|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_***Робототехника и комплексная автоматизация***\_\_\_\_\_\_

КАФЕДРА \_***Системы автоматизированного проектирования (РК-6)***\_

**ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

Студент\_\_\_\_\_\_\_\_\_\_Фёдоров Артемий Владиславович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*фамилия, имя, отчество*

Группа\_\_***РК6-81Б***\_\_\_

Тип практики \_\_\_\_\_\_\_\_\_\_\_***Преддипломная***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Название предприятия\_\_\_\_***НИИ АПП МГТУ им. Н.Э. Баумана***\_\_\_

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_Фёдоров А.В\_\_\_**

*подпись, дата фамилия, и.о.*

Руководитель практики

от кафедры **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_Витюков Ф.А.\_\_**

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***2024 г.***

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ

Заведующий кафедрой *РК6*

\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_ *А.П. Карпенко* \_

« \_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

**ЗАДАНИЕ**

**на прохождение производственной практики**

**\_\_\_\_\_\_*Преддипломная*\_\_\_\_\_\_**

Тип практики

Студент

\_\_\_Фёдоров Артемий Владиславович\_\_\_\_\_\_\_\_ \_\_***4***\_\_ курса группы \_***РК6-81Б***\_

Фамилия Имя Отчество № курса индекс группы

в период с \_***13 мая 2024***\_\_\_ г. по ***26 мая 2024*** г.

*Предприятие:* \_\_\_***НИИ АПП МГТУ им. Н.Э. Баумана***\_\_\_\_\_\_\_\_\_\_

*Подразделение:* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(отдел/сектор/цех)

*Руководитель практики от предприятия (наставник):*

***Киселев Игорь Алексеевич, директор НИИ АПП МГТУ им. Н.Э.Баумана***

(Фамилия Имя Отчество полностью, должность)

*Руководитель практики от кафедры:*

***Витюков Фёдор Андреевич***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия Имя Отчество полностью, должность)

*Задание:*

**1.** *Используя движок Unreal Engine 4 разработать боевую систему ближнего боя.*

**2.** *Для разработанной системы ближнего боя настроить поддержку многопользовательской игры с использованием клиент-серверной архитектуры.*

**3.** *Разработать искусственный интеллект для неигрового персонажа, позволяющий продемонстрировать разработанную боевую систему.*

Дата выдачи задания ***14 мая 2024***г.

Руководитель практики от предприятия  **\_\_\_\_\_\_\_\_\_\_ /\_\_*И.А. Киселев*\_\_/**

Руководитель практики от кафедры  **\_\_\_\_\_\_\_\_\_\_ /\_\_Ф.А.Витюков\_/**

Студент **\_\_\_\_\_\_\_\_\_\_ /\_\_А.В.Фёдоров\_\_/**

СОДЕРЖАНИе

[***ВВЕДЕНИЕ 4***](#_Toc168048283)

[***1. КРАТКИЙ ОТЧЕТ О ВЫПОЛНЕННЫХ РАБОТАХ 5***](#_Toc168048284)

[***1.1. Разработка боевой системы 5***](#_Toc168048285)

[***1.2. Настройка многопользовательской игры 8***](#_Toc168048286)

[***1.3. Разработка искуственного интеллекта 10***](#_Toc168048287)

[***ЗАКЛЮЧЕНИЕ 12***](#_Toc168048288)

[***СПИСОК ЛИТЕРАТУРЫ 13***](#_Toc168048289)

**ВВЕДЕНИЕ**

Движок Unreal Engine является одним из наиболее популярных движков, предназначенных для создания видеоигр. Помимо этого, он используется во многих других сферах, например, в кинематографе и телевидении. Освоение инструментария, предоставляемого разработчиками из Epic Games, является полезным навыком для дальнейшей деятельности в области разработки современных видеоигр.

В рамках выполнения преддипломной практики поставлена цель - разработки боевой системы ближнего боя, настройки игры в многопользовательском режиме, а также создания искусственного интеллекта (ИИ) врага для демонстрации боевой системы. Для достижения цели предполагается выполнение следующих задач:

* Разработка боевой системы с возможностью интеграции с многопользовательским режимом игры.
* Настройка многопользовательского режима игры.
* Создание неигрового персонажа «врага» с настраиваемым искусственным интеллектом, позволяющим продемонстрировать разработанную боевую систему.

