

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК  
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

## КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «Ігровий додаток Wayward pines»

за спеціальністю 122 «Комп'ютерні науки»,  
освітньо-професійна програма «Інформаційні технології проектування»

**Виконавець роботи:** студент групи ІТ-61/7 Христенко Максим Олександрович

**Кваліфікаційна робота бакалавра  
захищена на засіданні ЕК**

**з оцінкою** \_\_\_\_\_ «\_\_» \_\_\_\_\_ 2020 р.

Науковий керівник

\_\_\_\_\_  
(підпис)

к.т.н., доц., Федотова Н.А.  
(науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії

\_\_\_\_\_  
(підпис)

Шифрін Д. М.  
(науковий ступінь, вчене звання, прізвище та ініціали)

Засвідчую, що у цій дипломній роботі немає  
запозичень з праць інших авторів  
без відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Суми-2020

Сумський державний університет  
Факультет електроніки та інформаційних технологій  
Кафедра комп'ютерних наук  
Секція інформаційних технологій проектування  
Спеціальність 122 «Комп'ютерні науки»  
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ  
Зав. секцією ІТП

\_\_\_\_\_ В. В. Шендрик  
«\_\_\_\_\_» \_\_\_\_\_ 2020 р.

## З А В Д А Н Н Я НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Христенко Максим Олександрович

**1 Тема роботи** Ігровий додаток Wayward pines

керівник роботи Федотова Наталія Анатоліївна, к.т.н., доцент

затверджені наказом по університету від «14» травня 2020 р. № 0576-III

**2 Строк подання студентом роботи** «1» червня 2020 р.

**3 Вхідні дані до роботи** Потреба у створенні гри за власним сюжетом

**4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)** 1) Аналіз предметної області, 2) Моделювання та проектування, 3) Програмна реалізація додатку.

**5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)** 1) Діаграми сценаріїв використання, 2) IDEF0 ігрової системи, 3) Декомпозиція діаграми ігрової системи IDEF0, 4) Діаграма класів аналізу 5) Діаграма послідовності, 6) Діаграма станів ігрової системи, 7) Діаграма діяльності компоненту ігрової системи 8) Діаграма компонентів

**6. Консультанти розділів роботи:**

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

**7. Дата видачі завдання** 01.10.2019

## КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Планування робіт	До 06.03.20	
2	Розробка технічного завдання	До 15.03.20	
3	Проведення аналізу предметної області	До 22.03.20	
4	Проведення структурно – функціонального моделювання	До 10.04.20	
5	Розробка додатку	До 11.05.20	
6	Тестування додатку	До 17.05.20	
7	Оформлення та здача пояснювальної записки та файлів розробленого проекту	До 01.06.20	

Студент \_\_\_\_\_  
(підпис)

Христенко М.О.

Керівник роботи \_\_\_\_\_  
(підпис)

к.т.н., доц. Федотова Н.А.

## РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра: «Ігровий додаток Wayward pines ».

Пояснювальна записка складається зі вступу, трьох розділів, висновків, списку використаних джерел із десяти найменувань, двох додатків. Загальний обсяг роботи – 72 сторінка, у тому числі 55 сторінок основного тексту, 2 сторінки списку використаних джерел, 15 сторінок додатків.

Кваліфікаційну роботу бакалавра присвячено розробці ігрового додатку Wayward pines В роботі проведено аналіз існуючих додатків проведено моделювання та проектування ігрової системи. У роботі описано хід розробки ігрового додатку та варіанти використання для надання інформації про те як запустити та почати гру.

Практичне значення роботи полягає у розробці таких частин як:

- Створення вимог до гри;
- Створення сюжету для надання індивідуальності гри;
- Створення 3Д моделі головного персонажу та моделей оточення;
- Реалізація ігрових рівнів, а також створення взаємодії між користувачем та елементами оточення;

Результатом проведеної роботи є створений за допомогою рушія «Unreal Engine 4» ігровий додаток.

Ключові слова: 3D модель, Unreal Engine 4, матеріал, текстура, візуалізація, анімація, .

## ЗМІСТ

ВСТУП.....	6
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Огляд останніх досліджень у сфері створення ігор.....	8
1.2 Аналіз ігрових додатків – аналогів.....	9
1.3 Постановка задачі.....	14
2. МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ.....	16
2.1 Моделювання ігрової системи.....	16
2.2 Проектування ігрової системи.....	20
3. ПРОГРАМНА РЕАЛІЗАЦІЯ ДОДАТКУ.....	23
3.1 Програмна реалізація.....	23
3.2 Використання додатку.....	49
ВИСНОВОК.....	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	55
ДОДАТОК А.....	57
ДОДАТОК Б.....	65

## ВСТУП

У останні роки, з розвитком комп'ютерної техніки, ключовою стала індустрія комп'ютерних ігор, яка з появою першого ПК від компанії ІВМ у 1981 розвивалися й покращувалися, та з гоном призвела до розвитку індустрії комп'ютерних ігор.

З різноманітністю комп'ютерних ігор та відмінністю між механіками жанрів та технологіями створення, перед розробниками поставав вибір, або використовувати ігровий рушій з відкритого доступу, та підводити ігрові механіки під його можливості. Або ж поступово створювати свій рушій для забезпечення всіх потреб команди розробників, та уникнення у процесі створення гри таких моментів, що рушій не зможе правильно відображати та підтримувати деякі механіки, що були б потрібні для реалізації гри та надання їй індивідуальності.

Розробникам потрібно буде використовувати так звані «костилі», тобто вигадувати спосіб при якому рушій буде виконувати ту ж саму задачу, але при цьому вона буде використовувати певну послідовність дій, які зв'язані між собою, та будуть імітувати ігрову ситуацію яку запланували розробники у сюжеті гри.

Актуальністю даної роботи є те, що з розвитком сучасних ігрових рушіїв та легкого доступу до їх отримання, перед нами постає можливість створювати ігри різного жанру. Комп'ютерні ігри поділяються за жанрами: екшени, шутери, аркади, спортивні, квести, рольові (RPG/MMORPG), перегони, симулятори, логічні, пригоди, стратегії та ін.

Кожен із цих жанрів має свої особливості, та потребує унікальних знань для їх створення. В свою чергу ігри являють собою дуже складні системи, які розробляються десятками або навіть сотнями професіоналів в галузях програмування, дизайну, анімацій.

Предметом дослідження є створення комп'ютерної гри «Wayward pines».

Об'єктом дослідження є можливість створення комп'ютерних ігор бази ігрового рушія Unreal Engine 4.

Метою даної роботи є розробка комп'ютерної гри «Wayward pines» за допомогою ігрового рушія Unreal Engine 4.

На час виконання дипломної роботи були виділені такі основні завдання як:

- Написання вимог до гри;
- Створення повноцінного сюжету;
- Створення 3Д моделі головного персонажу та моделей оточення;
- Реалізація ігрових рівнів (Сцен);
- Створення взаємодії між користувачем та елементами оточення;

# 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Огляд останніх досліджень у сфері створення ігор

За останні роки індустрія ігор як і розвиток персональних комп'ютерів не стоїть на місці. З покращенням так званого комп'ютерного заліза аналогічно покращилася й графіка в комп'ютерних іграх, створюється все більше додатків з відкритою масштабною картою, де розробники все частіше піклуються про те, щоб не загаяти гравця в певні рамки. В свою чергу це призвело до створення ігор з варіативним вибором реплік або вчинків, які давали б можливість повноцінної зміни фіналу гри.[4]

Таким чином, розробники дають гравцям уявити себе на місці головного героя та можливість обирати розвиток подій та діалогів між неігровими персонажами, та поринути у світ гри з головою. За останні роки лише декілька ігор на думку світової спільноти та видатного ігрового сайту IGN присвоїло звання «ігор, які змінили індустрію» таким іграм як: The Witcher 3 Wild Hunt, World of Warcraft, Devil May Cry, Street Fighter та серія ігор Silent Hill[6].

Щоб реалізувати зазначені вище механіки потрібні сучасні рішення в сфері ігрових рушіїв. Майже кожна велика студія з розробки ігор наймає програмістів щоб створити собі ігровий рушій під власні цілі. Тому у наш час для створення ігор існує велика кількість рушіїв від різноманітних компаній, але не всі вони представлені у вільному доступі та простий користувач не зможе використовувати їх у власних цілях [5].

На сам перед лише декілька компаній таких як Epic games, Unity Technologies, та Crytek надають свої ігрові рушії Unreal Engine, Unity 5, CryEngine для вільного використання та навчання.

Тому з вище вказаного, ігрова індустрія не стоїть на одному місці, постійно розвивається з'являються нові засоби для досягнення поставленої мети, створення



ігор. А можливість доступу хоча б до невеликої кількості безкоштовних рушіїв дає змогу простим користувачам відчувати себе автором та створити додаток своєї мрії.

## 1.2 Аналіз ігрових додатків – аналогів

Так як вже було зазначено ігрова індустрія розвивається, майже кожного дня. Існує велика кількість додатків які істотно відрізняються між собою, тому головним критерієм порівняння буде жанр гри «Survival horror»[3], будуть розглянуті та порівняні ігри даного жанру розроблені на різних рушіях та оцінені якість графіки та простота розуміння геймплею. Буде розглянуто мови програмування на яких були створені додатки.

У якості аналогів будуть виступати такі ігри як:

- Scare на базі рушія CryEngine;
- Ridge на базі рушія Unreal Engine;
- The story of Henry Bishop на базі рушія Unity;

Усі приведені вище ігри це проекти створені однією людиною, ігрові додатки які по мірі проходження гри розкривають перед гравцем цікаву історію. Отже наступним кроком стане поступовий опис сюжету ігор, приведені рисунки на яких буде представлена графіка та деякі елементи геймплею.

Після того як усі додатки будуть розглянуті та порівняні, буде зазначено основну інформацію, та технічні характеристики ігор. Після не великої ігрової сесії у всі вище наведені ігри було отримано дані для формування рецензії для того щоб описати кожен з додатків.

Комп'ютерна гра Scare – це хоррор від першої особи у якому гравцю дає можливість опинитися у ролі людини яка потрапила у авіа катастрофу. Зав'язка сюжету починається з того, що головний герой приходить до тям вже на землі

після катастрофи, без розуміння - що коїться і де він, головному персонажу потрібно знайти допомогу у місті, яке знаходиться неподалеку від нього, але там він нікого не знаходить, окрім рації, по котрій з ним зв'язується якийсь чоловік, що опинився у заручниках у цьому місті та прохає, щоб йому допомогли[7].



Рисунок 1.1 – Представлення гри Scare

Дана гра розроблена за допомогою рушія CryEngine на мові програмування C++, та Flowgraph який представляє собою графічну мову програмування базовану на мові C++, що слугує для написання логіки гри.

Рекомендовані системні вимоги для запуску даної гри досить не великі це:

- Процесор: Intel Core2 Quad Q9450 / AMD Phenom II X4 940;
- ОЗУ: 2 GB (Windows XP) / 3 GB (Windows Vista / Windows 7);
- Відеокарта: NVIDIA GeForce GT 440 с 1024 MB пам'яті.

Дана гра повністю безкоштовна для того щоб отримати до неї доступ необхідно зареєструватися у CryDev.Net.

Комп'ютерна гра Ridge – це хоррор від першої особи за основу якої взята легенда про місцевість під назвою «Медведицька гряда». Дана місцевість

розташована у Волгоградській області, місцеві жителі якої постійно бачать речі, які неможливо описати з наукової точки зору. Головними особливостями є відкрита ігрова мапа з використанням реальної місцевості, зміна погодних умов та часу доби.[9]



Рисунок 1.2 – Представлення гри Ridge

Дана гра розроблена за допомогою рушія Unreal Engine 4 на мові програмування на C++ з використанням Blueprint як графічної мови програмування що базується на мові C++ для написання логіки гри.

Рекомендовані системні вимоги для запуску даної гри досить не великі це:

- Процесор: Intel Core i5;
- ОЗУ: 4 GB;
- Відеокарта: GeForce GTX 770;
- ОС: Windows 10.

Дана гра розповсюджується за допомогою сервісу Steam.

Комп'ютерна гра The story of Henry Bishop – це хоррор від першої особи, у якій поетапно під час проходження гри розповідається історія головного персонажу Генрі Бішопа, який працював ріелтором у 70х роках та намагався

продати маєток. Одного разу йому зателефонував невідомий з проханням продати будинок, після цього йому довелося поїхати інспектувати величезний дім у якому почали відбуватися моторошні речі. Особливостями гри є заплутаний сюжет, атмосфера, та декілька можливих варіантів розвитку сюжету [8].



Рисунок 1.3 – Представлення гри The story of Henry Bishop

Дана гра розроблена за допомогою рушія Unity на мові програмування C#. Рекомендовані системні вимоги для запуску даної гри досить не великі це:

- Процесор: Intel Core i5;
- ОЗУ: 8 GB;
- Відеокарта: 3 GB;
- ОС: Windows 10.

Дана гра розповсюджується за допомогою сервісу Steam на платній основі.

Після того, як ігри були оглянуті та випробувані, результати виведені у таблицю 1.1, у якій зазначені порівняння обраних ігрових додатків за визначеними критеріями для оцінки ігор [1]. У таблиці для легкості розуміння була обрана десятибальна система оцінювання.

Усі дані для порівняння додатків були зібрані з відгуків користувачів сервісу цифрової дистрибуції ігор Steam та перенесені у таблицю 1.1 наведену нижче [8,9].

Таблиця 1.1 - Порівняння обраних додатків

Назва	Основні характеристики			
The story of Henry Bishop	A	Графіка:6/10	Музика: 7/10	Інтерфейс: 7/10
	B	Геймплей:6/10	Режими: 6/10	Складність: 4/10
	C	Сюжет: 7/10	Персонажі:7/10	Новизна: 5/10
Ridge	A	Графіка:10/10	Музика: 6/10	Інтерфейс: 7/10
	B	Геймплей:7/10	Режими: 6/10	Складність: 5/10
	C	Сюжет: 6/10	Персонажі:8/10	Новизна: 5/10
Scare	A	Графіка:8/10	Музика: 8/10	Інтерфейс: 7/10
	B	Геймплей:6/10	Режими: 6/10	Складність: 6/10
	C	Сюжет: 5/10	Персонажі:6/10	Новизна: 5/10

Наступним кроком для отримання результатів оцінювання якості гри виходячи з вище наведених оцінок за допомогою формули наведеної у таблиці 1.2, виводиться фінальна оцінка [2].

Таблиця 1.2 - Формула знаходження кінцевої оцінки гри

Оцінка= (A+B+C)/3	$A=(A_1+A_2+A_3)/3$	A1	A2	A3
	$B=(B_1+B_2+B_3)/3$	B1	B2	B3
	$C=(C_1+C_2+C_3)/3$	C1	C2	C3

Отримана нами оцінка складається з трьох основних критеріїв оцінювання та декількох жанрових складових, у даному випадку це жанр хоррор. Фінальна оцінка наведена у таблиці 1.3.

Таблиця 1.3 Фінальна оцінка для кожної з ігор

Назва	A	B	C	Оцінка (max 10)
The story of Henry Bishop	6.66	5.33	6.33	6.1/10
Ridge	7.66	6	6.33	6.6/10
Scare	7.66	6	5.33	6.3/10

Отже всі вище наведені ігрові додатки приведені в одному жанрі хоррор, але вони істотно різняться один від одного, як технологіями реалізації за допомогою різних ігрових рушіїв так і за мовою програмування, та на сам перед якістю графіки та обмеженнями, які встановлені ігровими рушіями. У всіх трьох іграх відрізняється повнота прописаного сюжету та розкриття персонажів, їх дій та по мірі просування далі по сюжету.

### 1.3 Постановка задачі

На основі попереднього аналізу предметної області для досягнення мети проекту у ході виконання дипломної роботи буде розроблено комп'ютерну гру для створення якої будуть виконані наступні кроки.

- Ознайомлення з літературними джерелами та посібниками по створенню ігор на базі рушія Unreal Engine 4;
- Написання вимог до гри;
- Створення повноцінного сюжету;

- Створення 3Д моделі головного персонажу та моделей оточення;
- Реалізація ігрових рівнів (Сцен);
- Створення взаємодії між користувачем та елементами оточення;
- Звукозапис голосу, та звуків оточення для надання відповідної атмосфери;
- Створення файлу типу «.exe» для запуску гри;
- Тестування роботи гри;
- Реліз гри на платформі Windows 7,10.

У ході всіх перелічених вище дій буде отримано повноцінний ігровий додаток, який у свою чергу розкриє можливості ігрового рушія Unreal Engine 4, та покаже його потенціал при створенні ігор, який у подальшому можливо буде використовуватися у навчальних цілях, або для підготовки спеціалістів у сфері створення ігор.

## 2. МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ

### 2.1 Моделювання ігрової системи

Для того щоб наочно показати варіанти використання ігрової системи були розроблені діаграми сценаріїв використання, які у свою чергу наведені нижче на рис. 2.1, де зображена проектована система, яка представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання.

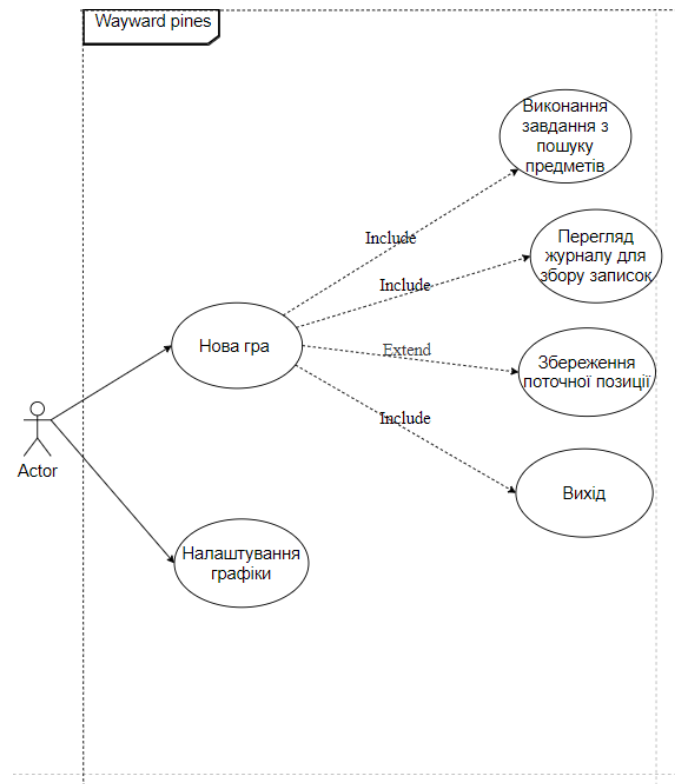


Рис 2.1 – Діаграми сценаріїв використання

Щоб наочно продемонструвати функціональне моделювання ігрової системи була розроблена IDEF0 діаграма, яка демонструє побудову ієрархічної системи діаграм.

Спочатку проводиться опис системи в цілому та її взаємодії з навколишнім світом (контекстна діаграма), після чого проводиться функціональна



декомпозиція – тобто система розбивається на підсистеми, кожна з яких описується окремо (діаграмами декомпозиції), та далі підсистема розбивається на дрібніші частини і цей процес повторюється до того часу поки системою не буде досягнуто потрібної міри деталізації.

Представлення даних діаграм наведені нижче на рис. 2.3 – 2.4.

