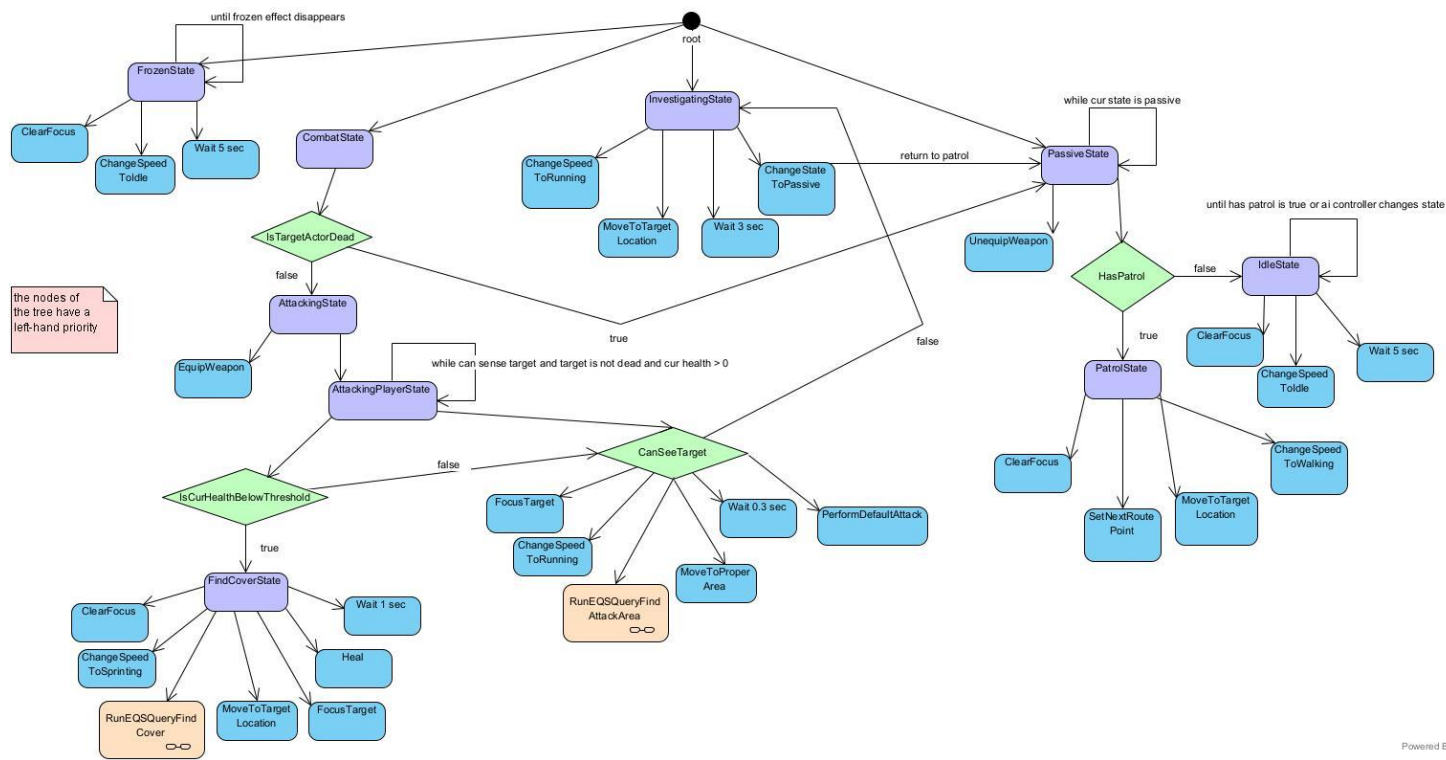
Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Перечисление	Назначение
E_DamageType	Тип урона: ближний бой, атака метательным оружием, урон от окружения
E_EnemySense	Тип восприятия ИИ: зрение, слух, реакция на урон
E_EnemyState	Состояние ИИ: пассивное (патруль), атакующее, оглушенное, исследующее, мертвое
E_MovementSpeed	Скорость актера-персонажа: по умолчанию, неподвижное состояние, ходьба, бег, спринт

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

STATE MACHINE DIAGRAM AI BEHAVIOR TREE

Диаграмма состояний: [версия из облака](#)



Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

[illegible]