А также использование следующих инструментов:

* Движок Unreal Engine 4 и технологии Actor Replication и Remote Procedure Calls для создания возможности игры в многопользовательском режиме.
* Технология Behavior Tree для создания алгоритмов поведения ИИ.

1. **КРАТКИЙ ОТЧЕТ О ВЫПОЛНЕННЫХ РАБОТАХ**
   1. РАЗРАБОТКА БОЕВОЙ СИСТЕМЫ

Для удобства управления состоянием игрового персонажа был разработан список состояний. В каждый момент времени персонаж может находиться только в одном из этих состояний. Перечень состояний выглядит следующим образом:

* Idle (покой) – персонаж не выполняет никаких действий.
* Winding Up (замах) – персонаж замахивается оружием.
* Attacking (атака) – персонаж наносит атаку и проводит трассировку удара.
* Recovering (восстановление) – персонаж восстанавливается после атаки, возвращая оружие в исходное положение.
* Blocking (блок) – персонаж защищается от входящей атаки.
* Stunned by Attack (Оглушение Атакой) – персонаж оглушен входящей атакой и временно не может совершать действий.
* Stunned by Block (Оглушение Блоком) – персонаж оглушен после того как его атака была успешно заблокирована и временно не может совершать действий.

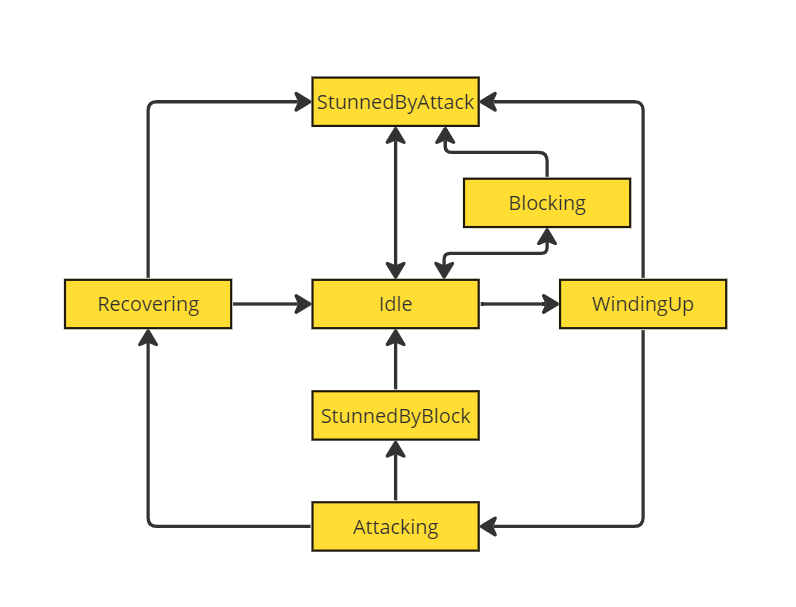


Рисунок 1. Граф состояний персонажа и возможных переходов между ними

Также в список состояний были добавлены переходные состояния, нужные для обеспечения синхронизации между игроками при действиях, которые внезапно меняют состояние персонажа.

* Interrupt Winding Up (прерывание замаха) – персонаж заканчивает замах раньше для нанесения более быстрой но более слабой атаки.
* Block Impact (Успешный блок) – персонаж успешно блокирует входящую атаку.

Листинг 1. Список возможных состояний персонажа.

|  |
| --- |
| *UENUM(BlueprintType)*  *enum class ECharacterState : uint8*  *{*  *Idle UMETA(DisplayName="Idle"),*  *WindingUp UMETA(DisplayName="Winding Up"),*  *InterruptWindingUp UMETA(DisplayName="Interrupt Winding Up"),*  *Attacking UMETA(DisplayName="Attacking"),*  *Recovering UMETA(DisplayName="Recovering"),*  *Blocking UMETA(DisplayName="Blocking"),*  *BlockImpact UMETA(DisplayName="BlockImpact"),*  *StunnedByAttack UMETA(DisplayName="StunnedByAttack"),*  *StunnedByBlock UMETA(DisplayName="StunnedByBlock")*  *};* |

Для обработки действий, которые нужно выполнить при переключении состоянии персонажа, таких как начало анимаций (атаки, оглушения), сброс флагов и сброс таймеров была разработана функция ACustomCharacter::HandleCharacterStateChange.