Функціональне моделювання ігрової системи (в IDEF0)

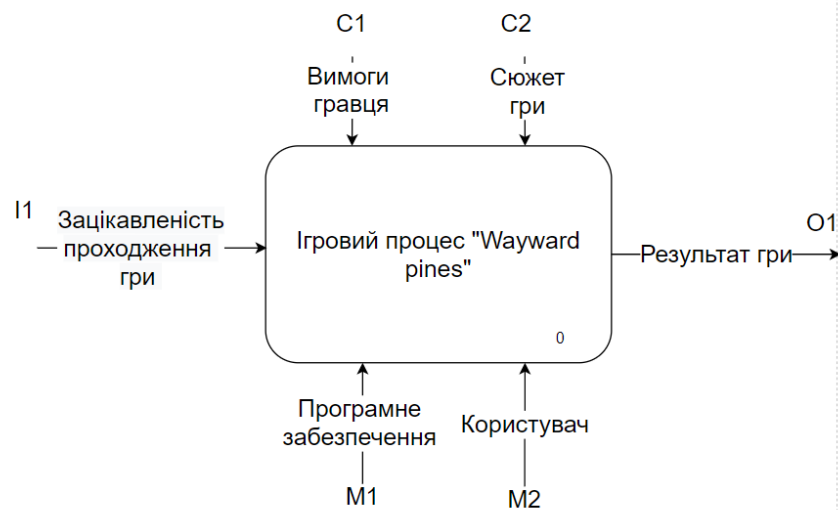


Рис 2.3 – IDEF0 ігрової системи

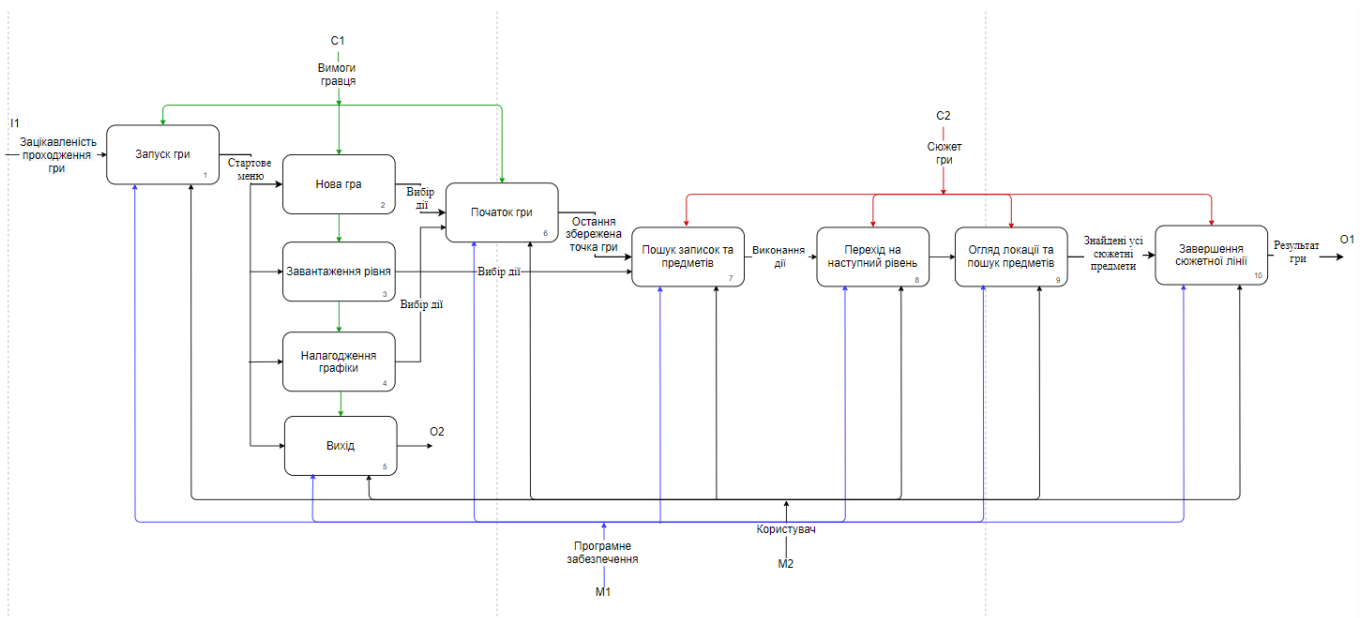


Рис 2.4 – Декомпозиція діаграми ігрової системи

Діаграма класів аналізу - це статичне представлення структури моделі, що відображає статичні елементи, такі як: класи, типи даних, їх зміст та взаємозв'язки між собою.

Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм.

Діаграма класів слугує для представлення статичної структури моделі системи у об'єктно-орієнтованому програмуванні. На цій діаграмі представлені класи, інтерфейси та об'єкти.

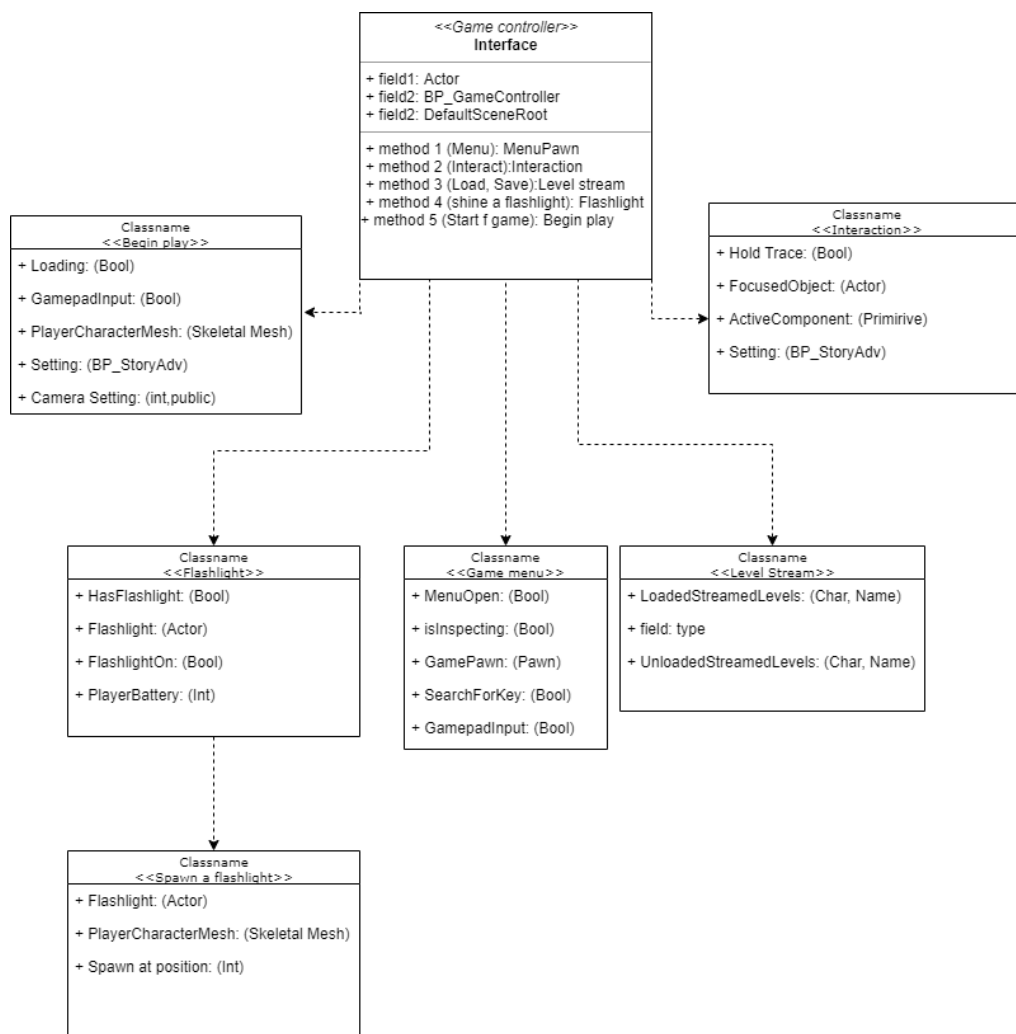


Рис 2.5 – Діаграма класів аналізу

Діаграма послідовності відображає взаємодії об'єктів впорядкованих за часом. Зокрема, такі діаграми відображають задіяні об'єкти та послідовність відправлених повідомлень.

На діаграмі послідовностей показано у вигляді вертикальних ліній різні процеси або об'єкти, що існують водночас.

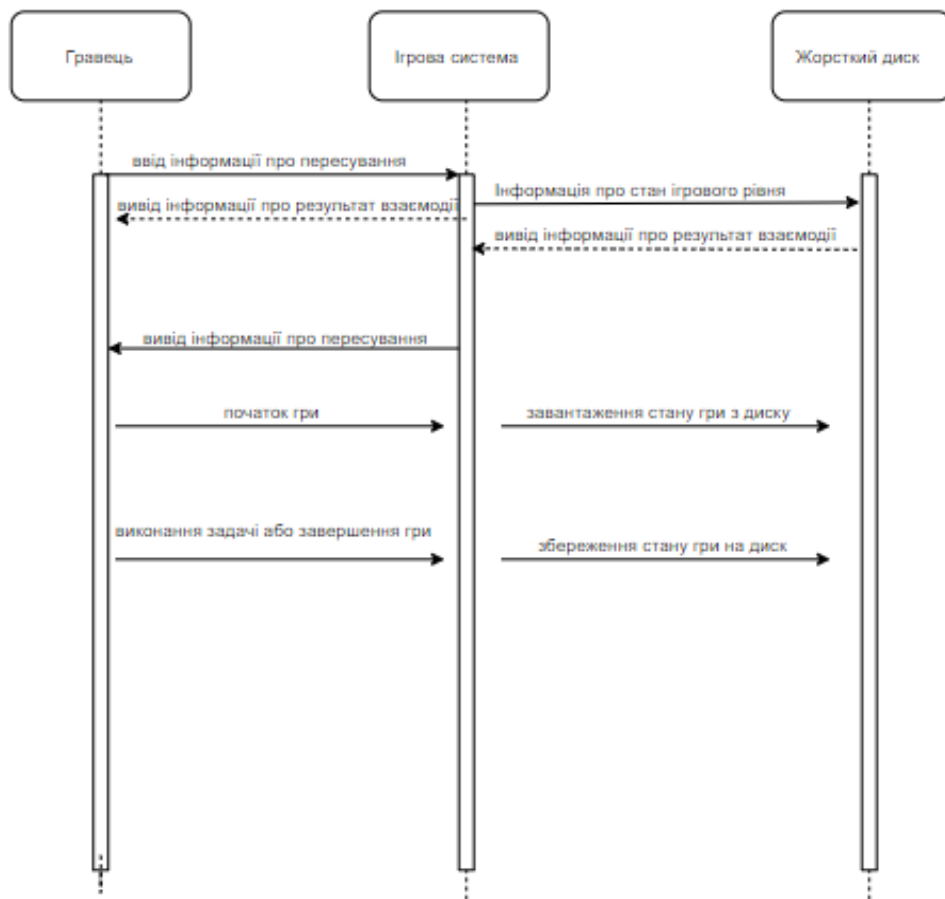


Рис 2.6 – Діаграма послідовності

## 2.2 Проектування ігрової системи

Діаграма станів – діаграма, що визначає зміну станів об'єкта у часі, одна з діаграм моделювання поведінки в UML [10].

Елементами діаграми є:

- Коло, що позначає початковий стан;
- Коло з невеличким точкою у середині, що позначає кінцевий стан;
- Стрілка, що позначає перехід. Назва події, що викликає перехід, відзначається над/під стрілкою.

Представлення даної діаграми наведено нижче на рис. 2.7.

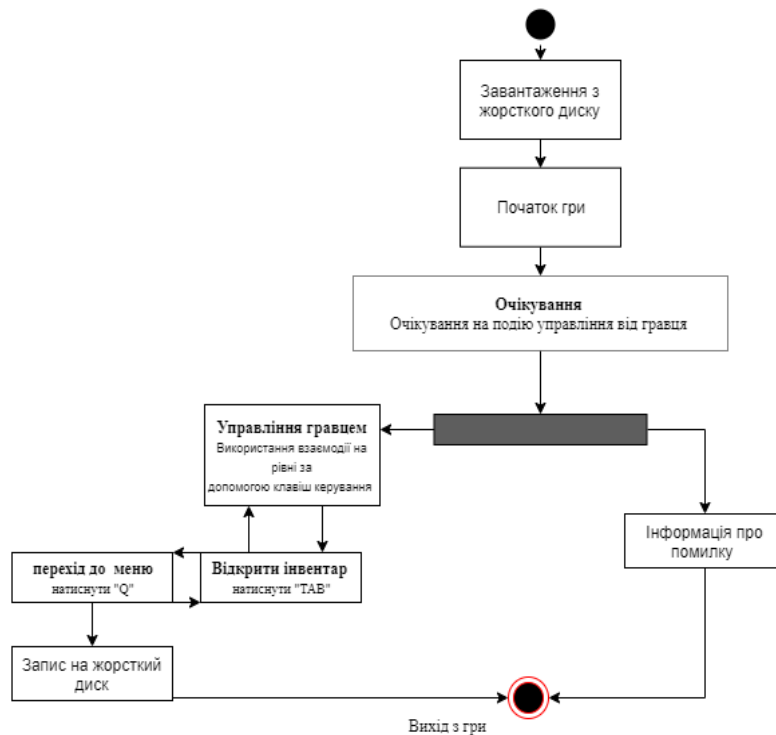


Рис 2.7 – Діаграма станів ігрової системи.

Діаграми діяльності це візуальне представлення діаграми діяльності який є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій [11, 12].

Представлення даних діаграм наведено нижче на рис. 2.8.

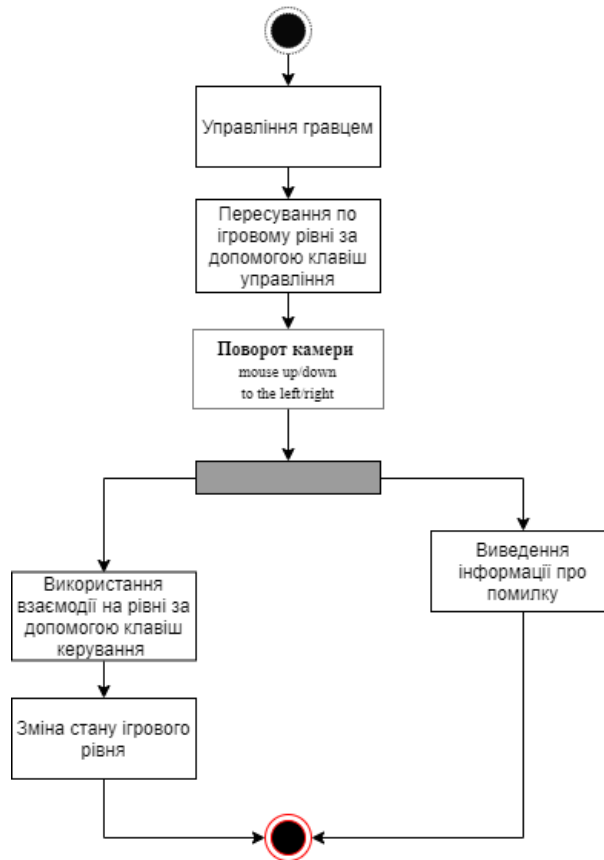


Рис 2.8 – Діаграма діяльності компоненту ігрової системи

Модель реалізації показує фізичні компоненти реалізації з метою полегшення їх сприйняття та управління. Модель реалізації визначає основні блоки впровадження, навколо яких організовані робочі групи, а також блоки, якими можна окремо привласнювати версії та змінювати їх.

Більш детальна модель реалізації може також включати файли вихідного коду низького рівня, а також їх взаємозв'язок з моделлю проектування. Такий ступінь подробиці рекомендується тільки в тому випадку, якщо є автоматична синхронізація між моделлю і файлами.

Представлення даної діаграм компонентів наведено нижче на рис. 2.9

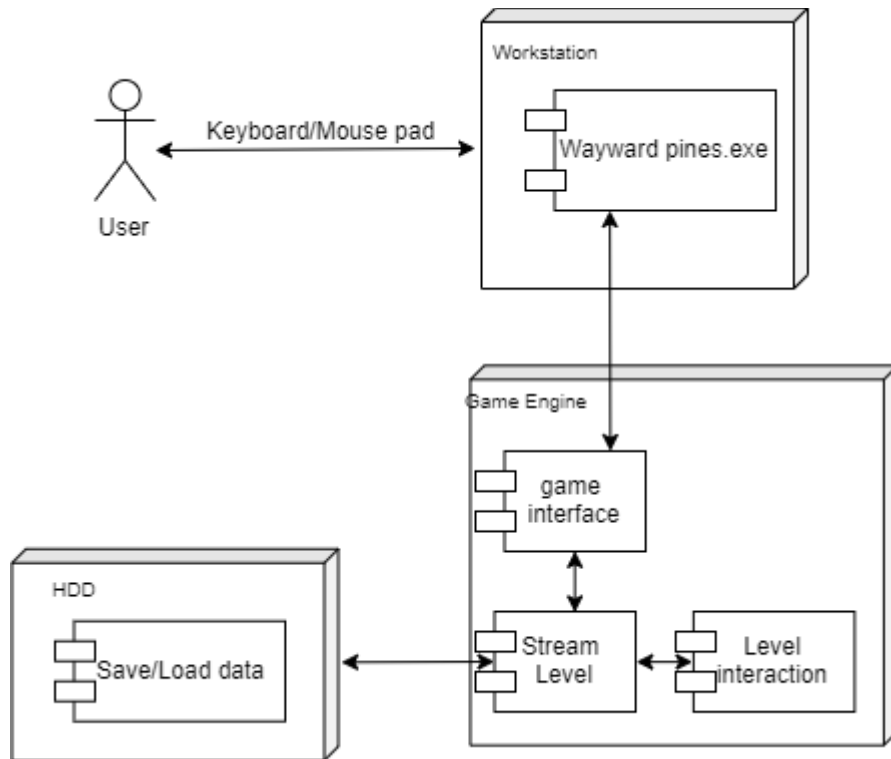


Рис 2.9 – Діаграма компонентів

У даному розділі були розглянуті основні етапи моделювання ігрової системи «Wayward pines», по кожному з них була оброблена довідкова інформація, та виходячи з неї були побудовані діаграми на яких показано варіанти використання ігрової системи, представлення функціонального моделювання ігрової системи за допомогою IDEF0, показано статичне представлення структури моделі за допомогою діаграми класів аналізу. Також були розроблені діаграми послідовності які відображають задіяні об'єкти та послідовність відправлених повідомлень, та діаграми станів, а також розроблені діаграми діяльності [13].

Отже на основі вивченої інформації та створених діаграм можна докладно подивитися на принципи роботи основних компонентів ігрової системи та на те як вони взаємодіють між собою та користувачем у ході ігрового процесу.

### 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ДОДАТКУ

#### 3.1 Програмна реалізація

Встановлення рушія та створення проекту.

Для того, щоб почати безпосередню роботу з Unreal Engine 4 нам необхідно встановити рушій. Щоб це зробити потрібно відкрити лаунчер «Epic Games», вибрати пункт «Unreal Engine» та натиснути на кнопку вказану на рис. 3.1.

