

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

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

**на тему: «Інтерактивний квест додаток для вивчення історії
Воскресенської церкви м.Лебедин»**

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

Виконавець роботи: студент групи ІТ.м-91 Дементієнко Дмитро Сергійович

**Кваліфікаційну роботу
захищено на засіданні ЕК
з оцінкою**

«__» грудня 2020 р.

Науковий керівник

(підпис)

к.т.н., доц., Баранова І.В.

Голова комісії
(підпис)

Шифрін Д.М.

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.

ЗАВДАННЯ
на кваліфікаційну роботу магістра студентіві

Дементієнко Дмитро Сергійович
(прізвище, ім'я, по батькові)

1 Тема проекту Інтерактивний квест додаток для вивчення історії Воскресенської церкви м. Лебедин

затверджена наказом по університету від «26» листопада 2020 р. № 1824-III

2 Термін здачі студентом закінченого проекту «09» _____ грудня _____ 2020 р.

3 Вхідні дані до проекту _____ методології створення ігрових додатків, технології інтерактивності, методології навчання за допомогою квест-ігор _____.

4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) аналіз предметної області, постановка задачі та методи дослідження, планування роботи, структурно функціональний аналіз проекту, підготовка 3D моделей до імпорту у рушій, підготовка ігрової сцени, реалізація програмних зв'язків між об'єктами, тестування

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) результат дослідження аналогів, етапи проектування, структурно функціональний аналіз проекту, діаграма варіантів використання, практична реалізація, демонстрація роботи додатку

6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання _____.

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів випускної проекту	Термін виконання етапів проекту	Примітка
1.	Ідентифікація	03.09.20 – 10.09.20-	
2.	Визначення вимог	11.09.20 – 18.09.20	
3.	Створення WBS	19.09.20 – 22.09.20	
4.	Створення OBS	23.09.20 – 24.09.20	
5.	Визначення ризиків	30.09.20 – 01.10.20	
6.	Імпорт 3D моделей	02.10.20 – 06.10.20	
7.	Розташування 3D моделей в локації	07.10.20 – 10.10.20	
8.	Створення дизайну персонажу	11.10.20 – 13.10.20	
9.	Налаштування анімації персонажу	14.10.20 – 17.10.20	
10.	Створення інтерфейсу	23.10.20 – 28.10.20	
11.	Створення квест завдань	29.10.20 – 24.11.20	
12.	Апробація та імплементація	02.12.20 – 09.12.20	
13.	Здача документації	09.12.20 – 10.12.20	
14.	Архівація	10.12.20 – 11.12.20	

Магістрант _____

Дементієнко Д.С.

Керівник роботи _____

к.т.н., доц. Баранова І.В.

РЕФЕРАТ

Тема кваліфікаційної роботи магістра «Інтерактивний квест додаток для вивчення історії Воскресенської церкви м. Лебедин».

Пояснювальна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел із 31 найменування, додатків. Загальний обсяг роботи – 74 сторінки, у тому числі 61 сторінка основного тексту, 2 сторінки списку використаних джерел, 10 сторінок додатків.

Кваліфікаційну роботу магістра присвячено розробці ігрового квест додатку для ознайомлення з історією архітектурної пам'ятки - Воскресенської церкви.

В роботі проведено огляд методологій для створення квест завдань та досліджено інноваційність ігрових квест додатків у сучасному освітньому процесі. Проведено аналіз схожих за функціоналом додатків, для усунення слабких частин проекту та удосконалення сильних сторін,

Визначені етапи виконання проекту, проведено структурно-функціональний аналіз, створено діаграму використання, проведено планування робіт проекту.

Створено ігрову локацію для реалізації квесту із необхідними моделями. Розроблено зручний інтерфейс квест додатку та керування ігровим персонажем, логіку роботи додатку, запрограмовані необхідні реакції додатку на дії користувача.