Листинг 2. Функция ACustomCharacter::HandleCharacterStateChange, обрабатывающая переключение персонажа между разными состояниями.

|  |
| --- |
| *void ACustomCharacter::HandleCharacterStateChange()*  *{*  *switch (CharacterState)*  *{*  *case ECharacterState::Idle:*  *bLockSwordPosition = false;*  *break;*  *case ECharacterState::WindingUp:*  *bWantsToAttack = true;*  *CurrentWindupTime = 0;*  *for (int i = 0; i < AttackAnimations.Num(); ++i)*  *{*  *PlayAnimMontage(AttackAnimations[i], 1, FName("Default"));*  *}*  *break;*  *case ECharacterState::InterruptWindingUp:*  *for (int i = 0; i < AttackAnimations.Num(); ++i)*  *{*  *PlayAnimMontage(AttackAnimations[i], 1, FName("Attack"));*  *}*  *break;*  *case ECharacterState::Attacking:*  *bLockSwordPosition = true;*  *bTraceHasAHit = false;*  *break;*  *case ECharacterState::Recovering:*  *bLockSwordPosition = true;*  *break;*  *case ECharacterState::BlockImpact:*  *for (int i = 0; i < BlockImpactAnimations.Num(); ++i)*  *{*  *PlayAnimMontage(BlockImpactAnimations[i]);*  *}*  *CharacterState=ECharacterState::Blocking;*  *break;*  *case ECharacterState::Blocking:*  *//Start Block Animation*  *bLockSwordPosition = true;*  *CurrentBlockTime = BlockTime;*  *break;*  *case ECharacterState::StunnedByAttack:*  *for (int i = 0; i < StunAnimations.Num(); ++i)*  *{*  *PlayAnimMontage(StunAnimations[i]);*  *}*  *CurrentStunTime = AttackStunTime;*  *break;*  *case ECharacterState::StunnedByBlock:*  *//Start Stun Animation*  *for (int i = 0; i < StunAfterAttackAnimations.Num(); ++i)*  *{*  *PlayAnimMontage(StunAfterAttackAnimations[i]);*  *}*  *StunDirection=FVector2D(1,0);*  *CurrentStunTime = BlockStunTime;*  *break;*  *default:*  *break;*  *}*  *}* |

* 1. Настройка многопользовательской игры

Движок Unreal Engine 4 предоставляет широкий набор инструментов технологий для создания многопользовательских приложений. Технологии, которые были использованы в ходе данный работы:

* **Replication** (Репликация) – технология, позволяющая синхронизировать состояния объектов между сервером и клиентами. UE4 позволяет автоматически реплицировать переменные, функции и события.
* **Remote Procedure Calls** (удаленные вызовы процедур) или **RPC** – технология, позволяющий программе вызвать процедуру (функцию) на удаленном сервере так, как если бы она выполнялась локально.

В рамках данной практики была выбрана модель, в которой сервер обрабатывает всю важную для игрового процесса логику для избежание рассинхронизации игровых процессов разных клиентов. Так, сервер вычисляет, нанесет ли один игрок другому атаку или она будет заблокирована и имеет последнее слово за всеми переходами персонажей между различными состояниями.

С помощью технологии RPC была разработана функция *ACustomCharacter::SetCharacterState,* позволяющая обрабатывать переключение состояний персонажа на сервере либо выполнять удаленный вызов процедуры если функция была вызвана на клиенте.

Листинг 3. Функции ACustomCharacter::SetCharacterState, позволяющая клиенту менять состояние персонажа в синхронизации с сервером.

|  |
| --- |
| *void ACustomCharacter::SetCharacterState(ECharacterState NewState)*  *{*  *CharacterState = NewState;*  *HandleCharacterStateChange();*  *if (!HasAuthority())*  *{*  *Server\_SetCharacterState(NewState);*  *}*  *}* |

