

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по производственной практике
Тема: Разработка демо версии игры в жанре Tower Defense.

Студент гр. 1304

Шаврин А.П.

Руководитель

Брюханов К.В.

Санкт-Петербург

2024

ЗАДАНИЕ НА ПРОИЗВОДСТВЕННУЮ ПРАКТИКУ

Студент Шаврин А.П.

Группа 1304

Тема практики: Разработка демо версии игры в жанре Tower Defense.

Задание на практику:

Нужно создать демо версию игры в жанре Tower Defense с использованием игрового движка Unreal Engine 5. Для этого необходимо реализовать базовую функциональность игры, связанную с вражескими и дружественными не игровыми персонажами (NPC).

Сроки прохождения практики: 26.06.2024 – 09.07.2024

Дата сдачи отчета: 09.07.2024

Дата защиты отчета: 09.07.2024

Студент

Шаврин А.П.

Руководитель

Брюханов К.В.

АННОТАЦИЯ

В ходе практики создается игра в жанре Tower Defense. Для разработки используется игровой движок Unreal Engine 5. Игра разрабатывается без привязки к платформе. Демо версия игры состоит из трех основных частей – башни, вражеские и дружелюбные NPC, а также интерфейс меню. В ходе работы основная часть поставленных задач была связана с разработкой вражеских и дружелюбных NPC.

SUMMARY

During practice, a tower defense game is created. The game engine Unreal Engine 5 is used for development. The game is developed without reference to the platform. The demo version of the game consists of three main parts - towers, enemy and friendly NPC, and a menu interface. During the work, the main part of the assigned tasks was related to the development of enemy and friendly NPC.

СОДЕРЖАНИЕ

	Введение	5
1.	Формальное описание задач	6
2.	Используемые технологии	7
3.	Решение поставленных задач	8
3.1.	Создание структуры классов NPC	8
3.2.	Создание внешнего отображения NPC	8
3.3.	Создание анимации отображения NPC	8
3.4.	Движение вражеского NPC по траектории	9
3.5.	Смерть вражеского NPC	9
3.6.	Нанесение вражескими NPC урона на игровом уровне	10
3.7.	Получение вражескими NPC урона от башен	10
3.8.	Реализация поведения дружественного NPC	10
3.9.	Реализация боя между вражеским и дружественным NPC	11
	Заключение	12
	Приложение А. Отзыв руководителя	13

ВВЕДЕНИЕ

Цель практики – создание структуры и основной функциональности вражеских и дружественных NPC, для демо версии игры в жанре Tower Defense. Также в ходе практики происходит обучения принципам работы с Unreal Engine 5 и применение этих знаний.

Список задач для достижения поставленной цели:

- Создание структуры классов NPC (с различными типами)
- Создание внешнего отображения NPC
- Связь NPC с глобальными объектами уровня
- Реализация поведения вражеского NPC
- Реализация поведения дружественного NPC

1. ФОРМАЛЬНОЕ ОПИСАНИЕ ЗАДАЧ

В ходе работы были поставлены и реализованы следующие задачи:

1. Создание структуры классов NPC
2. Создание внешнего отображения NPC
3. Создание анимации отображения NPC
4. Движение вражеского NPC по траектории
5. Смерть вражеского NPC
6. Нанесение вражескими NPC урона на игровом уровне
7. Получение вражескими NPC урона от башен
8. Реализация поведения дружественного NPC
9. Реализация боя между вражеским и дружественным NPC

2. ИСПОЛЬЗОВАННЫЕ ТЕХНОЛОГИИ

В ходе работы над проектом использовались следующие инструменты и технологии:

1. Игровой движок Unreal Engine 5 как инструмент для создания игры.
2. Язык программирования C++ для реализации требовательных к быстродействию частей игры.
3. Библиотеки языка C++, предназначенные специально для работы с игровым движком Unreal Engine.
4. Notion в качестве инструмента для отслеживания и обновления статуса задач.
5. Gitlab в качестве системы контроля версий.

3. РЕШЕНИЕ ПОСТАВЛЕННЫХ ЗАДАЧ

3.1. Создание структуры классов NPC

Был создан базовый C++ класс NPC с именем AMob, унаследованный от класса APaperZDCharacter предоставляемого плагином PaperZD для UE5 для реализации 2D персонажей. В классе прописан присущий всем NPC функционал такой как жизни, скорость, направление и т.д.

От класса AMob наследуется два C++ класса:

- AEnemyMob, в котором будет прописан функционал, необходимый для вражеских NPC.
- AGuardianMob, в котором будет прописан функционал, необходимый для дружественных NPC.

От C++ классов AEnemyMob и AGuardianMob наследуются Blueprint классы BP_EnemyMob и BP_GuardianMob соответственно, для реализации визуального отображения и другой логики, использующий Blueprint инструменты UE5.

3.2. Создание внешнего отображения NPC

Был скачан tile set для внешнего отображения NPC, который после был преобразован в объект текстуры и разбит на спрайты. После спрайты были объединены в папки по назначению анимаций (направление, состояние NPC). После этого на основе разбитых спрайтов были созданы объекты Paper Flipbook, используемые для анимации состояний NPC.

После полученные Flipbook с состоянием покоя были привязаны к спрайтам классов BP_EnemyMob и BP_GuardianMob.

3.3. Создание анимации отображения NPC

На основе полученных объектов Paper Flipbook созданы объекты Animation Source предоставляемые плагином PaperZD для реализации конечной объединенной анимации для каждого NPC и всех его состояний. После на основе полученных объектов Animation Source созданы Animation Blueprint классы для каждого типа NPC, предоставляемые плагином PaperZD.

После чего полученные классы были привязаны к BP_EnemyMob и BP_GuardianMob. В созданных классах прописана вся необходимая логика проигрывания и переключения анимаций для соответствующих классов NPC.