Результатом проведеної роботи є ігровий квест додаток, який дозволяє перевірити та покращити свої знання щодо історії Воскресенської церкви. Практичне значення роботи полягає у використанні квест додатку під час навчальних заходів з краєзнавства в освітніх закладів міста Лебедин.

Ключові слова: 3D модель, ігровий рушій, інтерфейс, анімація, квест-завдання, blueprint, .

ЗМІСТ

Вступ.....	6
1 Аналіз предметної області.....	8
1.1 Огляд проблеми.....	8
1.2 Огляд аналогів.....	12
1.1.1 Outer wilds.....	12
1.1.2 Triviador.....	13
1.1.3 Quest hunter.....	15
1.3 Результат дослідження аналогів.....	16
2 Постановка задачі дослідження.....	18
2.1 Мета та задачі.....	18
2.2 Вибір інструментів реалізації.....	19
3 Проєктування розробки квест додатку.....	22
3.1 Етапи проєктування.....	22
3.2 Структурно-функціональний аналіз проєкту в нотації IDEF0.....	24
3.3 Створення діаграми варіантів використання.....	30
4 Реалізація інтерактивного додатку.....	32
4.1 Підготовка 3D моделей до імпорту.....	32
4.2 Розробка ігрової локації у рушії.....	33
4.3 Налаштування ігрового персонажу.....	43
4.4 Розробка інтерфейсу додатку.....	46
4.5 Реалізація квест-завдань.....	48
4.6 Тестування роботи додатку.....	57
Висновки.....	61
Список використаних джерел.....	62
Додаток А. Планування робіт.....	65
Додаток Б. Демонстрація роботи додатку.....	73

ВСТУП

На сьогоднішній день багато освітніх програм не викликають у школярів потрібного інтересу, а застарілі прийоми навчання не завжди мають змогу донести інформацію так, щоб людина запам'ятовувала її. Безкінечне читання книжок, та класичні тестування на папері не виправдовують свого використання.

Одним із варіантів вирішення цієї проблеми є створення ігрових квест-додатків, завдання яких буде спрямоване на покращення вивчення предметів та засвоєння знань. Відомо що квест-ігри значно краще допомагають людині засвоювати інформацію та набувати нових навичок, а створення додатків з використанням сценарію квест ігор стане вдалим способом підвищення інтересу в учнів [1]. Завдяки своїй структурі такий додаток може стати універсальним інструментом, який можна застосувати як для підведення підсумків з предмету, так і для введення в нові теми.

Метою проекту є створення квест-додатку, за сценарієм якого гравцю потрібно буде поетапно відповісти на ряд запитань на тему історії Воскресенської церкви (м. Лебедин) та її архітектурних особливостей.

Щоб отримати необхідний результат, потрібно реалізовувати проект згідно наступних задач:

- виконати аналіз предметної області;
- оглянути близькі за функціоналом додатки;
- обрати найбільш відповідний інструмент для роботи над проектом;
- структурувати етапи роботи;
- імпортувати 3D моделі;
- налаштувати анімації;
- реалізувати програмну частину додатку;
- провести тестування додатку;

Для успішної роботи над проектом потрібно сформулювати план робіт, а саме створити структуру робіт (діаграму WBS), розподілити відповідальних за кожне завдання в створеній структурі (діаграму OBS), організувати графік робіт згідно

встановлених термінів (діаграма Ганта), а також розробити функціональну модель та модель варіантів використання роботи додатку для розуміння того, які функції він буде виконувати.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд проблеми

Розвиток людини відбувається завдяки його активної діяльності. Основні принципи пізнання дітей шкільного віку реалізуються під час занять через прості методи і прийоми навчання. Під методом у даному випадку слід розглядати способи впливу або передачі знань. Прийоми реалізуються завдяки варіантам застосування певного методу. Тому методи і прийоми навчання в школах можуть бути практичними, наочними, словесними і ігровими.