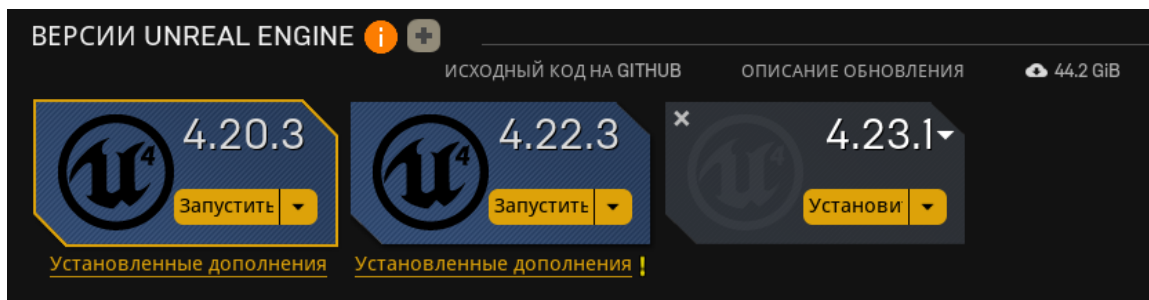


Рисунок 3.1 – Приклад встановлення ігрового рушія

Наступним кроком, після скачування та установки двигуна, потрібно натиснути кнопку «Запустити», де відкриється меню з вибором для створення проекту. На даному вікні відображені усі початкові налагодження такі як: вибір типу програмування «Blueprint» чи «C++» платформа проектування якість графіки, та стартовий контент рис. 3.2.

У нашому випадку, ми обираємо проект «Third Person» та нижче під вибором типу проекту обираємо платформу «PC», настройки графіки «Maximum Quality» та пункт для створення пустого проекту «No Starter Content», обираємо папку для зберігання проекту і останнім кроком натискаємо на кнопку «Create Project»

Після виконання усіх вище наведених дій буде створено проект у якому можна буде почати створювати гру приклад на рис. 3.3.

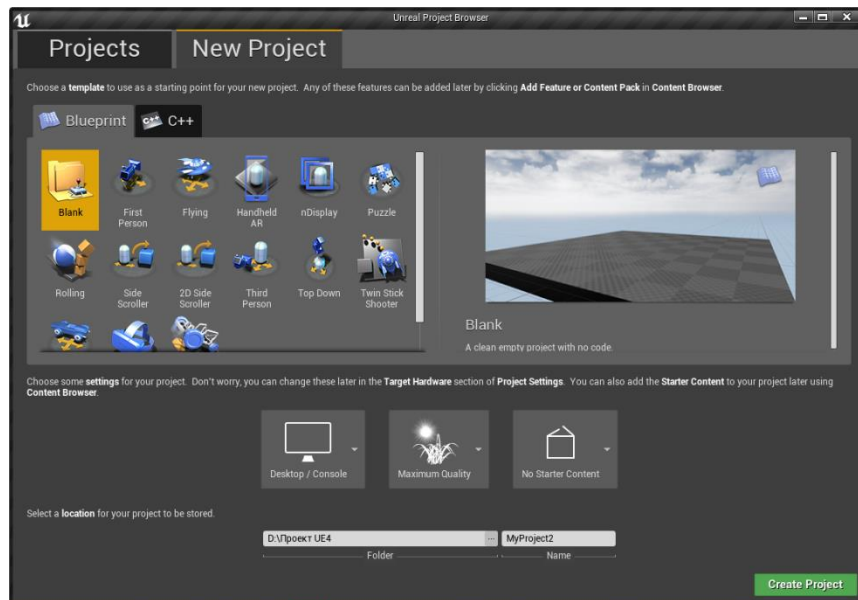


Рисунок 3.2 – Приклад вікна для обрання типу проекту

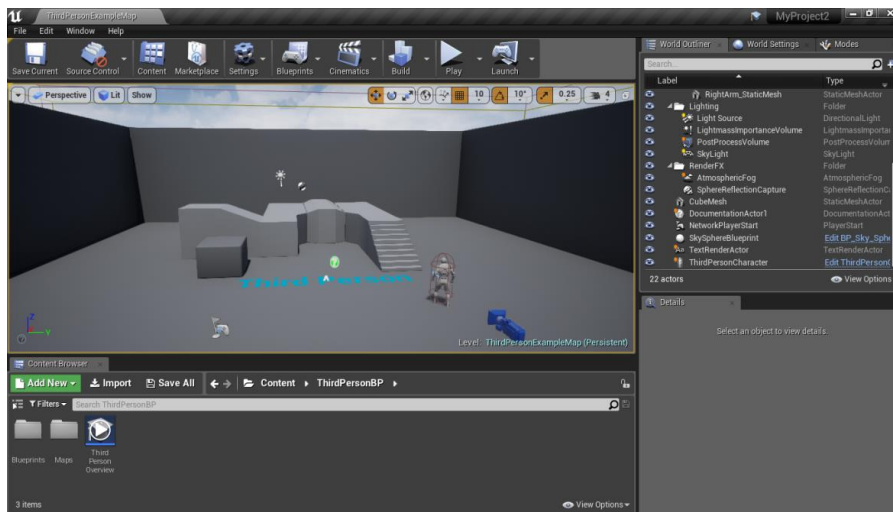


Рисунок 3.3 – Приклад створеного проекту

У даному вікні є велика кількість вкладок для виконання різних дій з обраним об'єктом, на сцені, перелік яких буде наведений нижче:

1) Content Browser – дана вкладка відповідає за взаємодію між всіма елементами у проекті та має можливість створення нових, приклад даного меню наведений на рис. 3.4;



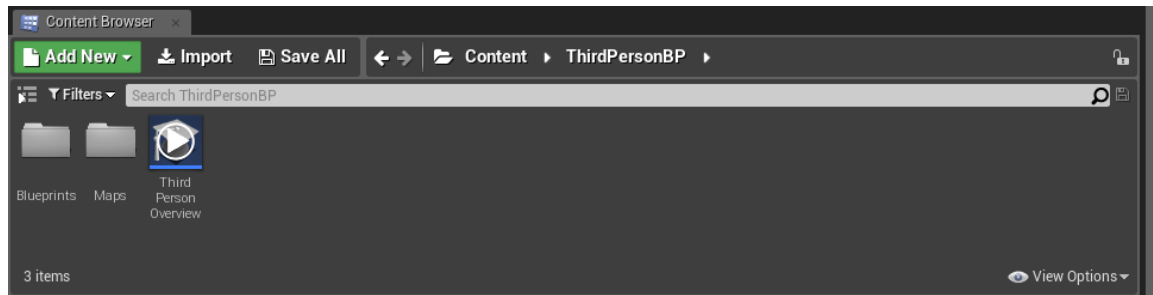


Рисунок 3.4 – Приклад вікна «Content Browser»

2) World Outliner – дана вкладка відображає усі елементи які присутні у даній сцені приклад даного меню наведений на рис. 3.5;

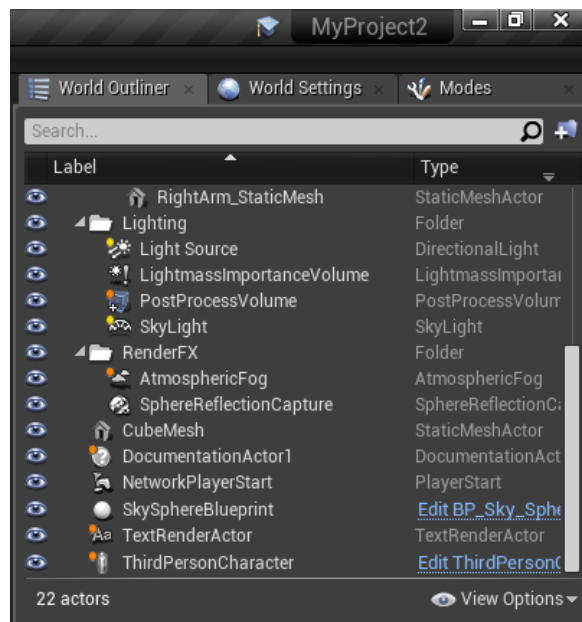


Рисунок 3.5 – Приклад вікна «World Outliner»

3) World Settings – дана вкладка відображає усі можливі для змінення параметри створеної сцени приклад даного меню наведений на рис. 3.6;

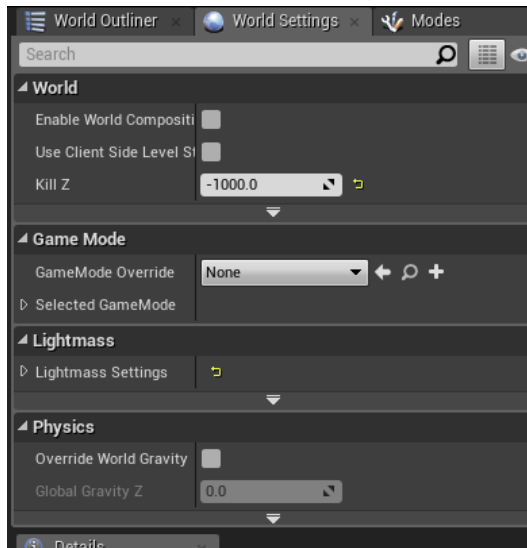


Рисунок 3.6 – Приклад вікна «World Settings»

4) Modes – дане вікно містить у собі усі основні режими роботи з об’єктами такими як камери, світло, тригери, та створення нового ландшафту, та редагування його за допомогою різних видів інструментів налагодження приклад даного вікна наведений на рис. 3.7;

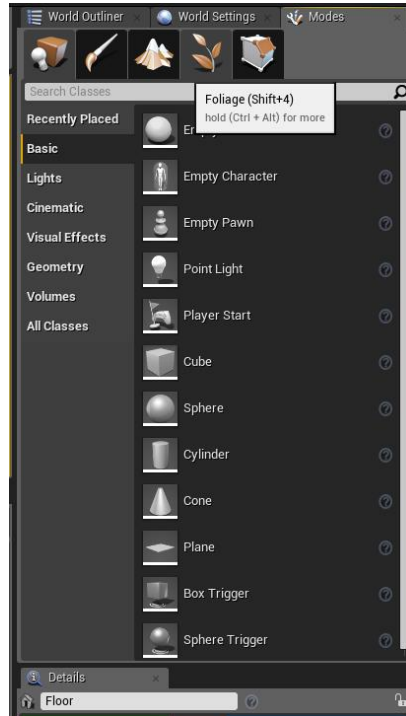


Рисунок 3.7 – Приклад вікна «Modes»

5)Details – дане вікно містить у собі усі ключові функції одного обраного в цей час об’єкта призначена для редагування та зміни параметрів приклад даного вікна наведений на рис. 3.8;

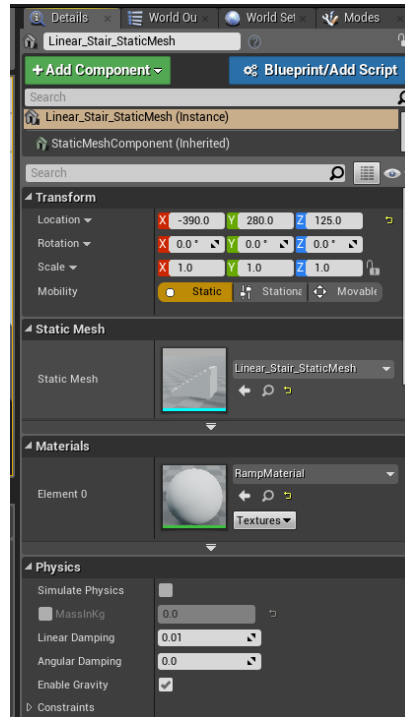


Рисунок 3.8 – Приклад вікна «Details»

6)Toolbar – Основна панель інструментів на якій розміщені кнопки керування проектом такі як:

- Save – призначений для зберігання проекту;
- Source Control – використовується для управління змінами коду та даних гри;
- Blueprints – містить в собі контроль над програмною частиною рівня;
- Build – відповідає за обробку текстур, тіней та об’єктів на сцені;
- Play відкриває вікно у якому запускається проект;
- Launch – вибір типу відкриття проекту у окремому лаунчері або у головному вікні, приклад даного вікна наведений на рис. 3.9;

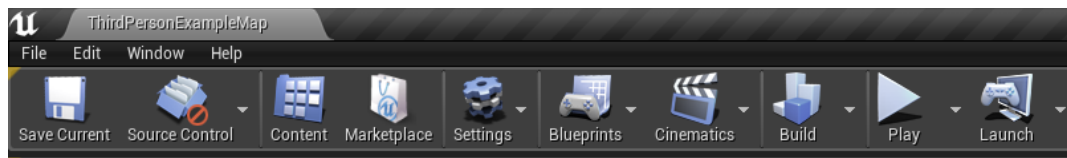


Рисунок 3.9 – Приклад вікна «Toolbar»

Створення основних дії гри.

Після того, як були розглянуті основні меню ігрового рушія Unreal Engine 4 переходимо до основних механік гри створення яких буде наведено у наступному розділі.

Першим кроком у створенні головного меню є створення додаткової мапи, на якій воно буде розташоване. Для цього нам необхідно створити об'єкт типу «Level», натискаємо ПКМ на вільному просторі вікна «Content browser» та обраємо відповідний об'єкт.

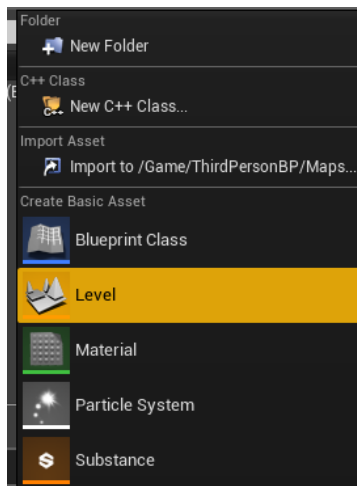


Рисунок 3.10 – Приклад створення об'єкту типу «Level»

Після того, як ми створили «Level» необхідно перейти на даний пустий рівень - натискаємо на об'єкт два рази лівою кнопкою миші.

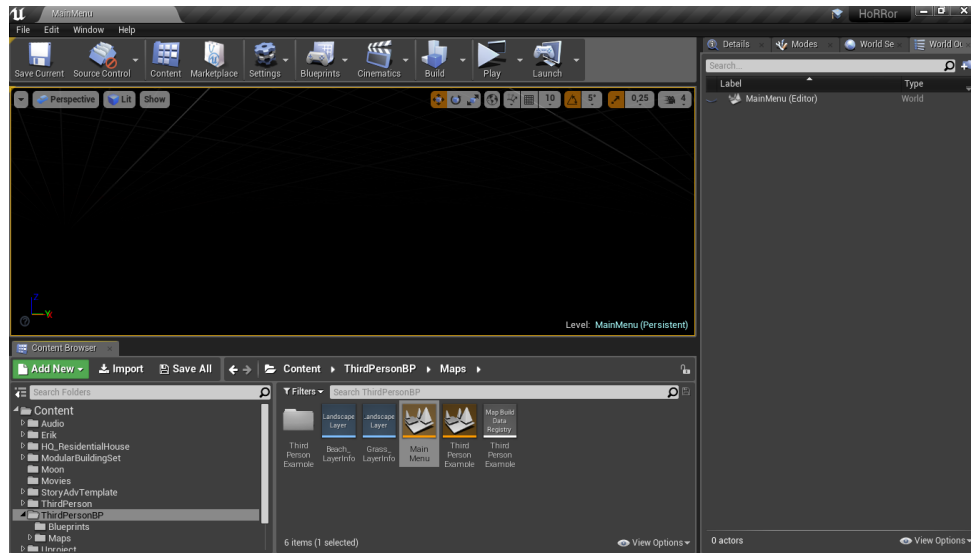


Рисунок 3.11 – Приклад створеного об'єкту типу «Level»

Наступним кроком у ході роботи з головним меню є створення користувацького інтерфейсу, об'єктом якого виступає «Widget Blueprint». Для його використання потрібно натиснути на вільному просторі вікна «Content browser» ПКМ та відкрити меню «User interface» та обрати «Widget Blueprint».

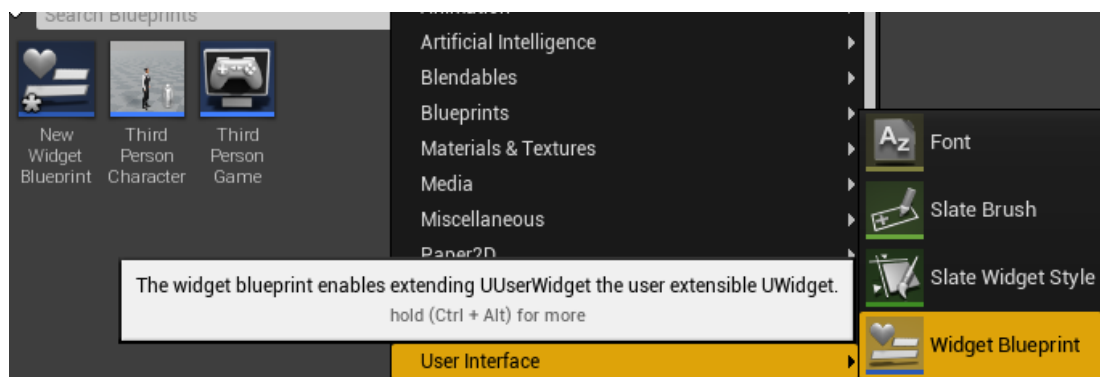


Рисунок 3.12 – Приклад створення об'єкту типу «Level»

Для подальшого використання та налагодження головного меню необхідно обрати меню під назвою «Palette», з якого треба методом «Drag and drop» використати основні керуючі елементи як:

- «Button»;
- «Image»;

- «Text»;
- «Vertical Box»;
- «Horizontal Box».

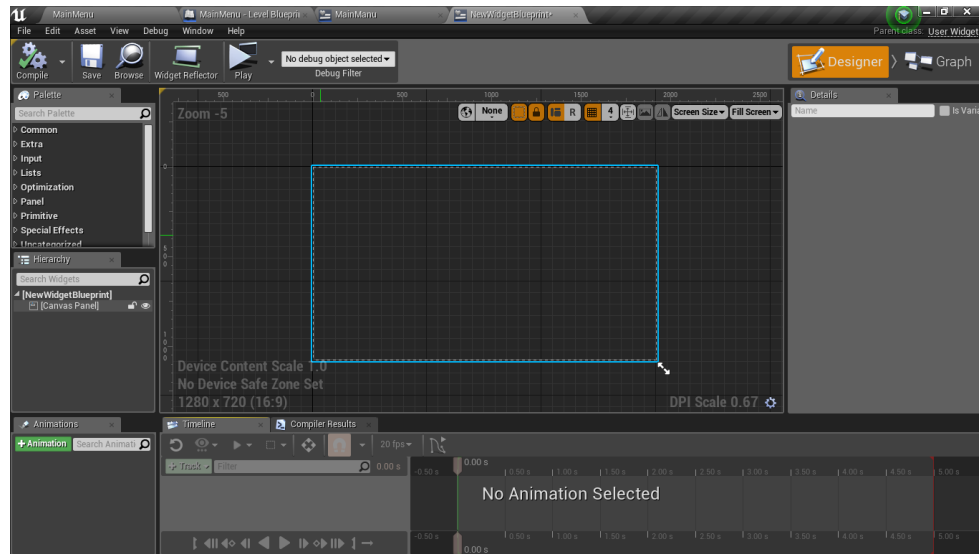


Рисунок 3.13 – Приклад створеного об'єкту типу «Widget Blueprint»

Всі вище наведені об'єкти у ході налаштування потрібно розмістити на робочій поверхні, налаштувати розміри та їх зовнішній вид. Приклад проведених налаштувань наведений на рис. 3.16.