3.4. Движение вражеского NPC по траектории

Для движения вражеского NPC по траектории используется логика движения по графу. Для этого на карте размещаются объекты TargetPoint, предоставляемые UE5. Каждый объект размещается в начале и конце тропинки, по которой должен идти NPC. Тем самым размещенные объекты служат узлами ориентированного графа, по которому будет перемещаться вражеский NPC.

Был создан C++ класс URoadNode от стандартного UE5 класса UObject. В данном классе прописана вся необходимая логика для узла ориентированного графа: имя, ссылка на родительский и дочерний узел, расположение узла на карте и т.д.

Был создан C++ класс URoadGraph от стандартного UE5 класса UObject. В данном классе прописана вся необходимая логика для ориентированного графа: словарь узлов, ссылка на узел финиша и массив ссылок на узлы начала движения (если старт будет не один). Также в графе прописана логика инициализации графа с получением всех объектов TargetPoint с карты для установки расположения узлов.

Таким образом при установке вражеского игрока на поле и передаче ему узла начала, он может гарантированно дойти до финального узла.

Для движения вражеского NPC и возможности расширения его поведения использовалась такая структура как Behavior tree, предоставляемая UE5 для реализации поведения NPC. В ней был прописан весь необходимый функционал для передвижения NPC от узла к узлу и переключению между узлами.

3.5. Смерть вражеского NPC

Смерть NPC была реализована внутри BP_EnemyMob класса, в котором каждый тик проверяется состояние здоровья NPC и запускается определенный алгоритм действий в случае выполнения данного условия. В случае смерти NPC

проигрывается анимация смерти прописанная в анимационном Blueprint, а после NPC удаляется с игрового поля, а на его месте появляется объект класса BP_Money. В данном случае был создан BP_Money класс с анимацией для него, который существует на поле несколько секунд, а после уничтожается и начисляет глобальному классу уровня определенное количество монет.

3.6. Нанесение вражескими NPC урона на игровом уровне

В созданном для вражеского NPC дереве поведений прописана необходимая логика, в которой по достижению конечного узла графа (окончании пути) происходит удаление NPC с поля, а у глобального класса уровня отнимается одна жизнь.

3.7. Получение вражескими NPC урона от башен

Для того что бы вражеский NPC мог получать урон от других объектов реализована функция нанесения урона NPC внутри класса BP_EnemyMob. Данная функция принимает на вход количество урона, которое необходимо нанести, и отнимает его от жизней NPC. Данная функция не реализована в родительском классе с целью расширения дальнейшего функционала игры.

3.8. Реализация поведения дружественного NPC

В демо версии дружественные NPC создаются башней и управляются игроком через башню. Данные NPC должны ожидать на точке появления вражеских NPC, замечать их, вступать в бой и изменять свою позицию по переданной информации через башню от игрока.

Для решения данной задачи для класса BP_GuardianMob также был создан объект Behavior tree для реализации поведения NPC. Внутри него прописана логика движения к определенной точке, поскольку через башню может меняться изначальное положение. Также для класса BP_GuardianMob подключен AI_Perception – инструмент предоставляемый UE5 для реализации сенсоров NPC таких как зрение, слух и т.д. Используя систему событий предоставляемую UE5 реализовано замечание вражеских NPC, запоминание и очистка целевого врага, которого необходимо атаковать. Атака реализована в дереве поведений в два этапа: дойти до врага, вступить в бой.

3.9 Реализация боя между вражеским и дружественным NPC

Бой между дружественным и вражеским NPC инициируется дружественным NPC. Когда объект BP_GuardianMob подошел к BP_EnemyMob и начал атаку, дружеский NPC запускает бой у BP_EnemyMob. Оба NPC сражаются по определенной задаче, прописанной внутри деревьев поведений каждого врага. Бой осуществляется периодическим нанесением урона противнику. Бой длится до тех пор, пока один из NPC не будет уничтожен. Если первым погибает дружественный NPC, то вражеский NPC продолжает двигаться по траектории, если же первым погибает вражеский NPC, то дружественный либо замечает нового врага и начинает новую атаку, либо возвращается на свою точку.

Таким образом, реализованная функциональность, в совокупности с другими частями приложения, позволяет выстроить базовый геймплей и создать техническую демо версию игры.

ЗАКЛЮЧЕНИЕ

В ходе производственной практики изучен движок для создания игр Unreal Engine 5, а также получен опыт разработки игр на данном движке.

Результатом работы является реализованная структура и базовая функциональность вражеских и дружеских NPC для демо версии игры в жанре Tower Defense. Для этого были выполнены все поставленные задачи.

Полученный результат полностью соответствует поставленной цели.

ПРИЛОЖЕНИЕ А

ОТЗЫВ РУКОВОДИТЕЛЯ

Отзыв

о прохождении производственной практики

Шаврин Алексей Павлович, студент группы 1304, третьего курса бакалавриата Санкт-Петербургского электротехнического университета “ЛЭТИ” им В.И. Ульянова, проходил практику в ООО “ДЕУСОПС” в должности разработчик на Unreal engine 5 с 26.06.2024 по 09.07.2024.

В течении практики обучающийся Шаврин Алексей Павлович выполнял разработку демо версии игры в жанре tower defense на Unreal engine 5.

В результате практики обучающийся разработал порученный ему функционал для демо версии игры в жанре tower defense на Unreal engine 5. Получаемую работу обучающийся Шаврин Алексей Павлович выполнял добросовестно, ответственно и в срок.

По результатам работы Шаврин Алексей Павлович производственную практику оцениваю на отлично.

Подпись руководителя практики:

Генеральный директор ООО “ДЕУСОПС”

09.07.2024

Брюханов К. В.