Однією з найбільш цікавих і близьких вихованцям форм організації педагогічного процесу, є гра. У малій українській енциклопедії зазначено: «Гра - тип осмисленої непродуктивної діяльності, де мотив лежить не в її результаті, а в самому процесі». Через гру педагог допомагає школяреві знайти необхідну мотивацію, яка повинна бути заснована на індивідуальних потребах, забезпечувати свободу вибору, надавати можливість отримання необхідних знань і навичок [1].

У квестах присутній елемент змагання, а також ефект несподіванки (несподівана зустріч, таємничість, атмосфера, декорації). Вони сприяють розвитку аналітичних здібностей, розвивають фантазію і творчість, так як учасники можуть доповнювати квести по ходу їх проходження.

Роль ігрової діяльності і дитячої гри як такої в рішенні задач різнобічного розвитку і виховання дітей шкільного віку незаперечно визнана в педагогіці [2]. Гра для людини є найбільш привабливою, природною формою і засобом пізнання світу, своїх можливостей, самовияву і саморозвитку. Для педагога змістовна, що відповідає інтересам дітей, правильно організована гра є ефективним педагогічним засобом, що дозволяє комплексно вирішувати різноманітні освітні і розвиваючі завдання [1]. У грі відбувається всебічна підготовка дитини до життя, з чого випливає, що ігри пов'язані

з усіма видами діяльності людини і виступають як специфічно дитяча форма і пізнання, і праці, і спілкування, і мистецтва, і спорту. Своєрідність сучасного педагогічного процесу зумовлюється новими вимогами до дошкільної освіти [3].

Дуже поширена думка, що педагогічна взаємодія дитини і дорослих має бути орієнтована на забезпечення розвитку кожної дитини, збереження його унікальності і самобутності, створення можливостей розкриття здібностей, нахилів [4]. Максимально ефективно ці умови реалізуються під час проведення квест - ігор. В сучасних умовах і підходах до освіти квест - ігри є актуальними [5]. Спираючись на думки багатьох авторів можна сказати що квест-ігри є прогресивним способом взаємодії співпраці дорослих і дітей.

В даний час в шкільній освіті активно використовуються різноманітні інноваційні технології [6]. Необхідно відзначити, що квест - технологія актуальна в контексті сучасних вимог до освіти. Використання квестів дозволяє піти від традиційних форм навчання дітей і значно розширити рамки освітнього простору. Це інноваційна форма організації освітньої діяльності дітей в шкільній освіті, яка сприяє розвитку активної діяльнійшої позиції людини в ході рішення ігрових пошукових або розумових завдань. Таким чином, квест - гра - це діяльність людини в якій вона самостійно новий практичний досвід, і набуває знань.

Квест-ігри в загальноосвітньому процесі як поняття з'явилася відносно нещодавно [7]. Треба сказати, що велику роль в цьому зіграли не тільки психологи, а й поява пару десятиліть назад комп'ютерних ігор жанру quest. Тому завдяки сучасним технологіям є можливість розроблення додатку для реалізації будь яких сценаріїв. Розглянемо основні аспекти, пов'язані з розумінням цього процесу, щоб отримати розуміння, наприклад, що таке квест-технології в освіті та вихованні підростаючого покоління. Слід зазначити, що сучасний світ дуже насичений подіями і безліч різноманітних організацій готові запропонувати дитині цікаві форми проведення часу, що уже стали класичні форми організації педагогічного процесу менш привабливими і викликають менш яскраві емоції у вихованців [8]. А, як відомо, без інтересу немає результату або він досягається в меншому обсязі. У даній ситуації квест-додаток буде виступати помічником в організації освітнього процесу і

реалізації поставлених педагогічних завдань в незвичній для вихованців формі, тим самим підвищуючи їх інтерес і емоційну віддачу.