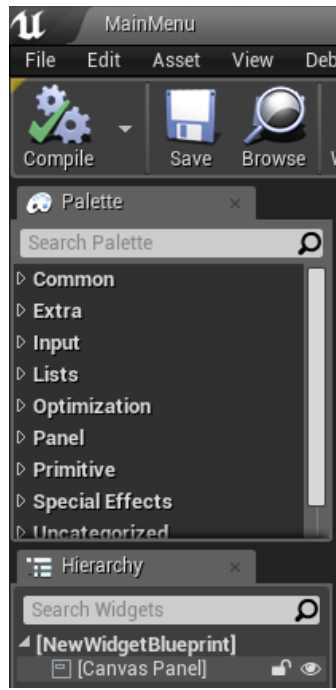


Рисунок 3.14 – Перелік об'єктів меню «Palette»

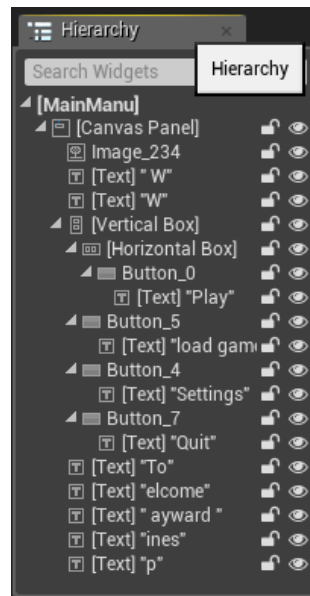


Рисунок 3.15 – Перелік об'єктів та кнопок на головному меню

На рис. 3.16 вказаний перелік об'єктів та кнопок на головному меню. За їх допомогою ми будемо створювати основний вигляд меню. Для цього ми використаємо такі об'єкти як «Text», «Canvas panel», «Button» та «Image». Для кожного з них будуть прописані індивідуальні сценарії Blueprints.

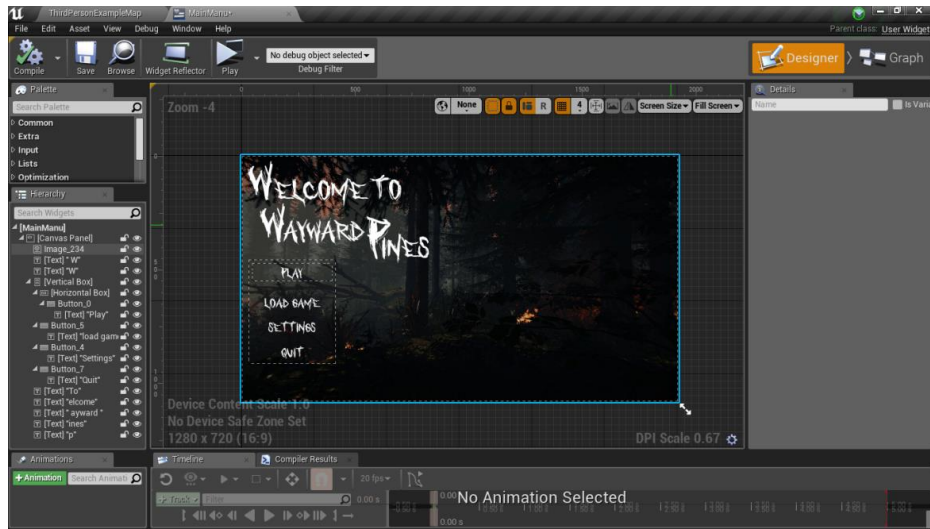


Рисунок 3.16 – Приклад створеного головного меню

Розробка переходів (дії при натисканні на керуючі кнопки).

Наступним кроком роботи з налагодженням роботи меню є створення переходів по натисканню на керуючі кнопки.

Для того, щоб створити даний перехід, потрібно натиснути на кнопку у полі «Hierarchy» та у полі «Details» вибрати подію «On Click» та натиснути на неї - відкриється поле «Event Graph»



Рисунок 3.17 – Приклад створених подій у полі «Event Graph»



Далі йде індивідуальне налагодження кожної з кнопок які використовуються у даному проєкті, нижче буде наведений список даних кнопок з коротким описом та прикладом їх роботи.

Перелік створених кнопок для керування грою на етапі головного меню:

- «Play»;
- «Load game»;
- «Setting»;
- «Quit»;

Якщо було обрано «Load game» реалізуємо функцію зберігання та завантаження ігрового процесу. При натискання на кнопку «Setting» перед користувачем з'явиться меню вибору налаштування графіки приклад наведений на рис. 3.18.

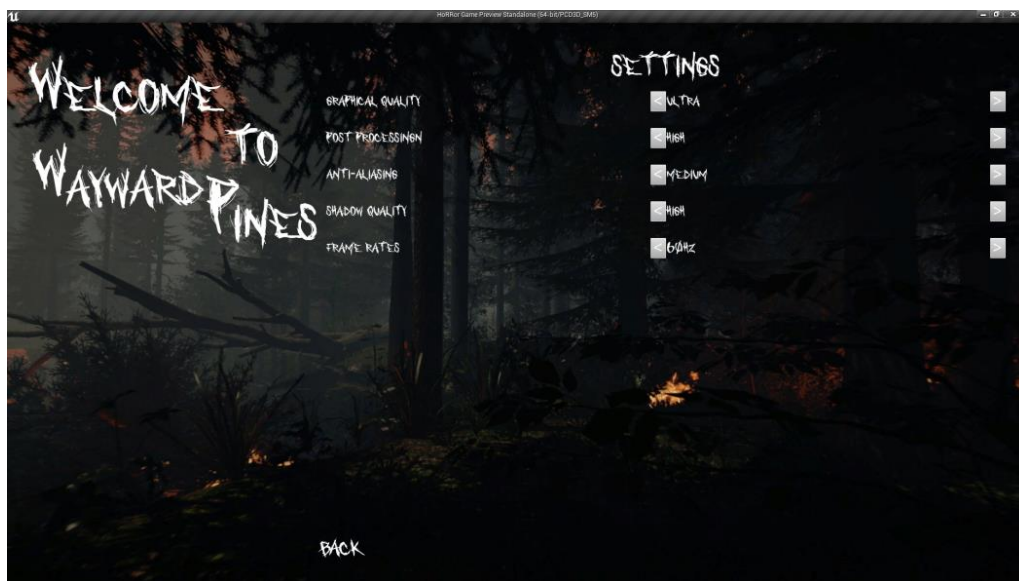


Рисунок 3.18 – Приклад меню зміни налаштування графіки

Для реалізації даного меню було створено блупронт, на який додано змінні типів «array of text» — для зберігання масивів команд ігрового рушія, для зміни якості графіки та «integer» для зберігання індексу обраної команди від 0 до 3, тобто у кожному текстовому масиві зберігаються чотири команди, які

відповідають за зміну якості графіки. Приклади даних команд та приклад програмної реалізації зміни при натисканні на кнопку приведені на рис. 3.19

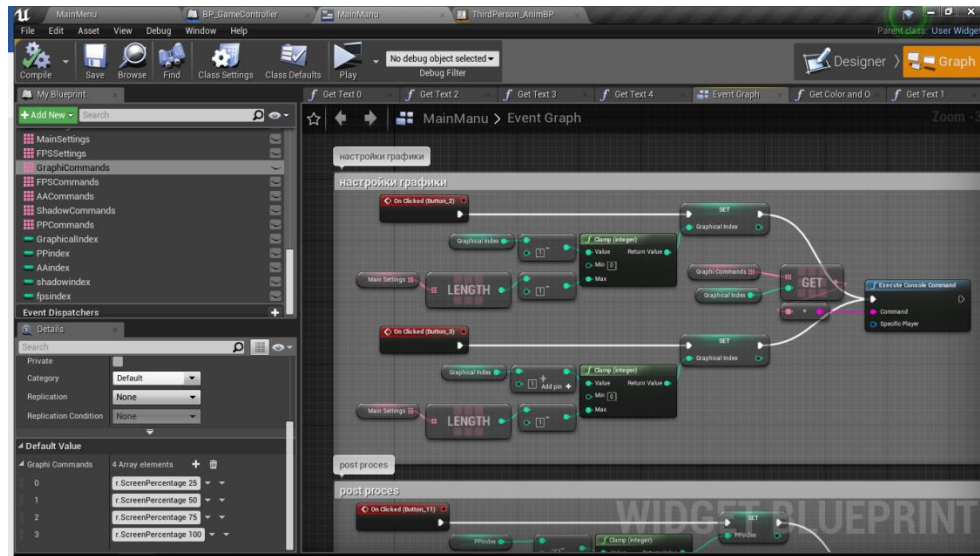


Рисунок 3.19 – Приклад змінних та реалізації блупринта

Після натискання на кнопку «Play» для користувача починається завантаження першого ігрового рівня. Для цього необхідно створити «On click» подію при натисканні на кнопку та додати функцію «Open Level», приклад наведено на рис. 3.20.

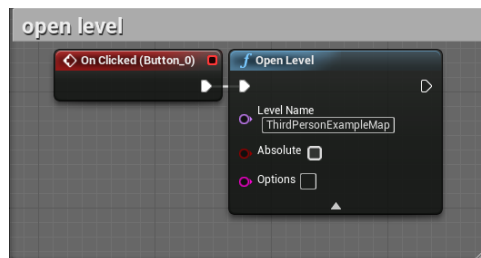


Рисунок 3.20 – Приклад реалізації блупринта події «On click»

Після натискання на кнопку «Quit», ігровий додаток завершує роботу, приклад реалізації блупринта наведений на рис. 3.21

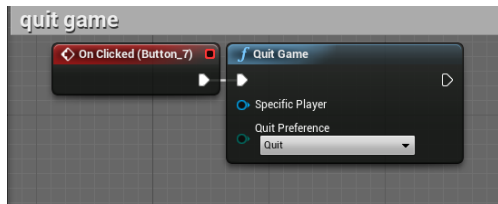


Рисунок 3.21 – Приклад реалізації блупринта події на кнопку «Quit»

Наступним кроком виконання роботи є створення моделей за допомогою додатку Autodesk 3ds Max, та їх подальший імпорт у ігровий рушій.

Для прикладу роботи з даним додатком буде приведено створення та імпорт одної моделі у вигляді чашки, яка у подальшому використанні у двигуні буде одним із ключових елементів геймплею.

Першим кроком у ході виконання моделі було встановлено використання міліметрів як одиниць величини вимірювання у меню під назвою «Units Setup», приклад наведений на рис. 3.22. Далі було створено два примітива типу «Циліндр» за допомогою поля під назвою «Standard Primitives» приклад наведений на рис. 3.23

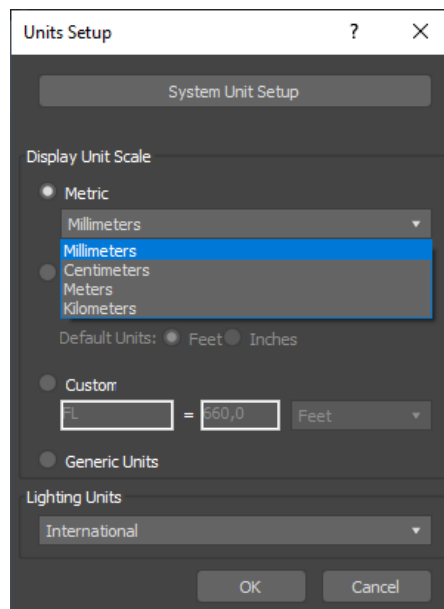


Рисунок 3.22 – Приклад встановлення одиниць вимірювання

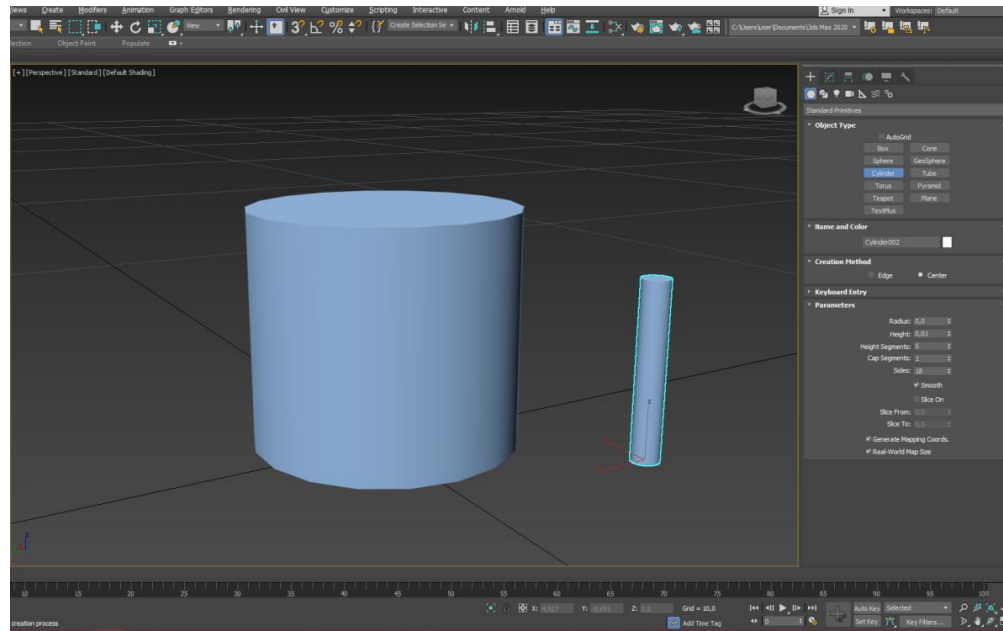


Рисунок 3.23 – Приклад створених циліндрів

Після створення цих двох елементів, їх потрібно конвертувати у тип «Editable Mesh» що дасть змогу за допомогою сітки ключових точок змінювати об'єкти, та придати їм вигляд, що буде більш менш схожий на потрібний нам предмет інтер'єру. Далі після для того щоб сумістити ці два елементи у один об'єкт потрібно використати модифікатор «Attach» та сумістити їх приклад наведений на рис. 3.24

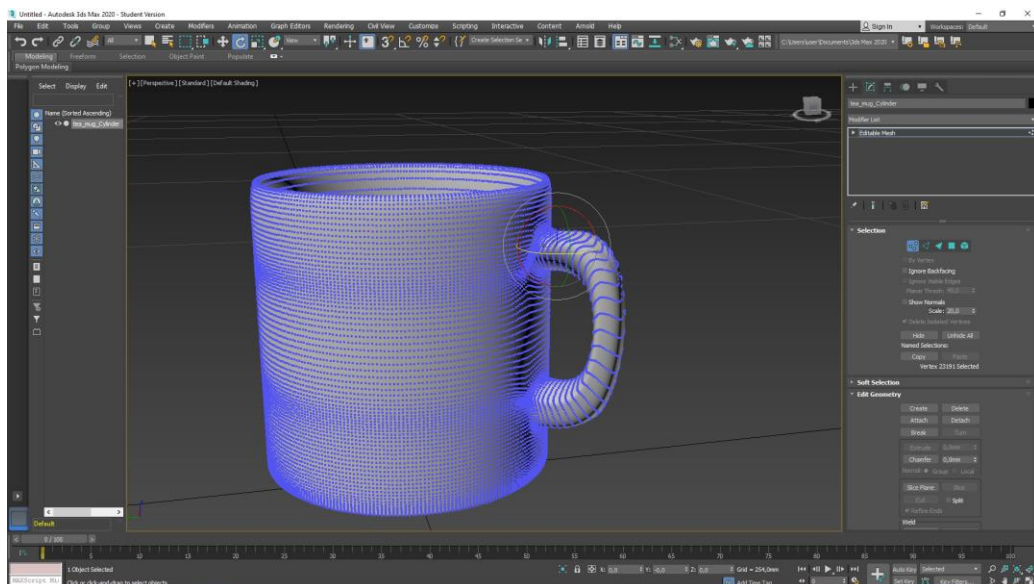


Рисунок 3.24 – Приклад суміщених об'єктів

Далі після того, як модель створена, необхідно імпортувати її у двигун, для цього нам необхідно натиснути на об'єкт щоб його виділити, потім обрати у меню «File» з випадаючого списку «Import», обрати директорію у якій буде збережено об'єкт та обрати тип .obj чи .fbx, приклад наведений на рис. 3.25.

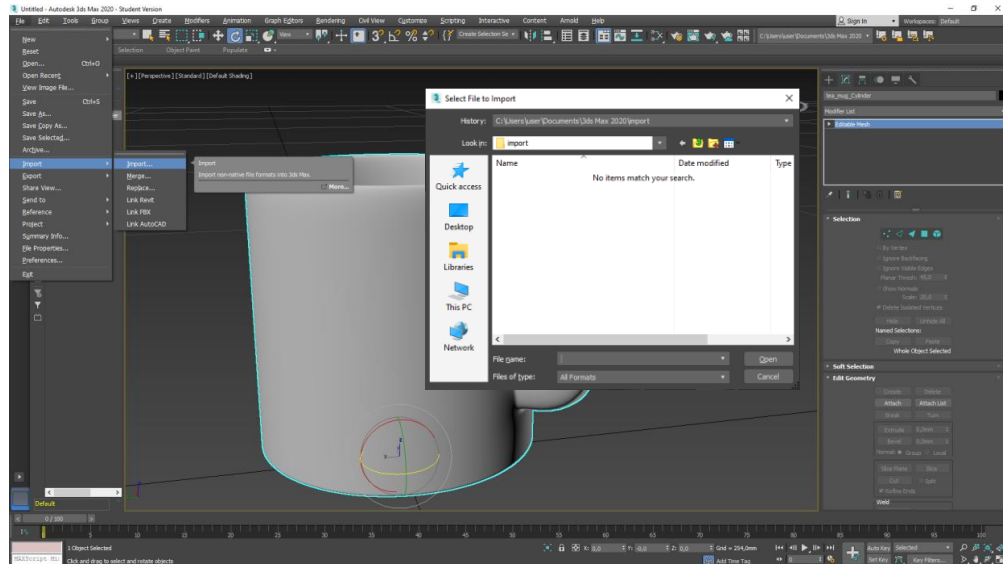


Рисунок 3.25 – Import об'єкту

Наступним етапом у створенні гри є запис звуків, який у свою чергу буде реалізовано за допомогою додатку Audacity, який слугує для запису редагування звуків, голосу та музики.

Першим кроком у роботі з додатком потрібно створити звукову доріжку для цього у меню необхідно натиснути «Треки» далі обрати «Додати новий» та «Сtereo трек» приклад наведений на рис. 3.26

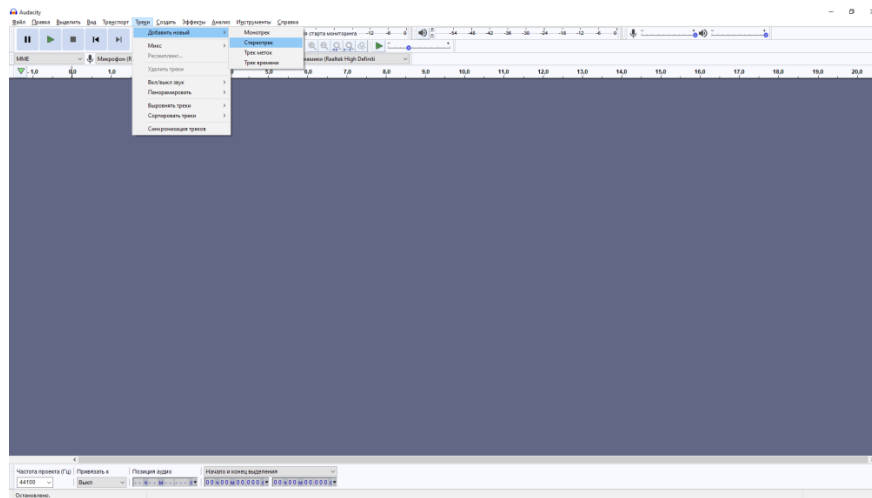


Рисунок 3.26 – Створення звукової доріжки

Після чого натиснути на кнопку «Записати новий трек» вона зображена червоним колом на верхньому полі меню, та почати записувати діалоги, що були описані у сюжеті гри приклад наведений на рис. 3.27.