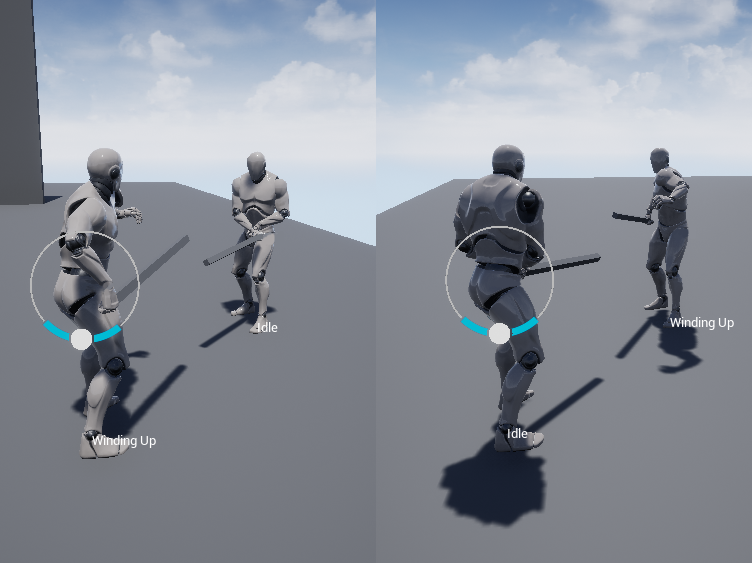
****

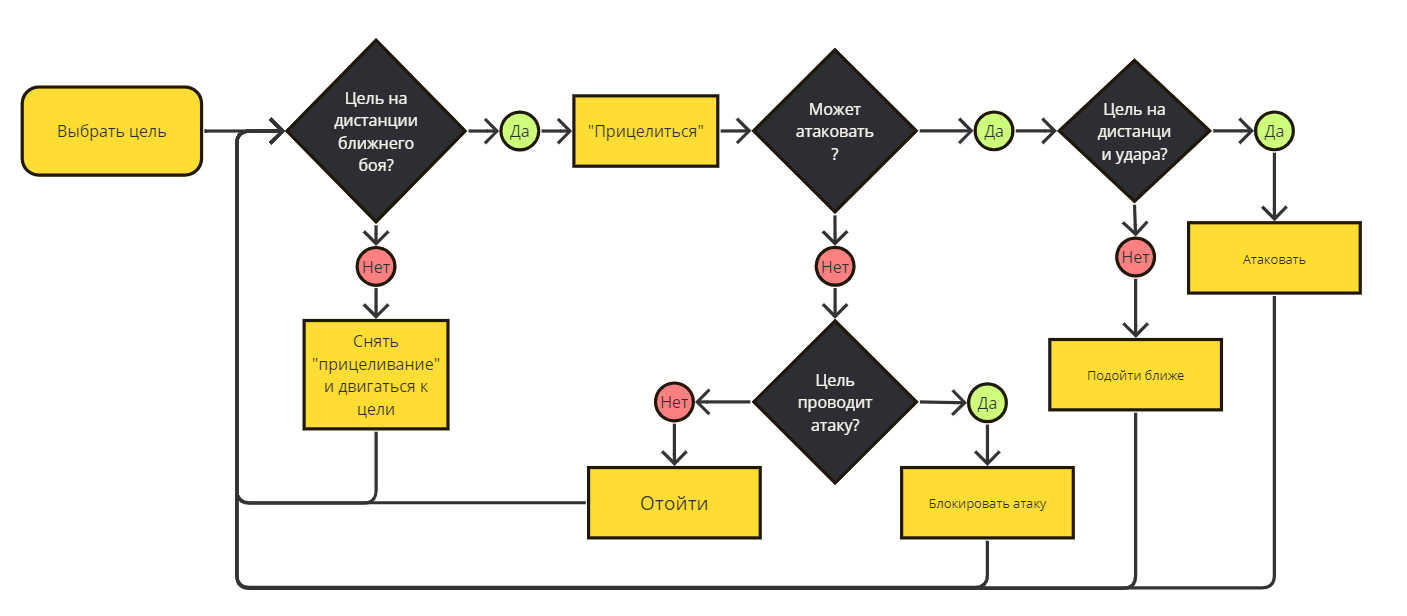
Рисунок 2. Демонстрация работы игры в многопользовательском режиме. Представлены изображения с двух клиентов, подключенных к одному серверу.

* 1. РАЗРАБОТКА Искуственного интеллекта

Unreal Engine 4 предоставляет широкий набор инструментов для создания искусственного интеллекта (ИИ) неигровых персонажей (NPC):

* **Behavior Trees** (Деревья поведения): Структуры данных, позволяющие описывать и организовывать поведение NPC в иерархическом виде
* **Blackboards** (Чёрные доски): Работают совместно c Behavior Trees и служат для хранения и обмена данными между различными частями ИИ системы.
* **Perception System** (Система восприятия): Система восприятия позволяет NPC "чувствовать" окружение, реагировать на звуки, видеть других персонажей и объекты.