Квест - це один з основних жанрів ігор, що вимагають від учасника рішення розумових завдань для просування по сюжету. Сюжет може бути визначеним або ж давати безліч результатів, вибір яких залежить від дій гравця. Поняття «квест» в педагогічній науці визначається як спеціальним чином організований вид дослідницької діяльності, під час якої здійснюється пошук інформації за вказаними адресами, що включає і пошук цих адрес чи інших об'єктів, людей або завдань.

Актуальність використання квестів сьогодні усвідомлюється всіма. Нове покоління вимагає використання в освітньому процесі сучасних технологій і такою технологією повинен стати проект. Сьогоднішня картина показує, що сучасні люди краще засвоюють знання в процесі самостійного добування і систематизування нової інформації.

Використання квестів сприяє вихованню і розвитку якостей особистості, що відповідають вимогам інформаційного суспільства, розкриття здібностей і підтримки обдарованості дітей [9]. Поняття «квест» спочатку використовувалося як літературознавчий термін. Так називається спосіб побудови сюжету, де досягнення мети головним героєм відбувається через подолання численних труднощів. Як показав аналіз, квест – один з основних жанрів комп'ютерних ігор, що представляє собою інтерактивну історію з головним героєм, керованим гравцем.

Найважливішими елементами гри в жанрі квесту є власне розповідь і обстеження світу, а ключову роль в ігровому процесі відіграють рішення головоломок і завдань, що вимагають від гравця розумових зусиль. Квест - це ігри, в яких гравцеві необхідно шукати різні предмети, знаходити їм застосування, розмовляти з різними персонажами в грі, вирішувати головоломки і т.д.

Технологія квест-гри прекрасно вписується в освітній процес. Квест - заняття може бути призначене як для групової, так і для індивідуальної роботи. Навчання в процесі роботи над таким квест-проектом осягає реальні процеси, надає змогу проживати конкретні ситуації, долучатися до проникнення всередину явищ, конструювати нові процеси, об'єкти. З точки зору інформаційної діяльності при

роботі над квест-додатком розробнику потрібні навички пошуку, аналізу інформації, вміння зберігати, передавати, порівнювати і на основі порівняння синтезувати нову інформацію [7]. В ході реалізації квест-додатку можна природним чином здійснювати інтеграцію освітніх областей, комбінувати різні види діяльності і форми навчальної роботи.

При розробці і проведенні квестів важливо використовувати такі принципи:

1. Доступність завдань - не повинні бути надто складні.
2. Системність - завдання повинні бути логічно пов'язані один з одним.
3. Емоційна забарвленість завдань. Завдання повинні бути заховані за ігровими формами і прийомами.
4. Розумність за часом. Необхідно розрахувати час на виконання завдань таким чином, щоб гравець не втомився і зберіг інтерес.
5. Використання різних видів діяльності під час проходження квесту.
6. Наявність видимого кінцевого результату.

Але головне в такій організації ігрової діяльності полягає в тому, що, виконавши одне завдання, гравець може отримати підказку до виконання наступного, що є ефективним засобом підвищення мозкової активності і мотиваційної готовності до пізнання і дослідження [9].

Інтерактивні додатки стирають кордони між формальним і неформальним навчанням, сприяючи отриманню нових знань. За допомогою електронних пристроїв учні з легкістю знаходять додаткові матеріали, щоб глибше зрозуміти навчальний матеріал, про який розповідалося в класі або аудиторії. Наприклад, існує величезна кількість додатків для вивчення мов, які взаємодіють з учнями в інтерактивній формі.

В результаті аналізу можна зробити висновки, що квест-додатки є інноваційними технологіями в сьогоденному навчанні. Завдяки можливості налаштовувати квест-додаток під різні теми, його використання стане незамінним під час навчального процесу. На жаль, попри ці переваги, ще не створено квест-додатку, який дозволяв би ознайомитися та запам'ятати інформацію про такий історичний архітектурний об'єкт як Воскресенська церва міста Лебедин. Саме тому й було вирішено розробити даний проєкт.