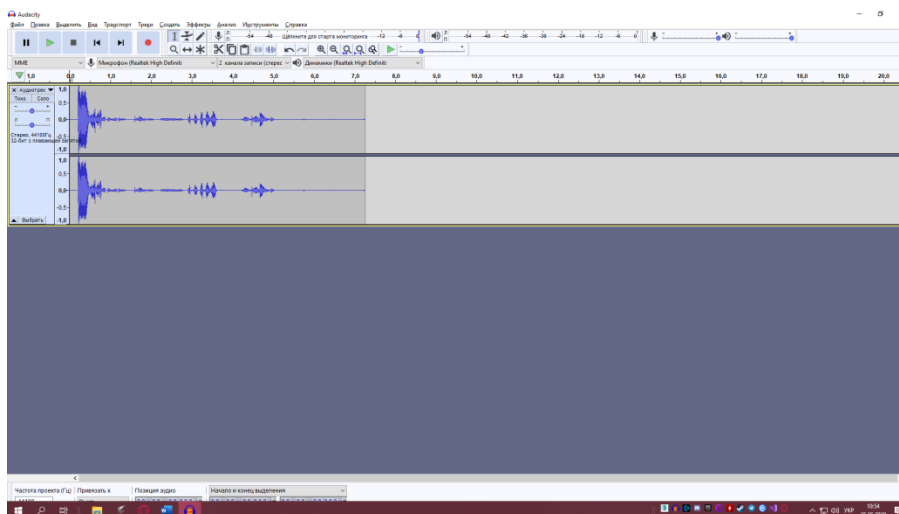


Рисунок 3.27 – Створена звукова доріжка з голосом

Останнім кроком після того як голос, або необхідний звук був записаний та оброблений, необхідно імпортувати його у ігровий рушій, для цього у меню «Файл» обрати «Імпорт» та «Аудіо». Після чого обрати директорію у якій буде збережено об'єкт та обрати тип .wav, приклад наведений на рис. 3.28.

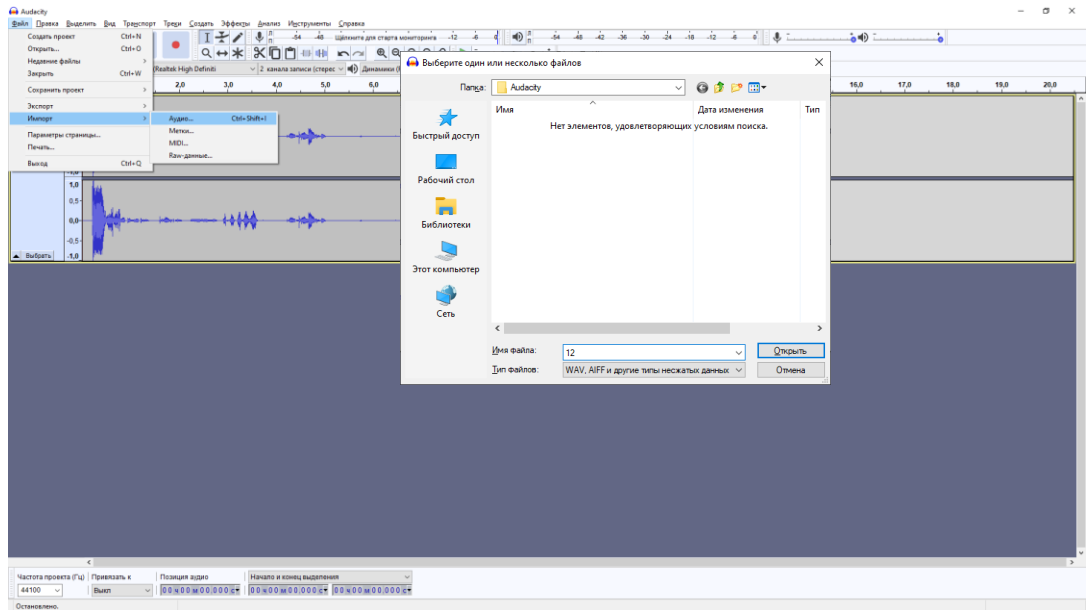
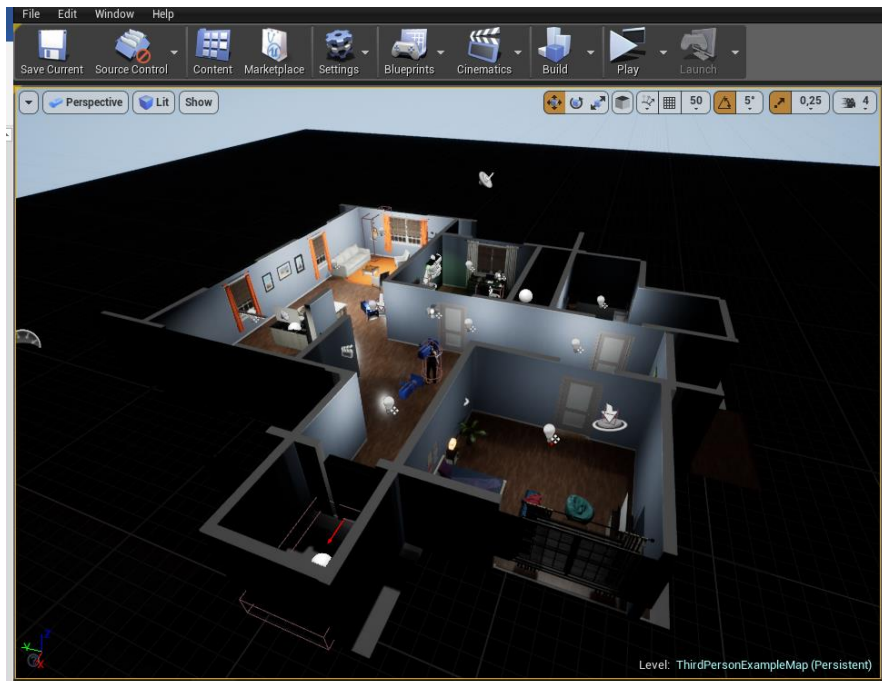


Рисунок 3.28 – Імпорт звукового файлу

Тепер після того, як головне меню спроектовано, записані усі звуки та створені моделі для сцени, які в свою чергу були імпортовані у двигун, переходимо до створення та дизайну наступної карти назвою якої є – квартира головного персонажу.



### Рисунок 3.29 – Приклад реалізації ігрового рівня

У якості стін були використані стандартні для Unreal Engine 4 примітиви. Елементи оточення – такі як шафи, двері, столи та інші елементи інтер'єру були розроблені з використанням додатку від компанії Autodesk 3Ds Max. Було проведено ряд дій, а саме – створення матеріалів для елементів оточення, щоб додати їм індивідуальності та придати реального вигляду.

Створено та анімовано головного персонажа, яким буде керувати гравець. Щоб він почав рухатися, спочатку потрібно додати події для кнопок управління, при натисканні починає програватися анімація руху та імітація пересування.

Наступним кроком потрібно налагодити скорості для двох режимів пересування – ходьба та біг, та розробити анімацію, яка буде залежно від параметрів скорості підбирати потрібну анімацію для ходьби або бігу відповідно. У подальшому на магістерську роботу планується розробити анімації для присідання, щоб додати елементи на карті, до яких можливо буде потрапити лише присівши.

Наступним кроком у виконанні роботи були створені взаємодії між елементами оточення – речі для пошуку та огляду, записки, ключі та ліхтарик. У якості прикладу виконання взаємодії буде приведено блупринт для появи (Spawn) ліхтарика та блупринт для опису його логіки (рис. 3.30 – 3.33).

Отже, тепер розберемо більш детально кожен блупринт та дії, що він буде виконувати. З початку йде дія вводу «Inputs», потім додається умова при дійсності якої буде виконуватися поява ліхтарика у руці головного героя - це блок «Branch». Далі в цьому блоці необхідно додати стан «Condition», при якому буде виконуватися послідовність, тобто «True», при якому далі виконується поява ліхтарика за допомогою команди «Spawn Actor» та прив'язка його до відповідного слоту на манекені за допомогою функції «Attach To Component» об'єктом якого є «Flashlight» і змінною являється «Player Character Mesh» , та «False» при якому ліхтарик не буде задіяно.



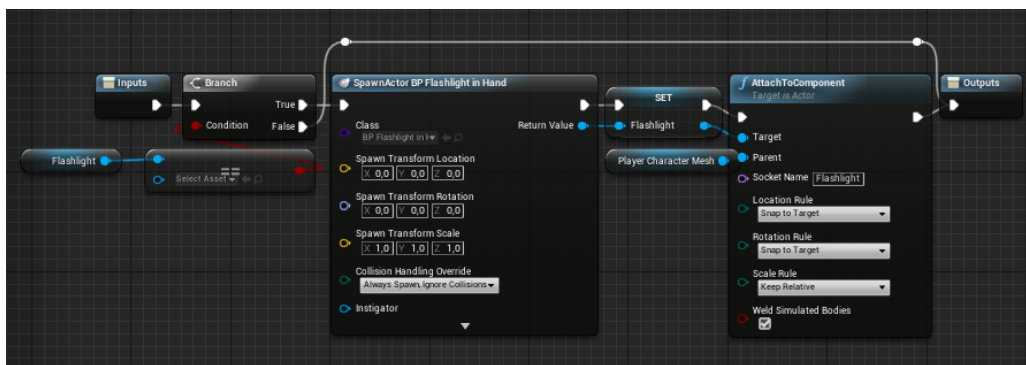


Рисунок 3.30 – Приклад Spawn на рівні об’єкту типу Flashlight

Після того, як рушій отримав значення «True», або відповідно «False», необхідно написати логіку, яка буде слугувати за ввімкнення та вимикання ліхтарика. Для цього необхідно створити подію «Event» при виникненні якої об’єкт буде виконувати відповідні дії.

Наступним кроком є додавання умови, згідно якої, якщо гравець має при собі ліхтарик, то починається дія ввімкнення світла від нього та вимикання при повторному натисканні на кнопку події. Якщо подія отримала значення «True», то далі застосовується функція «Play Sound at Location», яка слугує для програвання музики у даному випадку звуків ввімкнення та вимикання.

Далі використовується описана нами раніше функція «Spawn Flashlight» та перевірка - чи ввімкнений ліхтарик. Якщо ні, то у даному випадку йде заряджання батареї, змінною якої є «Player Battery1» та яка зберігає у собі кількість заряду з типом змінної «integer».

Наступною функцією, яку було використано для безпосереднього вмикання світла від ліхтарика, це «Set Visibility», яка використовує у якості об’єкта «Flashlight», в якому є в якості джерела світла «Spotlight».

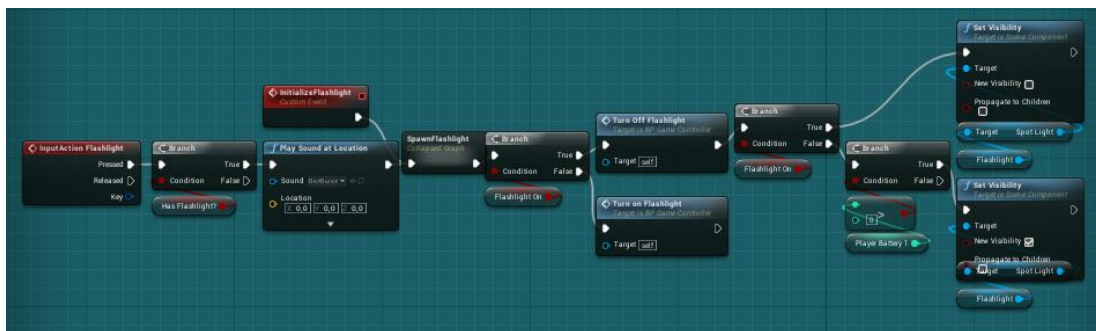


Рисунок 3.31 – Приклад послідовності для події ввім/вимк ліхтарика

Для того, щоб перевірки у наведеному вище блупринті працювали коректно, необхідно додати умови, що будуть перевірятися при зверненні до них. Для цього необхідно створити пару подій, одна з яких буде відповідати включеному ліхтарика, а інша виключеному. Та встановити за допомогою команди «Set» відповідне значення, яке відповідає тому, чи ввімкнений ліхтар та дії які він виконує, наприклад, якщо він задіяний, то йде витрата заряду батареї, а якщо вимкнений, то батарея заряджається на одне значення за кожні 0,2 секунди (що визначено як параметр, який встановлено у функції «Delay»).

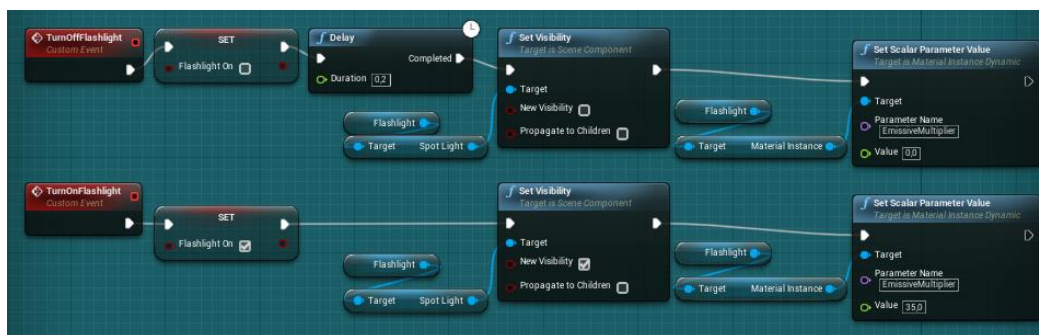


Рисунок 3.32 – Приклад умов для ввім/вимк ліхтарика

Після усіх налаштувань умов логіки ввім/вимк ліхтарика необхідно написати логіку, при якій, коли ліхтарик ввімкнений, він потребує енергію батареї, а коли вимкнений - заряджає її.

Для цього необхідно створити «Event tick», який активує відповідність часу у секундах, при якому у кожен момент часу, коли задіяний ввімкнений ліхтарик,

він потребує одну поділку батареї кожні 0,2 секунди, та коли він вимкнений - він відновлює заряд. Дані дії реалізовані за допомогою логічних операції  $int + int$  для заряду, де перше значення - це батарея, а друге - це значення, що буде додано до нього. Так само реалізовано віднімання заряду під час включеного ліхтарика за допомогою  $int - int$ , де перше значення - це батарея, а друге - це значення, що буде віднято від нього.

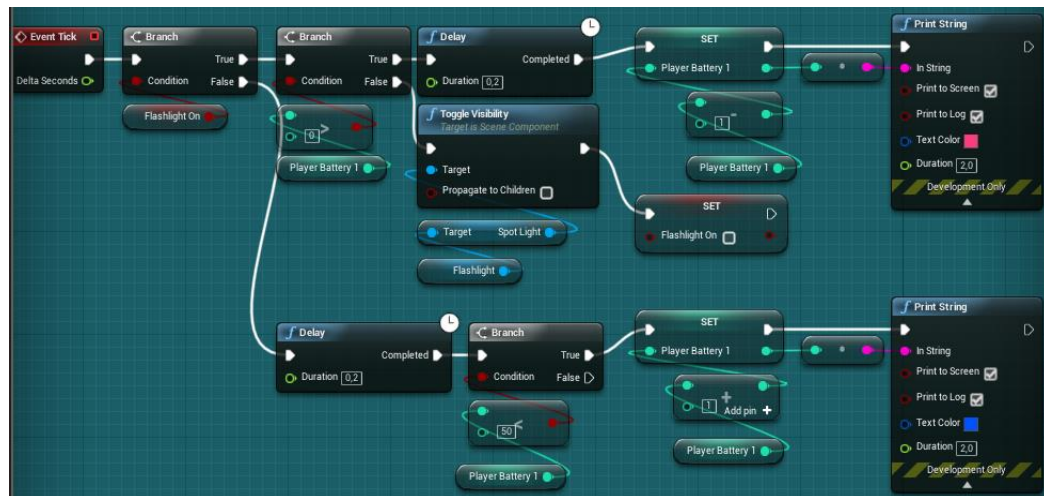


Рисунок 3.33 –Логіка розрядження та заряджання батареї ліхтарика

Наступним кроком реалізації проекту є налаштування персонажу, його вигляду та анімацій. Для цього необхідно створити такі об'єкти як «animation blueprint» та обрати скелет, на який буде додано нову анімацію (приклад наведений на рис. 3.34).

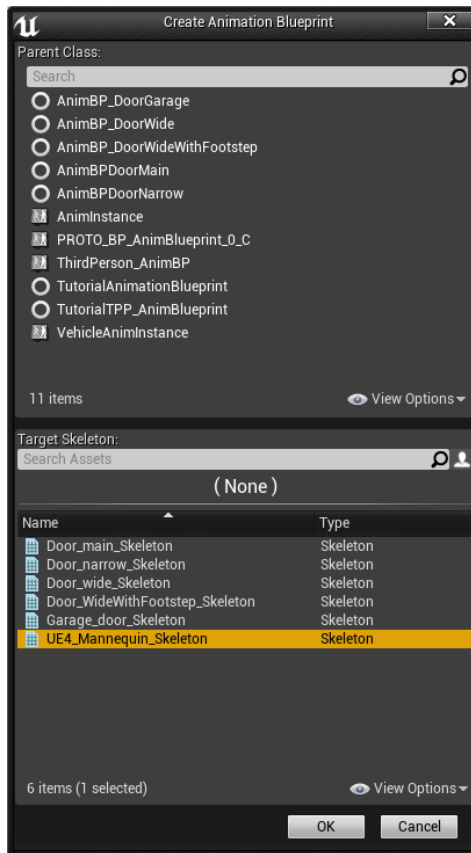


Рисунок 3.34 – Приклад поля вибору скелету для анімації

Далі перед нами відкривається два поля таких як:

1) Анімаційний граф – це граф, у якому проходить безпосереднє налаштування анімацій та їх представлення у вигляді ключових точок при звертанні до яких, вони будуть відтворюватися;

2) Граф події – це граф, який слугує для того, щоб за допомогою створення подій придати умови за яких у анімаційному графі буде коректно відобразитися фінальна анімація.