Для демонстрации боевой системы в однопользовательском режиме игры был разработан базовый алгоритм поведения неигрового персонажа, позволяющий ему следовать за игроком, защищаться от атак и наносить удары.

Рисунок 3. Логика поведения неигрового персонажа.

С помощью технологии Behavior Trees и визуального программирования Blueprints разработанный алгоритм поведения был перенесен в Unreal Engine 4.

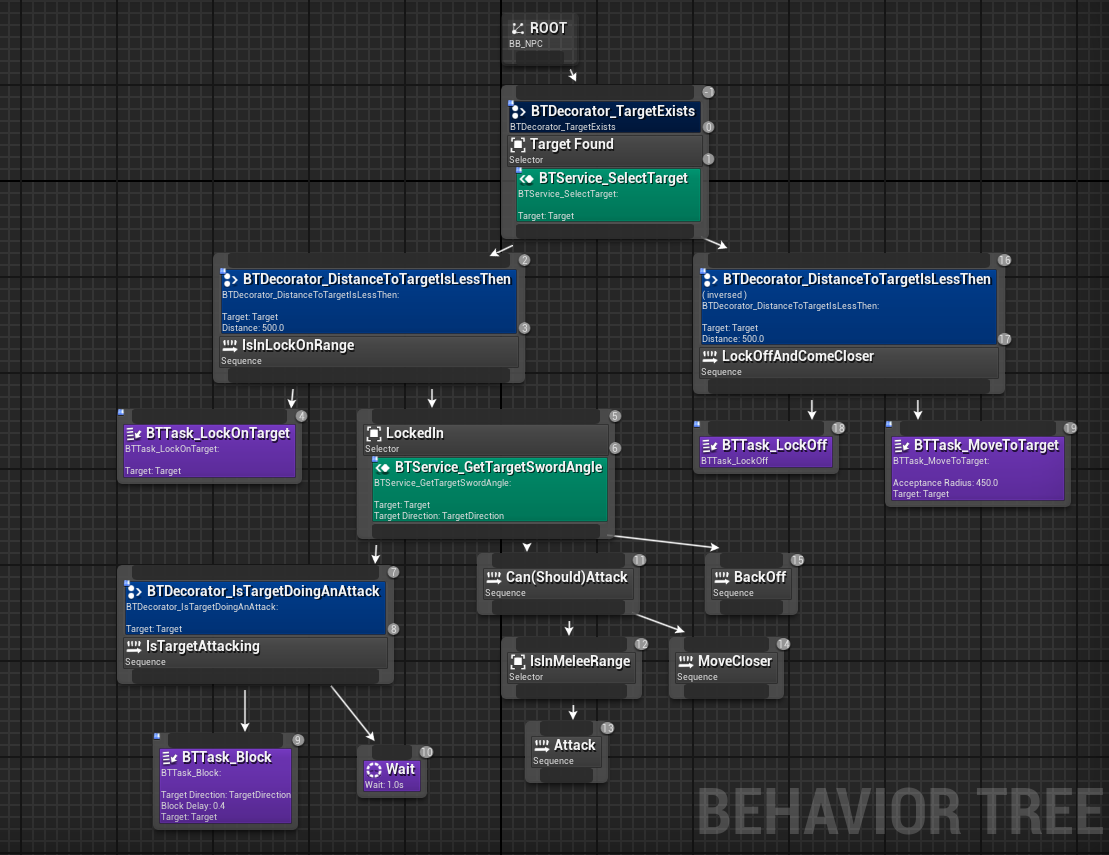


Рисунок 4. Логика поведения ИИ, представленная в виде дерева Behavior Tree.