1.2 Огляд аналогів

Для отримання чіткого представлення реалізації інтерактивного квест-додатку було оглянуто аналоги, які мають найбільш схожий функціонал та мету, або можуть запобігти виникненню помилок при розробці. Аналоги представлені наступними додатками:

- outer wilds;
- triviador;
- quest hunter.

1.1.1 Outer wilds

У Outer Wilds гравець управляє космонавтом-дослідником з видом від першої особи, досліджуючи тривимірний відкритий світ [10]. Гра дозволяє гравцеві вільно досліджувати навколишній світ, та шукати завдання. Outer Wilds не дає гравцеві будь-яких конкретних і явних завдань, крім підказок про місцезнаходження інших дослідників - неігрових персонажів, розкиданих по світу гри; в ході гри гравець повинен самостійно знайти відповіді на поставлені запитання. Наприклад, щоб перейти на наступний рівень в перший раз, персонаж повинен отримати коди запуску в місцевій обсерваторії, відповівши на ряд запитань.

Недоліком гри є час, який потрібен для проходження всіх завдань: світ гри кожен раз перезавантажується з початку через 30 хвилин, і гравцю потрібно починати з першої локації, при цьому виконані завдання зберігаються. Таке ускладнення гри є недоцільним і буде відштовхувати гравця й відволікати його від відповідей на запитання. Уже класичним є інтерфейс гри, де гравець повинен стежити за певними показниками – в цій грі вони представлені здоров'ям та запасом кисню космонавта – якщо той чи інший показник вичерпається, ігровий персонаж помре і гравцю потрібно буде починати спочатку або з останнього збереження ігрового процесу [1112].

Для виконання проекту корисно буде розглянути, як саме відбувається просування гравця по рівням при виконанні завдань, та розміщення підказок в локаціях.



Рисунок 1.1 – Діалогове вікно гри Outer wilds

1.1.2 Triviador

Кращою інтелектуальною грою називають безкоштовну онлайн гру Triviador [12]. Вона була запущена в 2002 році і на сьогоднішній день має велику армію прихильників. На даний момент гра доступна на 10 мовах.

Суть гри Triviador досить проста і захоплююча. Є ігрове поле, яке представляє собою карту певної країни, що складається з відповідних територій, і три гравці. У кожного гравця є спочатку своя територія. Під час гри задаються питання, і той, хто відповідає на них правильніше і швидше, починає захоплення сусідніх територій. Паралельно, відповідно їх вартості, гравцям нараховуються очки. Переможцем гри стане той, хто набере найбільше очок. Якщо Ви втрачаєте свою основну базу (територію, з якої починали гру), то гра для Вас закінчується в будь-якому випадку.

Головним недоліком гри є застарілі інструменти розробки [13], а саме Adobe Flash, до недоліків якого входять:

- для перегляду флеш-анімації в браузері необхідно встановити на ПК AdobeFlashplayer. Інакше сайти з flash-елементами не показуються в повному обсязі. Замість анімованої картинки відвідувач побачить порожнє поле. Щоб вирішити цю проблему в окремих випадках розробники сайтів встановлюють альтернативну картинку паралельно swf-файлу;
- контент, виконаний у флеш, потребує багато апаратних ресурсів і довго кешується, на його завантаження впливає швидкість інтернет-з'єднання;
- flash-технологія навантажує процесор і споживає багато оперативної пам'яті, а значить, плавне відтворення всіх елементів анімації гарантоване тільки для власників сучасних ПК;

Проте ще одним мінусом є недостатня інтерактивність гри, вона робить акцент лише на питання в стилі вікторини, відкидаючи на другий план різноманітність геймплею, що в свою чергу призводить до того гра швидко набридає.



Рисунок 1.2 – Реалізація запитань в грі Triviador

1.1.3 Quest hunter

Quest Hunter являє собою рольову гру з можливістю кооперативної гри в кількості до 4-х гравців [14]. Цей проект зосереджений на сюжетній складовій і різноманітності геймплею, що безсумнівно є сильною стороною цієї гри особливо при використанні бойової системи в стилі non target.