Ці графи з виконаними налаштуваннями в свою чергу приведені на рис. 3.35 – 3.35



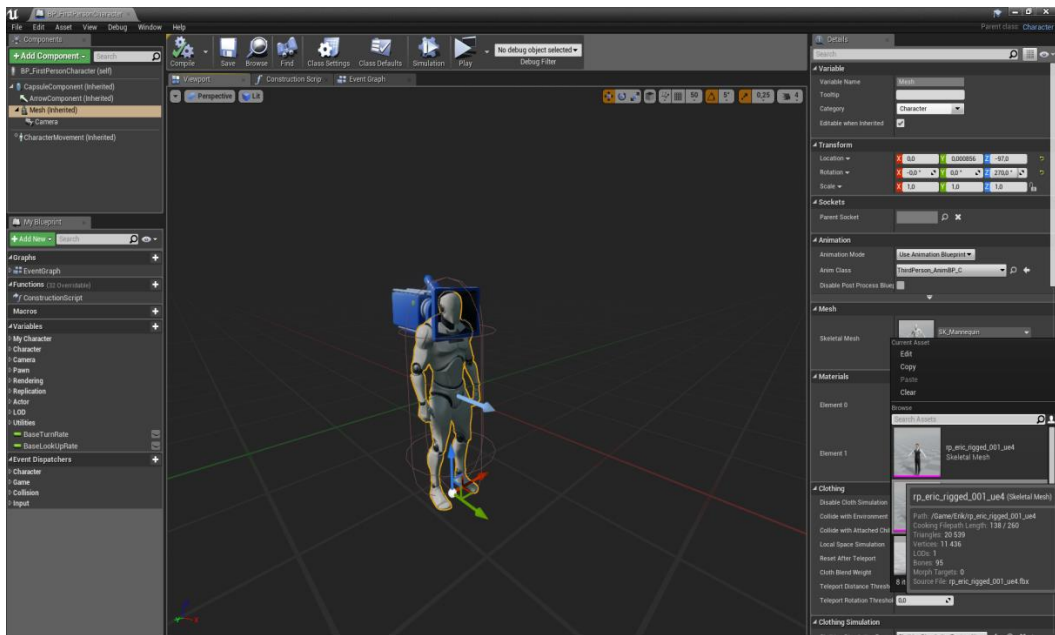


Рисунок 3.37 – Приклад поля вибору нової моделі

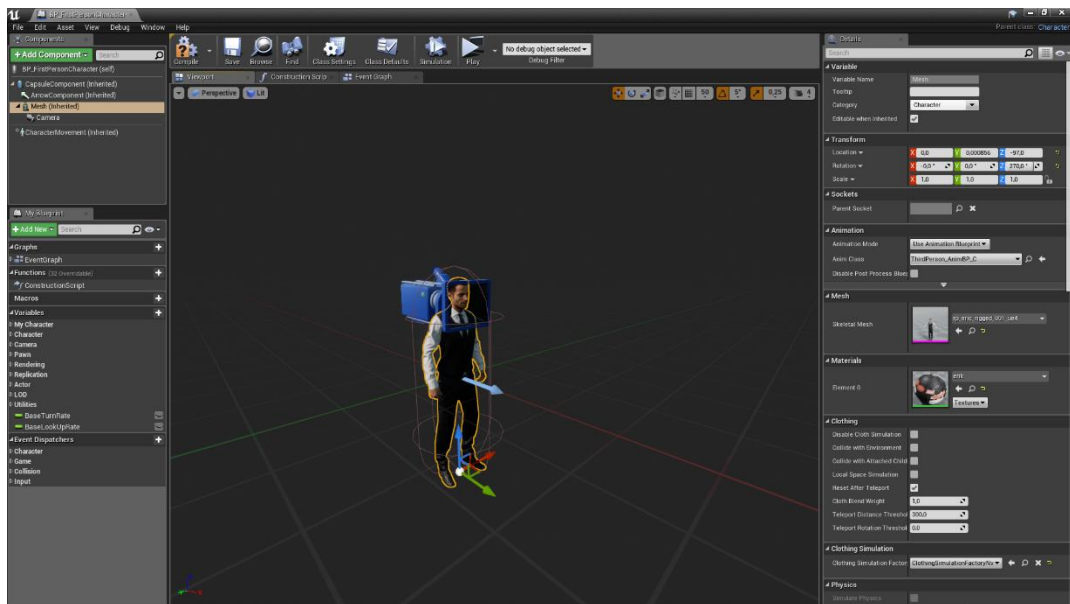


Рисунок 3.38 – Приклад обраної нової моделі

Останнім етапом перед створенням файлу типу «.exe», йде видалення матеріалів та моделей, що не були використані у проекті, це мінімізує подальший час для обробки комп'ютером даних при архівації та істотно зменшить кількість місця що вона буде займати на жорсткому диску приклад наведений на рис. 3.39.

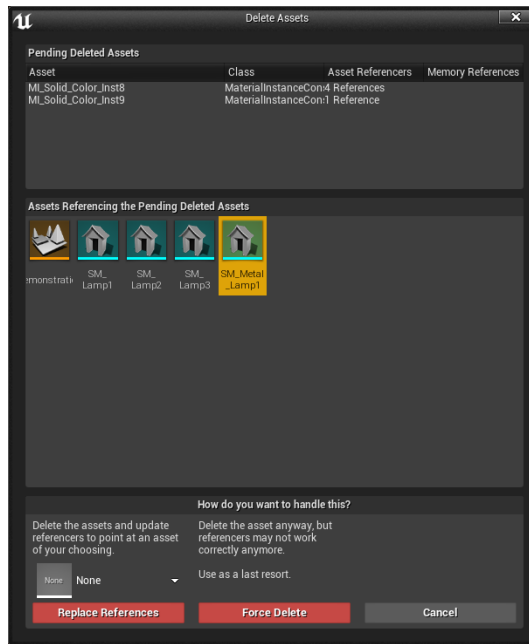


Рисунок 3.39 – Приклад видалення матеріалів та моделей

Завершаючий етап проекту це безпосереднє створення файлу типу «.exe». Щоб його створити необхідно вибрати у меню «File» далі пункт «Package project» потім «Windows» обрати «32 bit» чи «64 bit» систему, далі відкриється поле з вибором директорії для зберігання, та почнеться безпосередня компіляція проекту приклад наведений на рис. 3.40 – 3.42.

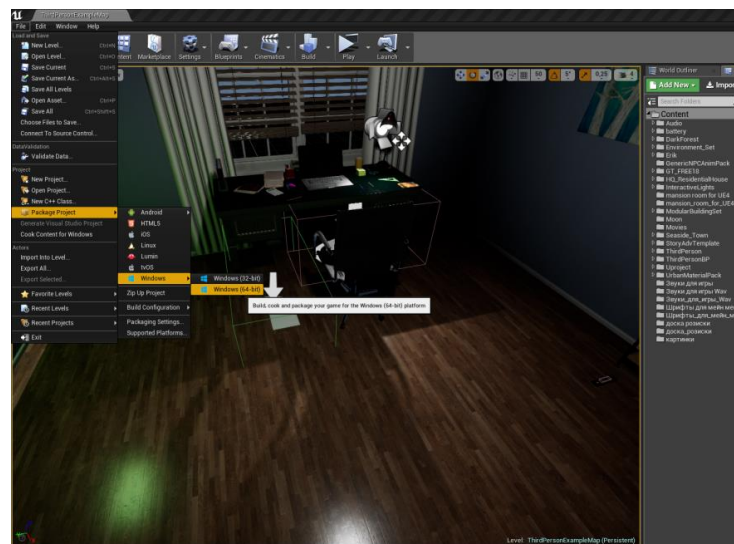


Рисунок 3.40 – Приклад параметрів вибору для компіляції

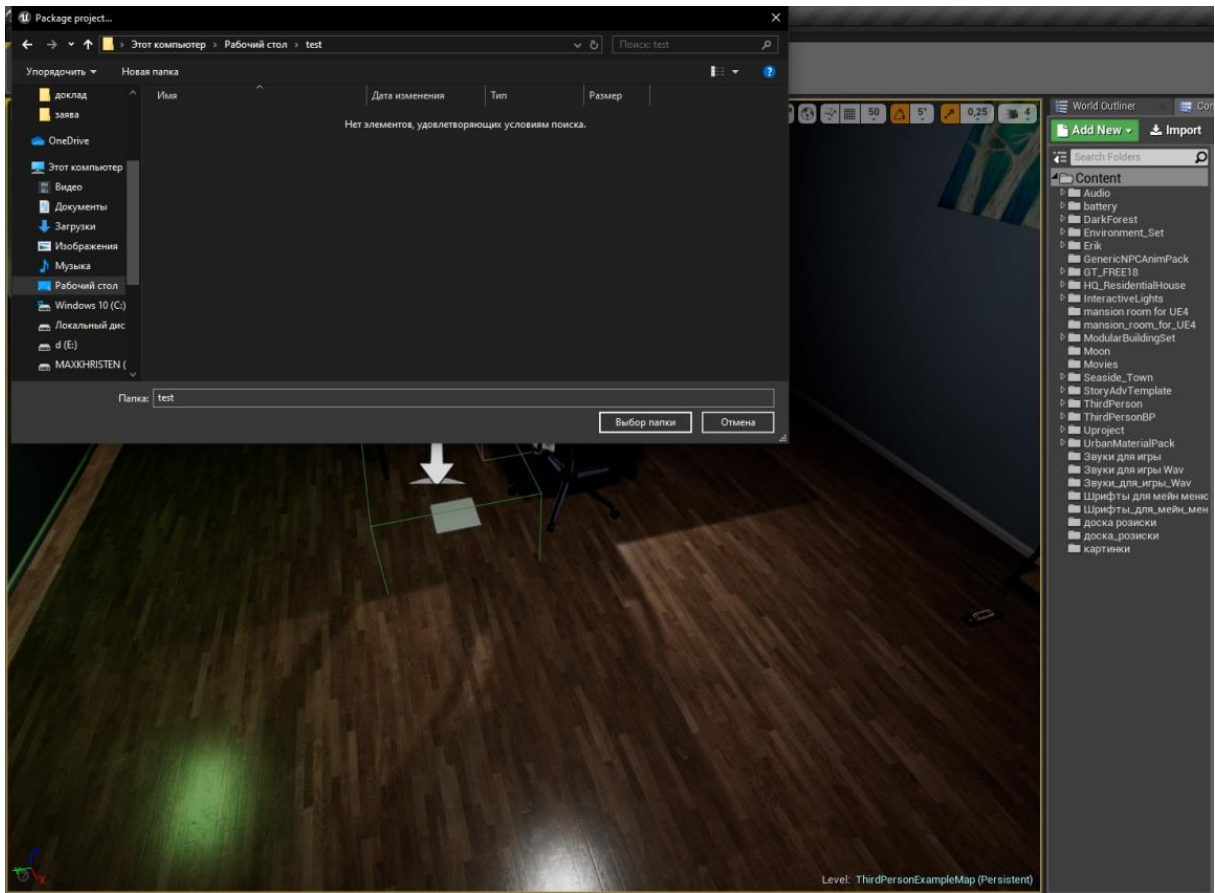


Рисунок 3.41 – Приклад вибору директорії для зберігання

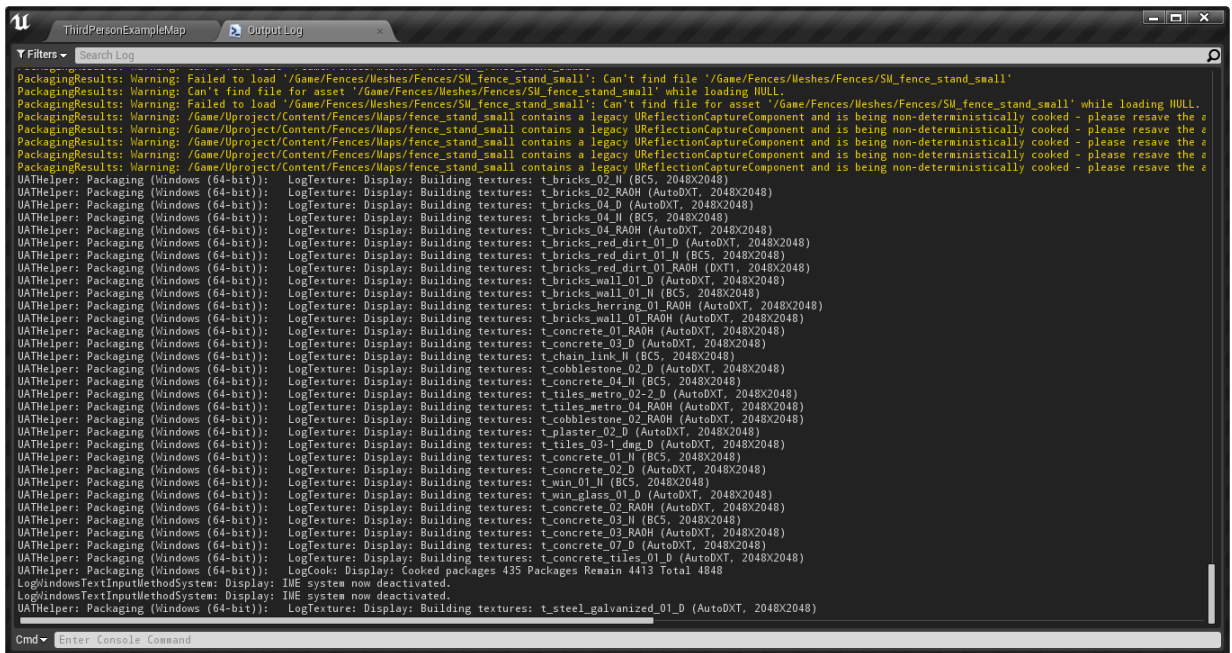


Рисунок 3.42 – Приклад процесу компіляції



## 3.2 Використання додатку

Для початку роботи з ігровим додатком «Wayward pines» необхідно запустити файл типу .exe після відкриття якого з'являється головне меню гри приклад наведений на рис. 3.43.



Рисунок 3.43 – Приклад процесу компіляції

Після того як головне меню завантажилось у гравця є вибір між такими полями меню як:

- Play (грати);
- Load game (Завантажити гру);
- Setting (Налагодження);
- Quit (Вийти).

Після натискання у меню на поле «Play» йде завантаження першого ігрового рівня та починається гра приклад наведений на рис. 3.44.

При натисканні у меню на поле «Load game» йде вибір між збереженими точками гри.

При натисканні у меню на поле «Setting» йде вибір між зміною параметрів якості відео приклад наведений на рис. 3.45.

При натисканні у меню на поле «Quit» ігровий додаток завершує роботу.



Рисунок 3.44 – Приклад процесу компіляції

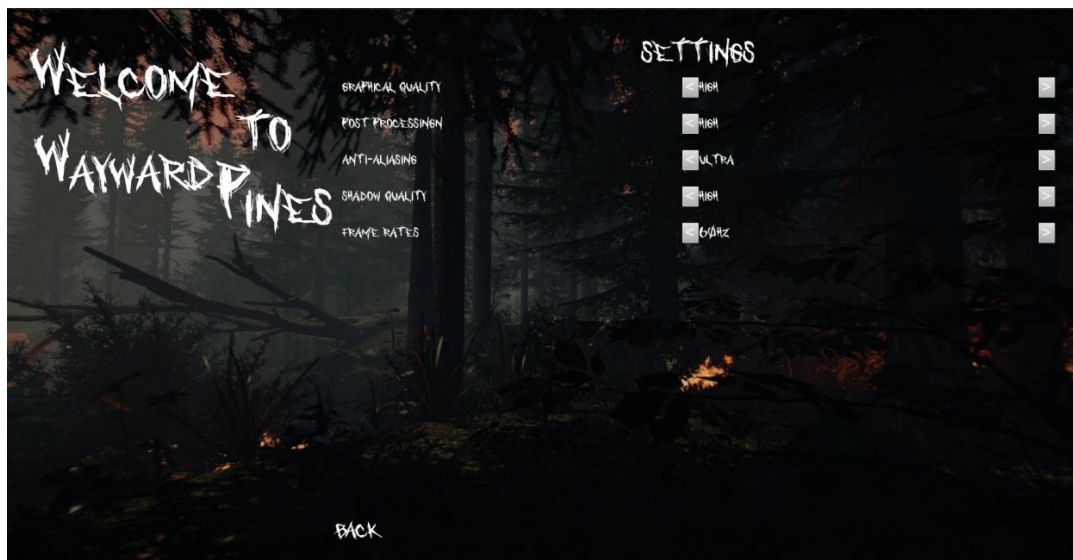


Рисунок 3.45 – Приклад меню зміни якості графіки

Після того, як користувач обрав «Play» та завантажився на ігровий рівень, він може почати рухатися сюжетом. До головного герою заготовлені сюжетні фази типу: «Потрібно відповісти на телефонний дзвінок» або «Потрібно знайти ключ він у цій кімнаті» при пересуванні по рівню.

Дані фрази слугують вказівками для гравця на те, що йому потрібно робити. Також на рівні розташовані елементи з активною взаємодією, які необхідно оглянути й взяти, щоб їх опис можливо було отримати в ігровому журналі, перехід до якого зроблено за допомогою клавіші «J» приклад даного меню наведено на рис. 3.46.

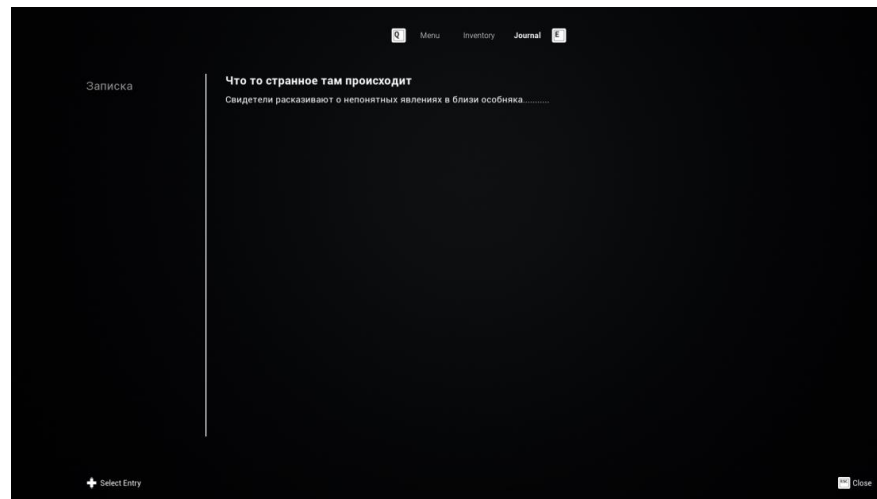


Рисунок 3.46 – Приклад журналу

Якщо гравець захоче оглянути об'єкти типу «Ключ» або інші, він зможе натиснути на клавішу «I», щоб відкрити інвентар та переглянути його приклад наведено на рис. 3.47.