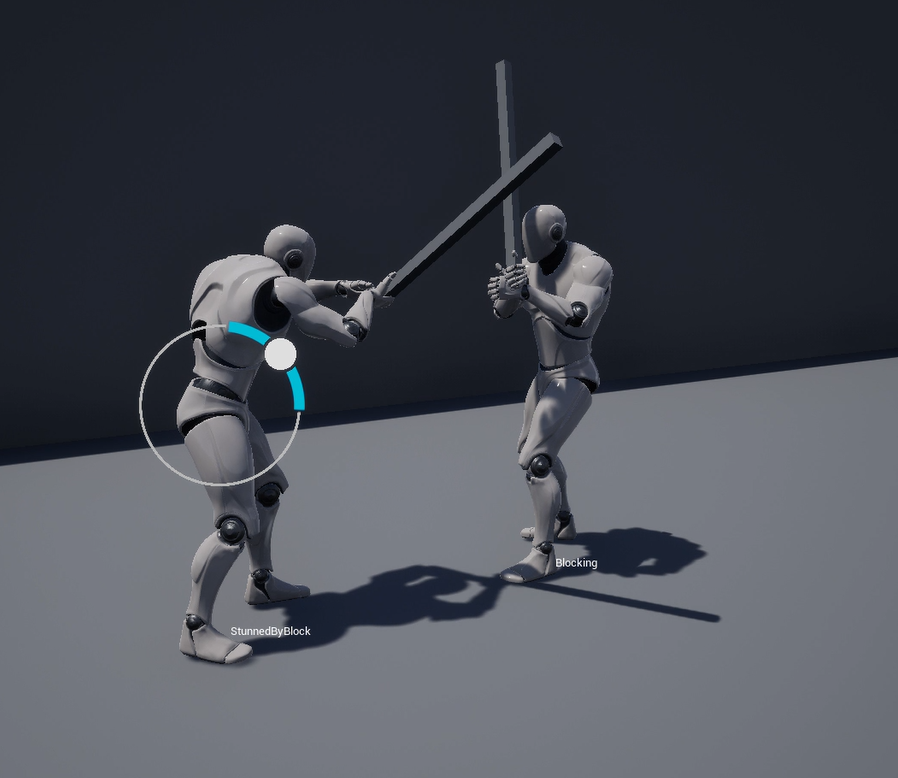


Рисунок 5. Демонстрация работы ИИ. Игрок (слева) наносит удар. Неигровой персонаж (справа) успешно блокирует удар.

**ЗАКЛЮЧЕНИЕ**

В рамках данной работы достигнута цель - изучение технологий разработки многопользовательских приложений и создания искусственного интеллекта неигровых персонажей:

* Разработана боевая система ближнего боя.
* Настроен многопользовательский режима игры.
* Создан неигровой персонаж «враг» с настраиваемым искусственным интеллектом, позволяющим продемонстрировать разработанную боевую систему.

В процессе прохождения практики были получены навыки работы с Unreal Engine 4.

**СПИСОК ЛИТЕРАТУРЫ**

1. Божко А.Н., Жук Д.М., Маничев В.Б. Компьютерная графика. [Электронный ресурс] // Учебное пособие для вузов. − М.: Изд-во МГТУ им. Н. Э. Баумана, 2007. - 389 с., - ISBN 978-5-7038-3015-4, Режим доступа: http://ebooks.bmstu.ru/catalog/55/book1141.html. Дата обращения: 10.02.2024.

2. Unreal Engine 4 Documentation // Unreal Engine Documentation URL: https://docs.unrealengine.com/. Дата обращения: 07.04.2024.

3. Animating Characters and Objects // Unreal Engine Documentation URL: https://dev.epicgames.com/documentation/en-us/unreal-engine/animating-characters-and-objects-in-unreal-engine?application\_version=5.2. Дата обращения: 07.04.2024.

4. Animation & Rigging – Blender Manual // Blender Manual URL: https://docs.blender.org/manual/en/latest/animation/index.html. Дата обращения: 18.03.2024.

5. Modeling – Blender Manual // Blender Manual URL: https://docs.blender.org/manual/en/latest/modeling/index.html. Дата обращения: 18.02.2022.

6. Programming Quick Start // Unreal Engine Documentation URL: <https://docs.unrealengine.com/5.0/en-US/unreal-engine-cpp-quick-start/>. Дата обращения: 29.12.2024.

7. Real-Time Character Animation Techniques // Image Synthesis Group Trinity College Dublin. URL: https://publications.scss.tcd.ie/tech-reports/reports.00/TCD-CS-2000-06.pdf Дата обращения: 05.03.2024.

8. An Indie Approach To Procedural Animation // Wolfire Games. URL: https://gdcvault.com/play/1020049/Animation-Bootcamp-An-Indie-Approach Дата обращения: 05.03.2024