Згідно сюжету гри персонажу потрібно буде досліджувати ігровий світ, запам'ятовувати певні аспекти, та виконувати завдання, що являють себе відповідями на запитання. Весь виконаний видом зверху, а у кожного персонажа є три комірки для здібностей і чотири предметів. Мінусом сюжету гри можна назвати постійну темряву: вона додає атмосферності, але відволікає від головного, а саме відповіді на запитання.

При створенні гри було використано Unity 5, який підтримує мови C# і UnityScript. API. При використанні мови на зразок C#, програміст не зобов'язаний використовувати покажчики (pointers), компіляція відбувається швидко, проте у Unity немає системи візуального скриптування, і щоб використовувати щось подібне, потрібно купувати сторонні додатки. Також недоліком Unity для проекту може стати неможливість редагування анімацій [1516].

Та взагалі, Quest Hunter є відмінною грою для дітей та їх батьків, адже її можна проходити разом, а наявність гумору, приємної картинки і нехитрого ігрового процесу який не лише розважає, а ще й навчає, додають шарму.



Рисунок 1.3 – Інтерфейс гри Quest hunter

1.3 Результат дослідження аналогів

Після проведення огляду додатків, функціонал яких частково надає можливість виконувати завдання, поставленні перед проєктом, було вирішено порівняти їх за декількома аспектами, щоб мати змогу уникнути тих мінусів, що знаходяться в переглянутих додатках.

Таблиця 1.1 – Порівняння додатків

Критерії аналізу	Outer wilds	Triviador	Quest hunter
Графічна частина	+	-	+
Інтерактивність	+	-	+
Складність	+	+/-	+/-
Інтелектуальність	+/-	+	-
Універсальність	-	+	-

З аналізу аналогів можна побачити, на які аспекти слід звернути увагу при роботі над власним проєктом, щоб він задовольняв визначені в таблиці критерії, виконуючи свої функції. Але слід звернути увагу, що квест-додаток носить в собі в першу чергу освітній характер.

Освітні ресурси і інформацію уже зараз можна зберігати в хмарних сховищах, а не на жорсткому диску, тому учні можуть працювати з великим об'ємом матеріалу, що знаходиться в одному місці.

Сучасне ПО дозволяє синхронізувати дані на кількох пристроях, тому учні можуть відновити роботу вдома на комп'ютері з того місця, де вона була припинена в аудиторії, і навпаки.

Завдяки додатку буде можливість перенести велику кількість цікавої історичної інформації в електронний формат.

Він надасть змогу підтримувати ситуаційне навчання, порівняно з традиційним процесом навчання, дозволить перемістити учнів в середовище, що максимально полегшує розуміння предмета, за допомогою різноманітних способів донесення інформації всередині додатку [16].

2 ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

2.1 Мета та задачі

Головною метою проєкту є створити інтерактивний квест додаток для вивчення історії Воскресенської церкви міста Лебедин.

Функціонал додатку матиме змогу зацікавити як учня, так і дорослу людину, тому при створенні сценарію було вирішено використовувати історичні відомості щодо Воскресенської церкви міста Лебедин.

Такий додаток надає інформацію про історичний архітектурний об'єкт Воскресенську церкву, її композиції, конструкцію і значення тощо.

Додаток забезпечуватиме зв'язок між формальним і неформальним навчанням.

Для реалізації проєкту потрібно вирішити такі задачі:

- проаналізувати предметну область використання квест-технології в освітньому процесі, визначити актуальність створення квест-додатку;
- провести аналіз близьких за функціоналом додатків, визначити їх недоліки та переваги, обрати необхідні інструменти і технології для реалізації всіх аспектів проєкту;
- розробити структуру та логіку роботи квест-додатку;
- підготувати та забезпечити коректність імпорту 3D моделей в ігровий рушій, налаштувати та перевірити правильність відображення анімацій в додатку;
- реалізувати програмні зв'язки між об'єктами проєкту;
- провести тестування та усунути можливі помилки.