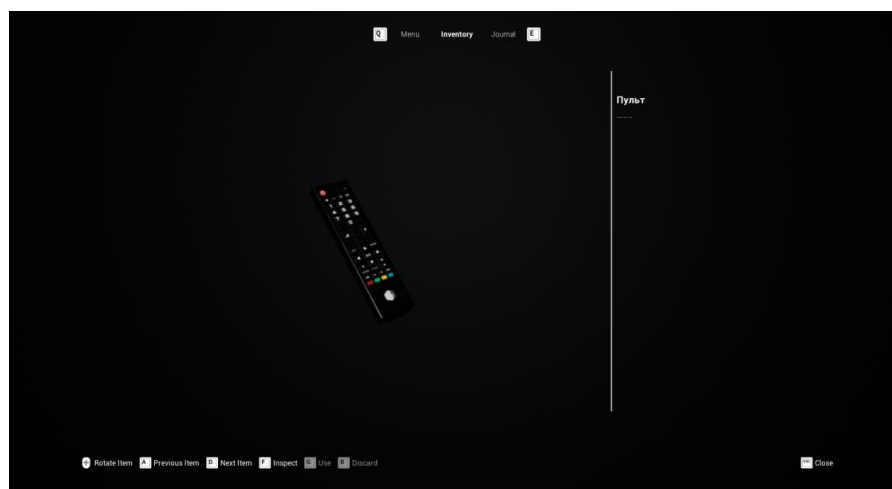


Рисунок 3.47 – Приклад процесу компіляції

Для того, щоб відкрити меню паузи, необхідно натиснути «Esc» з'явиться список дій меню, які можна використати приклад наведений на рис.3.48. А тепер більш конкретно по кожному пункту:

- «Resume» – це дія яка призводить до продовження гри;
- «Inventory» – це дія яка призводить до відкриття інвентарю;
- «Journal» – це дія яка призводить до відкриття журналу;
- «Save Game» – це дія яка призводить до зберігання поточної позиції у грі;
- «Load Game» – це дія яка призводить до загрузки гри зі збереженої позиції;
- «Quit Game» – це дія яка призводить до виходу з гри;

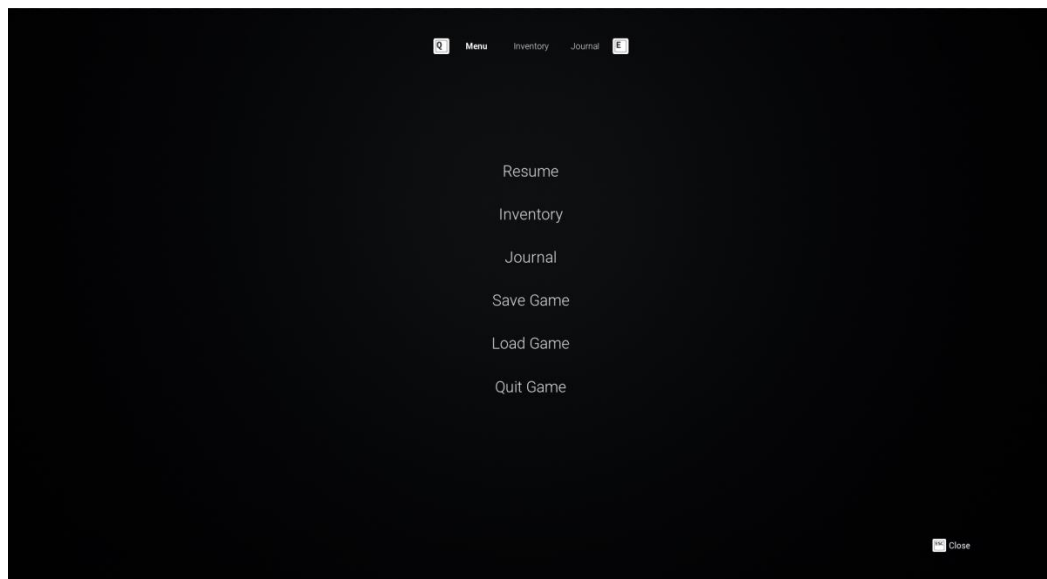


Рисунок 3.48 – Приклад процесу компіляції

## ВИСНОВОК

У ході створення дипломного проекту було виконано мету – розробка комп'ютерної гри «Wayward pines» за допомогою ігрового рушія Unreal Engine 4. У першій частині роботи було проведено огляд останніх досліджень у сфері розробки комп'ютерних ігор. Виявлено, що індустрія не стоїть на одному місці та постійно розвивається, з'являються нові засоби для створення ігрових додатків, а можливість звичайних користувачів отримати доступу до невеликої кількості безкоштовних рушіїв дає змогу відчувати себе автором нової гри.

Був проведений аналіз трьох ігрових додатків, розроблених на різних рушіях, для створення ігор, проведено порівняння за допомогою таблиці оцінювання ігор за критеріями такими як: якість графіки, сюжету, геймплею. Ці критерії виступали головними та було виділено шість допоміжних критеріїв для отримання більш точного оцінювання, яке проводилось на основі відгуків від користувачів сервісу Steam.

Було виявлено, що усі вище наведені ігрові додатки приведені в одному жанрі хоррор, але вони істотно різняться один від одного технологіями реалізації за допомогою різних ігрових рушіїв, мовою програмування, та на сам перед, якістю графіки та обмеженнями, які встановлені ігровими рушіями. У всіх трьох іграх відрізняється повнота прописаного сюжету та розкриття персонажів їх дій та по мірі просування далі по грі.

У другій частині роботи були розглянуті основні етапи моделювання ігрової системи «Wayward pines», по кожному з них була оброблена довідкова інформація та виходячи з неї, були побудовані діаграми, на яких показано варіанти використання ігрової системи.

Отже на основі вивченої інформації та створених діаграм можна докладно подивитися на принципи роботи основних компонентів ігрової системи та на те як вони взаємодіють між собою та користувачем у ході ігрового процесу.

У третій частині було описано головні етапи у роботі з ігровим рушієм Unreal Engine 4, приведено створення 3д моделей, зроблений звукозапис голосу та звуків оточення для надання відповідної атмосфери. Перед безпосереднім додавання вище наведених елементів у UE4, та налагодження їх у самому рушії. Також було проведено налагодження текстур для створених моделей, спроектовано та відтворено два ігрових рівня, та розроблені головні механіки гри та розроблені елементи управління й перемикання стану камери між першою та третьою особою. Розроблений програмний код для взаємодії між гравцем та елементами на ігровій мапі.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Система оцінок комп'ютерних ігор [Електронний ресурс]. – Режим доступу: [http://gamesisart.ru/theory\\_game\\_score.html](http://gamesisart.ru/theory_game_score.html) (дата звернення: 03.04.2020).
2. Система оцінок комп'ютерних ігор. Стаття 2 [Електронний ресурс]. – Режим доступу: [http://gamesisart.ru/theory\\_game\\_score\\_2.html](http://gamesisart.ru/theory_game_score_2.html) (дата звернення: 03.04.2020).
3. Жанри комп'ютерних ігор: Класифікація комп'ютерних ігор по жанрам [Електронний ресурс]. – Режим доступу: <http://fb.ru/article/179293/janryi-kompyuternyih-igr-spisok-klassifikatsiya-kompyuternyihigr-po-janram> (дата звернення: 09.04.2020).
4. Основні принципи розробки ігор [Електронний ресурс]. – Режим доступу: <https://habr.com/post/188372/> (дата звернення: 10.04.2020).
5. Кращі ігрові рушії для розробки ігор [Електронний ресурс]. – Режим доступу: <https://senfil.net/index.php?newsid=321> (дата звернення: 10.04.2020).
6. Ігри які змінили бачення жанру [Електронний ресурс]. – Режим доступу: <https://gmbox.ru/materials/41778-10-igr-navsegda-izmenivshih-tselie-zhanri>. (дата звернення: 10.04.2020).
7. Гра Scare [Електронний ресурс]. – Режим доступу: [https://www.playground.ru/files/igra\\_scare\\_ispravlennaya-63214/](https://www.playground.ru/files/igra_scare_ispravlennaya-63214/). (дата звернення: 10.04.2020).
8. Гра The Story of Henry Bishop [Електронний ресурс]. – Режим доступу: [https://store.steampowered.com/app/1116230/The\\_Story\\_of\\_Henry\\_Bishop](https://store.steampowered.com/app/1116230/The_Story_of_Henry_Bishop). (дата звернення: 10.04.2020).
9. Гра Ridge [Електронний ресурс]. – Режим доступу: <https://store.steampowered.com/app/492600/Ridge>. (дата звернення: 10.04.2020).

10. Unified Modeling Language [https \[Електронний ресурс\]](https://uk.wikipedia.org/wiki/Unified_Modeling_Language). – Режим доступу:[//uk.wikipedia.org/wiki/Unified\\_Modeling\\_Language](https://uk.wikipedia.org/wiki/Unified_Modeling_Language)
11. Діаграма станів автомата [Електронний ресурс]. – Режим доступу:[https://uk.wikipedia.org/wiki/Діаграма\\_станів\\_автомата](https://uk.wikipedia.org/wiki/Діаграма_станів_автомата)
12. Скінченний автомат [Електронний ресурс]. – Режим доступу:[https://uk.wikipedia.org/wiki/Скінченний\\_автомат](https://uk.wikipedia.org/wiki/Скінченний_автомат)
13. Об'єкт (програмування) [Електронний ресурс]. – Режим доступу:[https://uk.wikipedia.org/wiki/Об%27єкт\\_\(програмування\)](https://uk.wikipedia.org/wiki/Об%27єкт_(програмування))
14. Збірник тез квітневої конференції [Електронний ресурс]. – Режим доступу:  
<https://drive.google.com/file/d/1mayREf5nE56WrpZqS10qJrwj3mMTcqdG/view> (дата звернення: 21.04.2020)
15. Triggering Sequences from Gameplay [Електронний ресурс]. – Режим доступу:  
<https://docs.unrealengine.com/enUS/Engine/Sequencer/HowTo/TriggeringSequences/index.html> (дата звернення: 26.04.2020)
16. Blending Animation Blueprints with Sequencer [Електронний ресурс]. – Режим доступу:  
<https://docs.unrealengine.com/en-US/Engine/Sequencer/HowTo/BlendingAnimBPs/index.html> (дата звернення: 01.05.2020)
17. Unreal Projects and Gameplay [Електронний ресурс]. – Режим доступу:  
<https://docs.unrealengine.com/en-US/Gameplay/UnrealTerminology/index.html> (дата звернення: 15.05.2020)
18. User Interfaces & HUDs [Електронний ресурс]. – Режим доступу:  
<https://docs.unrealengine.com/en-US/Gameplay/Framework/UIAndHUD/index.html> (дата звернення: 25.05.2020)



# ДОДАТОК А

## ТЕХНІЧНЕ ЗАВДАННЯ на розробку додатку “Wayward pines”

2020

## 1.1 Основи для розробки

Розробка виконується на підставі індивідуального завдання до дипломного проекту бакалавра, виданого керівником Федотова Н.А.

Реалізація обумовлена створенням комп'ютерної гри жанру Survival horror під назвою "Wayward pines".

Тема дипломного проекту: «Створенням комп'ютерної гри жанру Survival horror під назвою "Wayward pines"».

Умовне позначення розроблюваної системи – Wayward pines.

## 1.2 Короткий опис сюжету гри для розробки першого рівня

Короткий опис сюжету гри:

Йшов 2018 рік у місті Портланд штату Мен. Саме тут розташований покинутий багато років тому маєток під назвою Wayward pines. Це місце не люблять місцеві, дивні події стаються тут. Але околиці притягують людей як магніт, сліди перебування яких завжди є на прилеглий території біля маєтку. Але, завжди лише сліди... Тут зникають люди.

Детектива Ітана Шоу наймає одна із сімей чия дитина зникла.

Опис головного героя:

- 1) Ім'я–Ітан Шоу;
- 2) Професія– Приватний детектив;

Після інциденту під час служби у поліції який призвів до звільнення з структури охорони правопорядку почав займатися приватною діяльністю та брався за майже кожен випадок зникнення чи викрадення людей, у сфері пошуку йому не було рівних.

Рівень перший (Квартира головного героя)

Опис локації: Квартира яка складається з чотирьох кімнат таких як:

- 1) Головна кімната суміщена с кухнею;

2)Кабінет головного героя;

3)Спальна кімната;

4)Туалет.

Початок(Поява персонажа на рівні)

Персонаж прокинувся у спальній кімнаті, гравець почув телефонний дзвінок у кабінеті.

Персонаж вийшов з спальної кімнати й проговорив «..Чому знову так рано, хтось не вміє користуватись годинником... знову».

«Все ж таки я відповім..» гравець починає йти у кабінет та підходить до столу з телефоном і відповідає:

– Емм, так? так, це Шоу. Уважно слухаю

– Доброго дня! Емм, ранку, вибачте. Мені порекомендували Вас... Я не розумію , що мені робити... Ви, гарний детектив, мені сказали....

– Зазвичай так. Що саме від мене необхідно?

– Хайкінг. Це був хайкінг з родиною. А там масток, ці крики , вони були чутні за милю. Ми хотіли допомогти... Але потім, Джейк ... Наш син... Він просто...

– Що він? Що сталося з хлопчиком місс?

– Він стояв позаду нас, він був наляканий криками... І тут грохот впавших речей. він тримав їх у руках. Ми шукали його, але.. ні сліду ... Копи також не знайшли його

– Як давно це трапилось, місс?

– Близько 12 годин... Допоможіть, я благаю Вас...

– Так, я зроблю все, що в моїх силах. Мені потрібна година на збори та і дорога не близька, десь опівночі я буду на місці...

– Дякую, допоможіть знайти сина, ми віддячимо сповна....

Дзвінок завершився персонаж встає – гравцю потрібно взяти ключі та направитися до двері та відчинити її.

Кінець(Завантаження наступного рівня)

### 1.3 Вимоги до програмного продукту

При реалізації і використанні ігрової системи повинні бути враховані вимоги до функціональних характеристик, надійності проекту, параметрів технічних засобів, інформаційної та програмної сумісності. З метою вирішення проблем були запропоновані та встановлені вимоги до гри «Wayward pines» таблиця А.1.

Таблиця А.1 – Перелік вимог до гри «Wayward pines».

№	Назва	Зміст	Тип	Зміст тесту
1	Дотримання певних стандартів при розробці гри	Використання нормативних документів які регулюють порядок створення додатку	Вимога до системи	Перевірити дотримання до стандартів та моделей розробки ігор
2	Розробка якісних моделей	Моделі які будуть використані у грі повинні бути високої якості	Вимога до компонент та	Масштаб та текстура моделі повинні співпадати та бути у високого розширення
3	Запис звуку	Звук повинен бути студійної якості для уникнення шумів та артефактів звуку	Вимога до компонент та	Перевірити звук на чистоту та відсутність рипіння
4	Розробка сюжету гри	Сюжет повинен бути логічно завершеним та без непорозуміння	Вимога до компонент та	Перечитати та перевірити на логічну завершеність
5	Оптимізація і тестування	Гра повинна підходити під системні вимоги персонального комп'ютеру	Вимога користувача	Підібрати комп'ютер який підходить під мінімальні системні вимоги гри та перевірити працездатність та ресурс ємність

### 1.4 Вимоги до функціональних характеристик

Програмний продукт повинен бути розрахований на кінцевого користувача.

Основні вимоги, яким повинен відповідати програмний продукт:

- Зручна робота користувача з грою;
- Якісні моделі;
- Записаний якісний звук;
- Розробка повноцінного (закінченого) сюжету гри.

Ігрова система повинна виконувати наступні функції:

- Відображати закладені ігрові функції;
- безвідмовно функціонувати;
- надавати можливість коректного відображення інтерфейсу користувача;
- надавати можливість коректного відображення предметів на рівні;
- надавати можливість вибору між елементами на рівні;

### 1.5 Вимоги до надійності

Програмний продукт повинен безвідмовно функціонувати і не приводити до збоїв операційної системи в аварійних ситуаціях. У разі виникнення збою повинні видаватися коректні повідомлення із зазначенням подальших дій.

Надійне функціонування програмного комплексу буде забезпечуватися при використанні сучасних ЕОМ, чіткому дотриманні рекомендацій. Забороняється видаляти будь-які файли проекту, доступ до них повинен бути обмежений.

Розташування елементів на головній сторінці сайту схематично показано на рис. А.1.

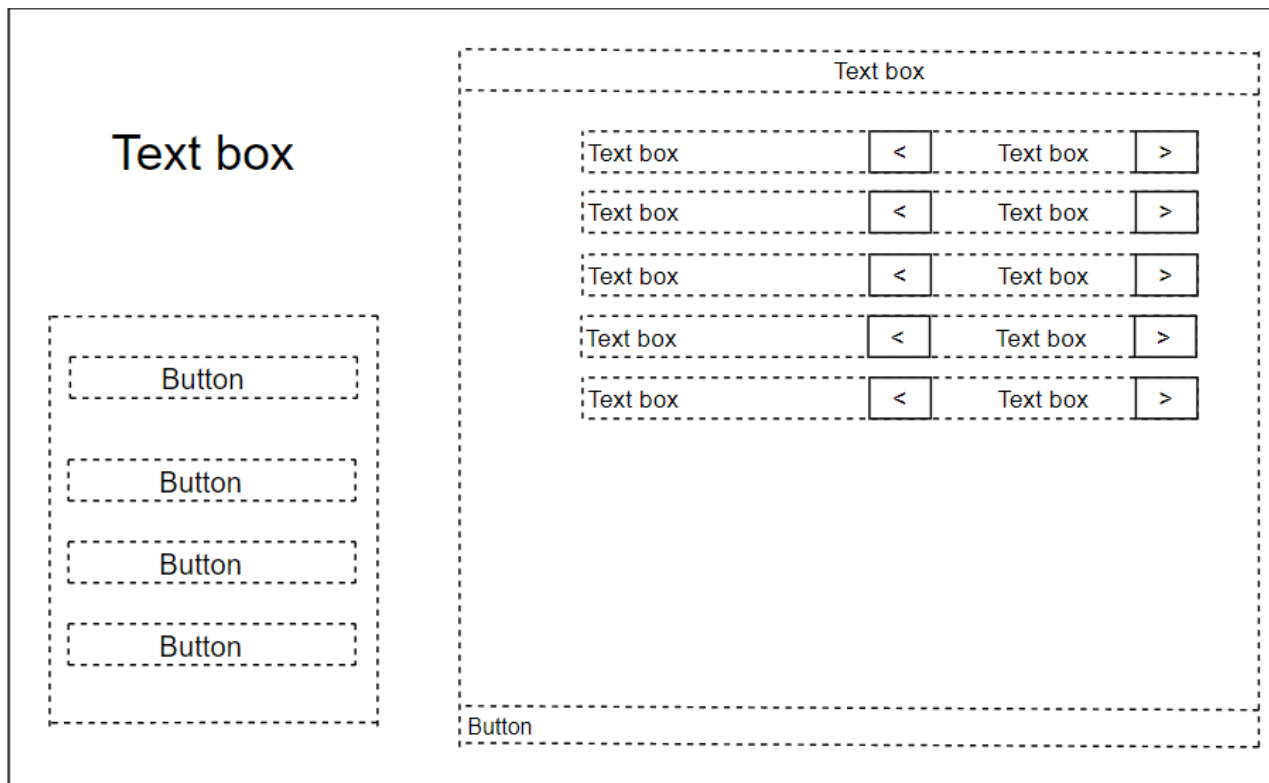


Рисунок А.1 – Приклад меню настройок гри

Типові навігаційні й інформаційні елементи що розташовані на головному меню у вкладці настройок:

- Шапка гри;
- Основне меню;
- Меню настройок;
- Керуючі кнопки.

Шапка гри зберігає у собі лише назву гри яка додана за допомогою елементу «Text Box». Основне меню зберігає у собі чотири керуючих елементи типу «Button» за допомогою яких здійснюється навігація усередині гри. Меню настройок відповідно здійснюється зміна якості графіки та рендеру при натисканні на елементи типу «Button».

## 1.6 Вимоги до складу і параметрів технічних засобів для Windows версії

Мінімальні вимоги до технічних засобів для нормального функціонування програми:

- процесор (мінімальні вимоги: I3-6500 – 3.0ГГц);
- оперативна пам'ять (мінімальні вимоги: 8 GB);
- відео-карта (мінімальні вимоги: відео-пам'ять – 3 GB);

### 1.7 Стадії і етапи розробки

На певних етапах розробки ігрової системи перед впливають індивідуальні завдання які потрібно виконати за допомогою певних програмних додатків.

На етапі №1 «Технічний проект» для досягнення поставленої цілі необхідно створити за допомогою додатку «Draw IO» схеми UML: діаграми прецедентів, діаграми класів і послідовності.

На етапі №2 «Робочий проект» для досягнення поставленої цілі потрібно використати текстовий редактор для написання сюжету гри, для розробки моделей використати додаток «3Ds Max», для звукозапису додаток «Audacity» та безпосередньо для створення гри рушій «Unreal Engine 4».

Розробка ведеться в декілька кроків, і включає етапи, наведені в таблиці А.2.