Даний додаток повинен реалізувати наступні можливості:

1. Квест-додаток повинен запускатись на ПК з ОС Windows.
2. Гравець повинен штрафуватись за неправильні відповіді.
3. Гравець може обирати власний порядок відповідей на запитання.

4. В додатку повинна реалізовуватись можливість отримати підказки на важкі запитання.
5. Правильна відповідь на запитання обирається за чотирьох варіантів за допомогою клавіш 1,2,3,4 на клавіатурі.
6. Ігрова мапа повинна бути розділена на три локації, доступ до яких відкривається протягом проходження додатку.
7. Локації в додатку повинні бути оформлена згідно теми роботи.
8. В результаті проходження квест-гри гравець має отримати бонус.

2.2 Вибір інструментів реалізації

Перед початком роботи над додатком потрібно визначити, які інструменти краще підійдуть для досягнення мети. Для цього розглядалося два ігрових рушія, а саме Unity і Unreal Engine 4, кожний рушій має свої переваги та мінуси, на які було звернено увагу при виборі [17].

Для початку потрібно було визначитися з типом квест-додатку, якого саме вигляду він буде. Варіант з 2D платформером є не таким привабливим, на відміну від 3D квесту від першої особи [18]. Тому був сформований список вимог, яким повинен задовольняти ігровий рушій:

- робота рушія з фізикою;
- зручність розробки для ПК;
- складність реалізації базових ігрових механік;
- робота з текстурами та освітленням;

Відповідно рушій, що найбільш відповідає цим пунктам, дозволить реалізувати поставлену мету роботи. Звичайно обидва рушія можуть впоратися з будь-яким з цих завдань, але залежно від того, що саме потрібно зробити, один рушій може бути набагато зручніше і оптимальніше [19].

При розробці мобільних додатків Unity буде ідеальним рішенням. Це підтверджується домінуванням Unity серед розробників мобільних ігор, а також великою кількістю плагінів для використання нативних можливостей мобільних платформ [20].

Якщо ціллю є розробка 2d гри, Unity теж буде прекрасним вибором, тому що саме у нього є прекрасні можливості для створення 2d ігор. Хоча Unreal Engine 4 останнім часом намагається заманити розробників мобільних додатків, обіцяючи не менші, а навіть більші можливості для 2d ігор [21 **Ошибка! Источник ссылки не найден.**].

При розробці 3d ігор, Unity також є дуже потужним інструментом для розробки 3D ігор. Хоча графічно він далеко не на тому рівні, що Unreal Engine 4, і при необхідності створювати next-gen графіку кращим рішенням буде Unreal Engine 4 [22].

При розгляді 3D проектів слід звернути увагу саме на Unreal Engine. На даний момент остання версія його це 4.25, випущена в травні 2020 року .

Unreal Engine 4 (UE4) розроблявся з 2003 року, про це свідчить напружена робота Epic Games над рушієм. Цей рушій являє собою чудову основу для створення фізики та графіки нового покоління [23]. Однією з переваг рушія є його якісне виконання динамічного освітлення, яке допомагає по-справжньому зануритись віртуальний світ.

Для роботи над 3D він справді став галузевим стандартом, вигравши безліч нагород від ігрових видань. Unreal не виявляє ознак стагнації, він використовується у найновіших проектах [24].

Компанія Epic Games, що розробляла рушій, багатьох здивувала, показавши, що програмне забезпечення рушія набагато простіше, ніж люди могли уявити. В Unreal Engine 4 можна створювати програмні зв'язки без прямого кодування, за допомогою нодів, що є сильною стороною при розробці.