Таблиця А.2 – Етапи розробки

	Етап/Термін виконання	Зміст робіт
1	Технічний проект 15.03.20 – 10.04.20	Аналіз і формалізація вимоги до ПК гри "Wayward pines". Розробка проекту комп'ютерної гри жанру Survival horror під назвою "Wayward pines" з використанням UML: діаграми прецедентів, діаграми класів і послідовності.
2	Робочий проект 10.04.20 – 11.05.20	Розробка програмної частини комп'ютерної гри "Wayward pines". Розробка сценарію гри, коду, моделей оточення, взаємодії на рівні, створення головного та

		ігрового меню для керування грою, проведення тестування та налагодження роботи гри.
--	--	---

Продовження таблиці А.2 – Етапи розробки

3	Оформлення 11.05.20 – 17.05.20	Оформлення текстової, програмної та графічної документації комп'ютерної гри "Wayward pines".
4	Тестування додатку 17.05.20 – 25.05.20	Розробка заходів тестування додатку
5	Впровадження 25.05.20 – 01.06.20	Розробка заходів впровадження комп'ютерної гри "Wayward pines" до використання.

### 1.8 Порядок контролю і приймання

Ухвалення створеного програмного комплексу полягає в тестуванні після настройки програмного продукту.

Контроль здійснюється кінцевими користувачами системи, підключеними на етапі тестування системи.

Прийом програмного комплексу здійснюється після його створення.

По завершенні розробки системи повинні бути проведені наступні види випробувань:

- тестування на захист від некоректного введення;
- тестування на захист від несанкціонованого використання;
- тестування на безвідмовну роботу та кількість помилок;
- перевірка на відповідність до системних вимог.



## ДОДАТОК Б

### 1. Планування робіт

#### 1.1 Планування змісту робіт

**Деталізація мети проекту методом SMART.** Продуктом дипломного проекту є гра «Wayward pines».

Результати деталізації методом SMART.

**S.** Розробити комп'ютерну гру «Wayward pines»

**M.** Результатом виконання проекту, має бути комп'ютерна гра, ігровий додаток мусить бути розміщений на жорсткому диску комп'ютера, пов'язані з ним дані та модулі мають бути вірно встановлені налаштовані та поєднані.

**A.** Проект потребує розробника із знанням Blueprints, Unreal Engine 4, мови програмування C++, та програмного забезпечення (ПЗ), яке використовується для розроблення даного додатку.

**R.** Мету реально досягнути, так як розробка Ігрових-додатків за допомогою сучасних ігрових двигунів, не вимагає надзвичайно складних дій але вимагає велику кількість ресурсів для розробки.

**T.** Обмеженість в часі зумовлена рішенням замовника, щоб отримати ігровий додаток як можна швидше.

Структурна декомпозиція робіт (work breakdown structure, WBS) - це ієрархічна структура робіт, побудована з метою логічного розподілу усіх робіт з виконання проекту і подана у графічному вигляді. Це сукупність декількох рівнів, кожний з яких формується в результаті розподілу роботи попереднього рівня на її складові. Елементом найнижчого рівня є група робіт, або так званий робочий пакет (work package).

WBS, створення гри «Wayward pines» представлений на рис. Б.1

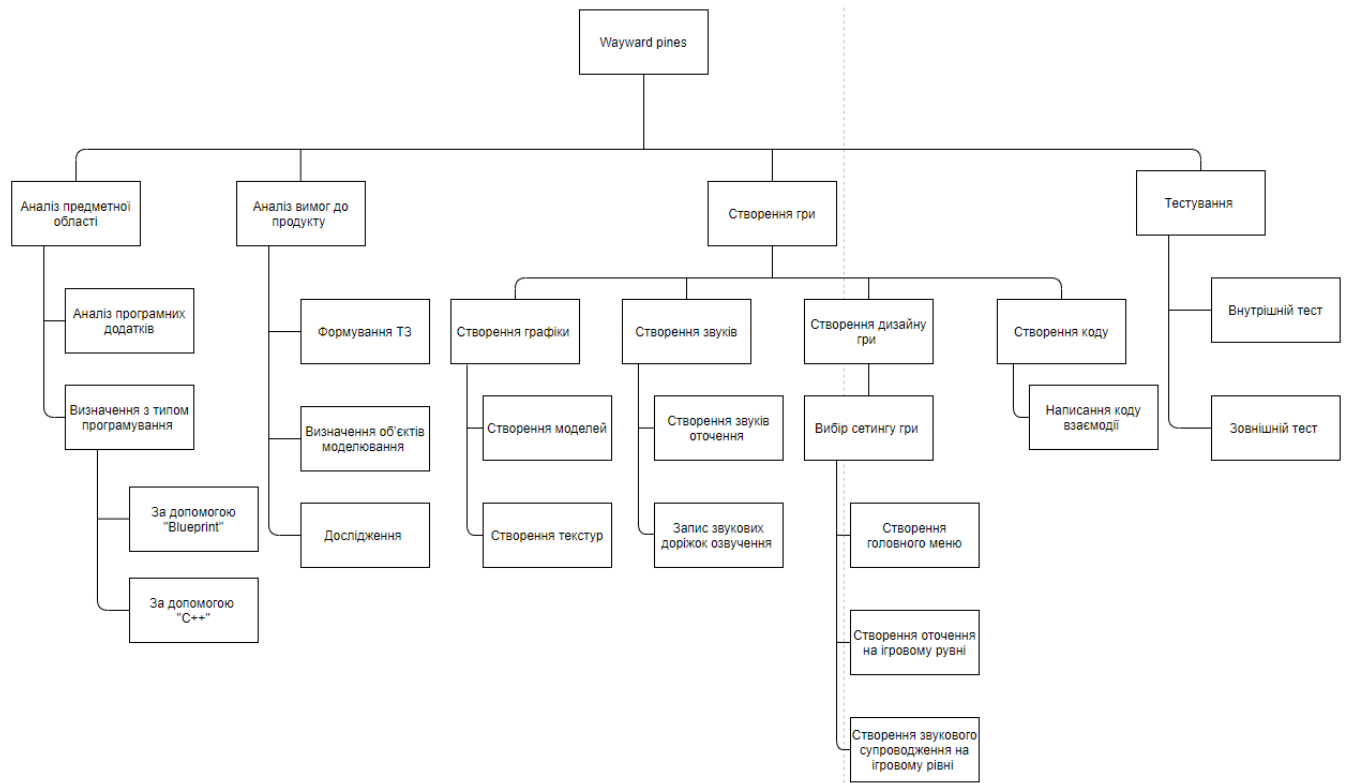


Рисунок Б.1 – WBS Створення гри «Wayward pines»

## 1.2 Планування структури виконавців

Наступним кроком розробки структури проекту є визначення організаційної структури (OBS) проекту.

Організаційна структура проекту (OBS) – є графічним відображенням учасників проекту (фізичних та юридичних осіб) та їхніх відповідальних осіб, залучених до реалізації проекту. На верхньому рівні OBS проекту знаходиться керівник та команда управління проектом; на наступному рівні – виконавці. Останнім рівнем OBS-структури є відповідальні особи виконавців. Це не обов'язково повинні бути керівники, а ті співробітники, яким доручено безпосередньо організувати і відповідати перед виконавцем за виконання конкретного елемента WBS-структури.

OBS, Створення гри «Wayward pines» представлений на рис. Б.2.

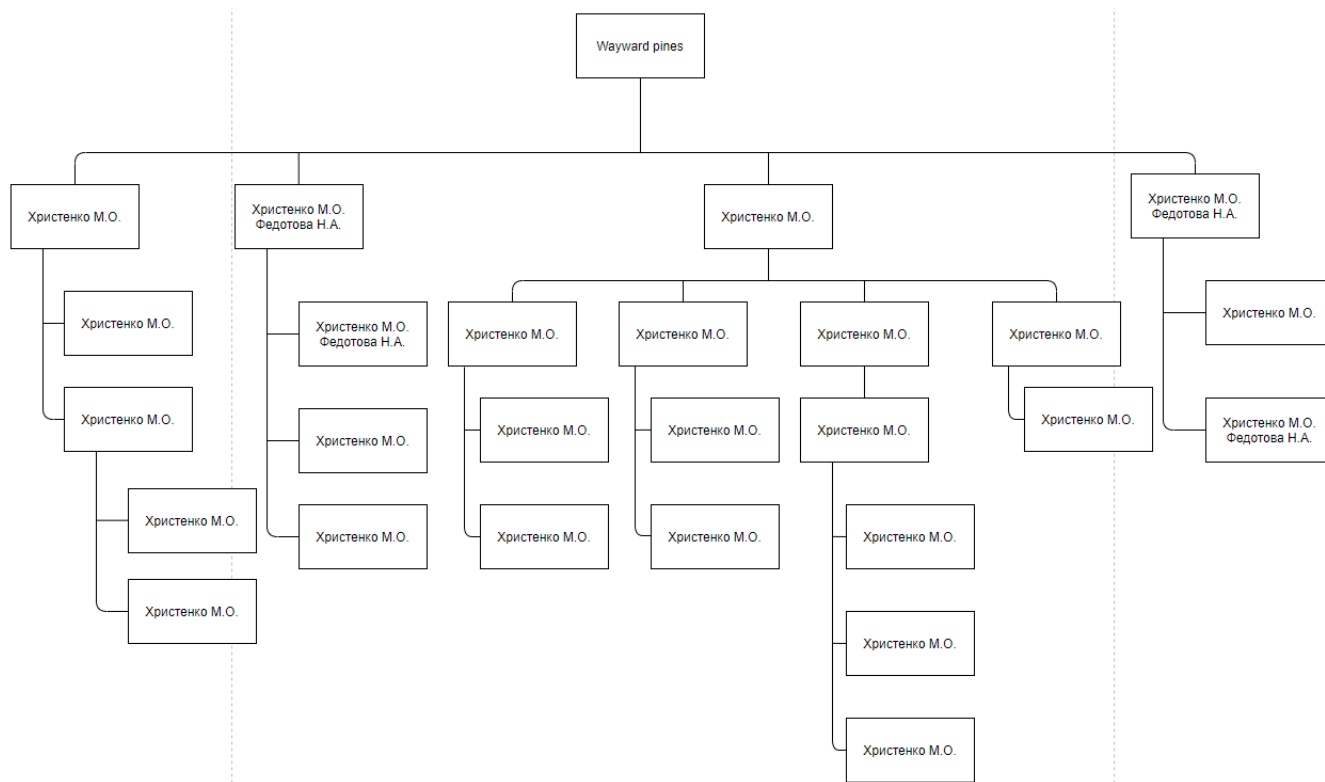


Рисунок Б.2 – OBS структура

### 1.3 Побудова матриці відповідальності

Матриця відповідальності (Responsibility Assignment Matrix) забезпечує опис і узгодження структури відповідальності за виконання пакетів робіт. Вона являє собою форму опису розподілу відповідальності за реалізацію робіт проекту із зазначенням ролі кожного з виконавців. Будується на основі WBS та OBS.

У таблиці Б.1 показано матрицю відповідальності проекту.

Таблиця Б.1 – RAM

№	Title	Христенко М.О	Щендрик В.В.	Федотова Н.А.
1	Аналіз програмних додатків	<b>R</b>	<b>A</b>	<b>I</b>
2	Визначення з типом програмування	<b>R</b>		<b>C</b>
3	Формування ТЗ	<b>R</b>	<b>I</b>	<b>C</b>
4	Визначення об'єктів моделювання	<b>R</b>		<b>C</b>
5	Створення моделей	<b>R</b>		<b>I</b>
6	Створення текстур	<b>R</b>		<b>I</b>
7	Запис звуків оточення	<b>R</b>		<b>I</b>
8	Запис звукових доріжок озвучення	<b>R</b>		<b>I</b>
9	Вибір сетингу гри	<b>R</b>		<b>I</b>
10	Створення дизайну головного меню	<b>R</b>		<b>I</b>
11	Створення оточення на ігровому рівні	<b>R</b>		<b>I</b>
12	Створення звукового супроводження на ігровому рівні	<b>R</b>		<b>I</b>
13	Написання коду для взаємодії між елементами	<b>R</b>		<b>I</b>
14	Внутрішній тест	<b>R</b>	<b>A</b>	<b>I</b>
15	Зовнішній тест	<b>R</b>	<b>A</b>	<b>C</b>

№	Code:	Stands for:
1	R	Responsible
2	A	<b>Accountable</b>
3	I	Inform
4	C	Consult

## 1.4 Побудова календарного графіку

Діаграма Ганта – це популярний вид діаграми який використовується для планування і контролю виконання проекту. Такий інтерактивний мережевий графік присутній практично у всіх системах управління проектами.

На діаграмі відображаються завдання і стадії проекту з урахуванням їх часу виконання. Завдання на діаграмі можуть бути залежними один від одного (наприклад, одна задача може починатися тільки після завершення другого). Крім того, може показуватися відсоток виконання кожного завдання і відповідальний за її виконання.

Діаграма Ганта представлена на рис. Б.4

★	▲ Гра Wayward pines	100 дней?	Чт 12.12.19	Ср 29.04.20		
★	▲ Аналіз предметної області	5 дней	Чт 12.12.19	Ср 18.12.19		
★	Аналіз програмних дода	5 дней	Чт 12.12.19	Ср 18.12.19		
★	Визначення з типом програмування	5 дней	Чт 12.12.19	Ср 18.12.19		
★	▲ Аналіз вимог до продукту	5 дней	Чт 19.12.19	Ср 25.12.19		
★	Формування ТЗ	5 дней	Чт 19.12.19	Ср 25.12.19	3;4	
★	Визначення об'єктів моделювання	5 дней	Чт 19.12.19	Ср 25.12.19	3;4	
★	▲ Створення гри	70 дней	Чт 26.12.19	Ср 01.04.20		
★	▲ Створення графіки	15 дней	Чт 26.12.19	Ср 15.01.20		
★	Створення моделей	10 дней	Чт 26.12.19	Ср 08.01.20	6;7	
★	Створення текстур	5 дней	Чт 09.01.20	Ср 15.01.20	10	
★	▲ Створення звуків	15 дней	Чт 16.01.20	Ср 05.02.20		
★	Запис звуків оточення	8 дней	Чт 16.01.20	Пн 27.01.20	11	
★	Запис звукових доріжок озвучення	7 дней	Вт 28.01.20	Ср 05.02.20	13	
★	▲ Створення дизайну гри	20 дней	Чт 06.02.20	Ср 04.03.20		
★	Створення дизайну головного меню	10 дней	Чт 06.02.20	Ср 19.02.20	14	
★	Створення оточення на ігровому рівні	8 дней	Чт 20.02.20	Пн 02.03.20	16	
★	Створення звукового супроводження на ігровому рівні	2 дней	Вт 03.03.20	Ср 04.03.20	17	
★	▲ Створення програмного коду	20 дней	Чт 05.03.20	Ср 01.04.20		
★	Написання коду для взаємодії між елементами	20 дней	Чт 05.03.20	Ср 01.04.20	18	
★	▲ Тестування	20 дней	Чт 02.04.20	Ср 29.04.20		
★	Внутрішній тест	10 дней	Чт 02.04.20	Ср 15.04.20	20	
★	Зовнішній тест	10 дней	Чт 16.04.20	Ср 29.04.20	22	

Рисунок Б.3 – Перелік виконаних робіт

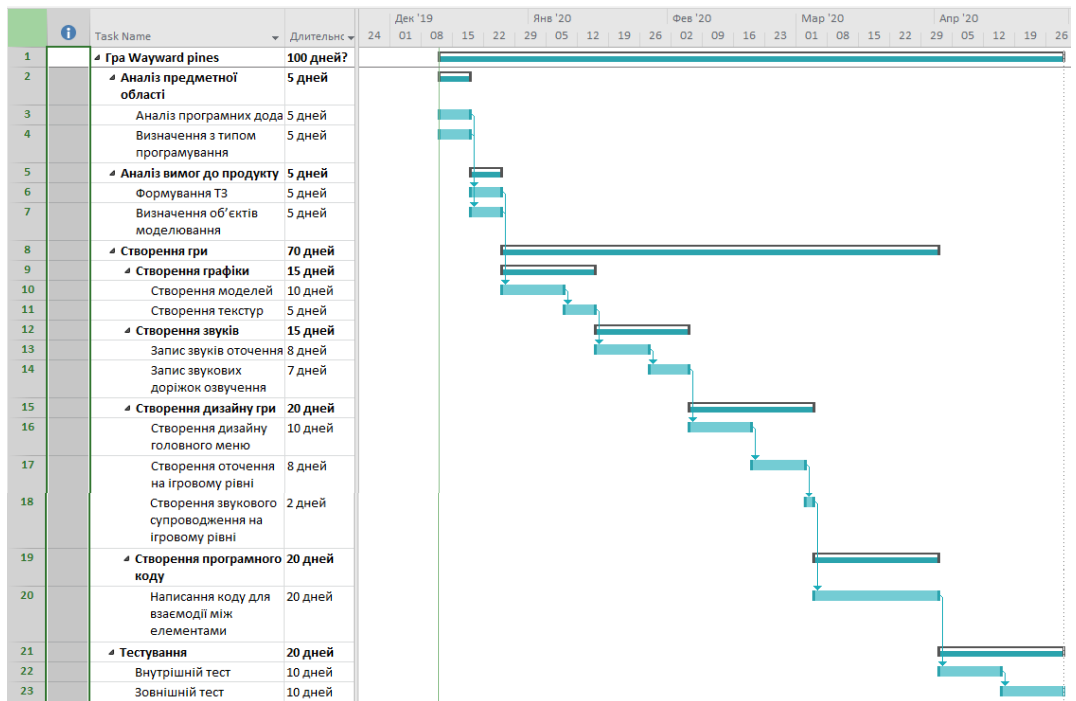


Рисунок Б.4 – Діаграма Ганта

## 1.5 Управління ризиками проекту

Ризик – це імовірна подія, яка у випадку своєї появи негативно або позитивно вплине на проект.

Процес управління ризиками включає наступні етапи:

1. ідентифікація
2. процес оцінювання ризиків, який включає в себе якісний, кількісний аналіз.
3. заходи реагування на ризики
4. моніторинг заходів і ризиків

Ризики проекту представлені у таблиці Б.2

Таблиця Б.2 – Risk Register

№	Опис ризика	Вплив	Вірогідність	R/V	Зменшення впливу
1	Необхідність доволіно скоротити тривалість завдання і / або запустити завдання паралельно, що збільшить ризик помилки	2	2	L	Показати зацікавленим сторонам план проекту і пояснити, що скорочення часу виконання замовлення може привести до неминучих помилок і низької якості продукції
2	Низька залученість клієнтів	1	1	L	Пояснити клієнту, що чим вище його залученість в проект, тим краще продукт, який він отримає
3	Проект не визначено	4	2	M	Завершіть економічне обґрунтування, якщо ще не надали, і переконайтеся, що мета чітко визначена
4	Незаплановані роботи, які будуть виконані на пізній стадії проекту	4	3	M	документувати всі припущення, зроблені при плануванні, повідомити менеджера проекту до початку проекту
5	Складні або неможливі завдання	5	4	H	Оновити штат або замінити виключити завдання
6	Втрата даних при помилках у роботі системи	5	3	H	Спробувати зробити відкат системи або найняти фахівців для відновлення даних

Таблиця Б.3 – Probability / Impact Matrix

<b>5</b>					
<b>4</b>					<b>5</b>
<b>3</b>				<b>4</b>	<b>6</b>
<b>2</b>		<b>1</b>		<b>3</b>	
<b>1</b>	<b>2</b>				
<b>Вплив/Вірогідність</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>

Рівні критичності:

- 1 Negligible
- 2 Low
- 3 Medium
- 4 High
- 5 Critical