Інструмент візуального програмування за допомогою нодів blueprint забезпечує чітке представлення програмної взаємодії між об'єктами гри під час сеансів PIE та SIE. Елементи керування нодами розташовуються на панелі інструментів і стають

активними, коли гра запущена. Під час цього також з'являються різні режими керування налагодженням залежно від типу blueprint, і поточного стану сеансу [24]. Деякі інструменти стають доступні до використання лише тоді, коли це доречно, наприклад, коли взаємодія між об'єктами завершилася.

Blueprint Watch Window призначена для прискорення налагодження, надаючи доступ до змінних та вузлів, які потрібно переглянути. Дані перегляду з кожного проекту, який відкривається в Редакторі, і який є частиною поточного списку, об'єднуються в єдиний список, що заповнюється поточними даними, коли виконання Blueprint буде призупинено. Використовуючи цей список, редактор нодів надає можливість перевіряти змінні та виходи функцій, а також легко переходити між ними. Масиви, карти та інші структури даних можна розширити, роблячи детальний аналіз будь-якої інформації, яку вони містять, швидким та зручним[25].

Оглянувши всі переваги та недоліки, було прийнято рішення використовувати Unreal Engine 4, оскільки в проекті буде робитись акцент на 3D графіку при роботі з якою Unreal Engine має більше переваг над Unity.

3 ПРОЄКТУВАННЯ РОЗРОБКИ КВЕСТ ДОДАТКУ

3.1 Етапи проєктування

Реалізація ідеї квесту у вигляді додатку є актуальним способом розробити інструмент, що буде корисним для освітніх цілей. Для цього є багато різноманітних способів та інструментів для розробки.

Поетапна робота над проєктом дозволить вчасно та без критичних помилок організувати його реалізацію. Схематично етапи роботи над додатком наведено на рисунку 3.1.

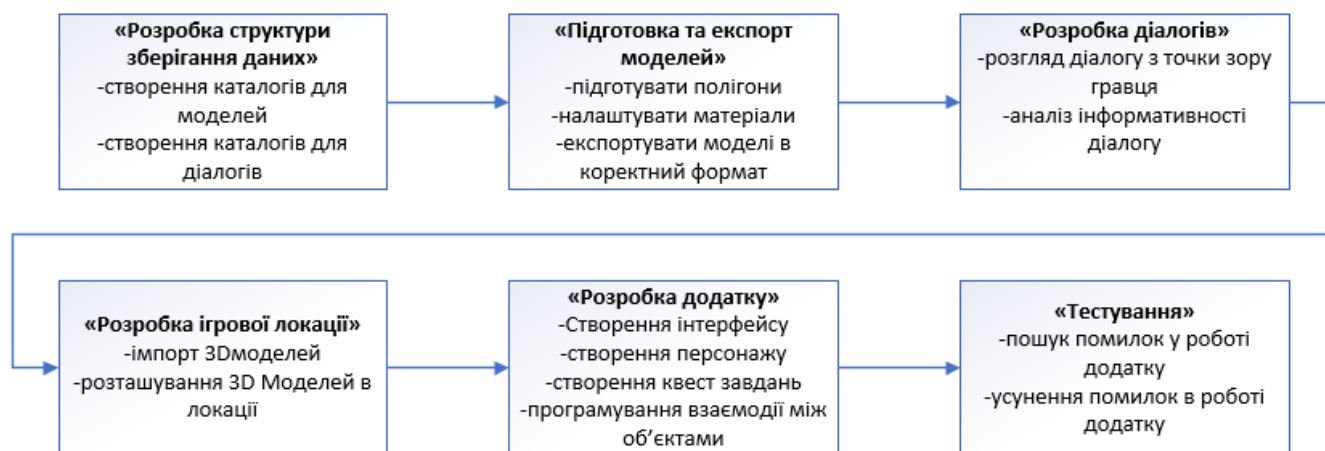


Рисунок 3.1 – Схема етапів проєктування

Одним з важливих етапів розробки квест додатку є формування коректної структури зберігання даних. У грі квесту потрібні каталоги, що зберігають персонажів, діалоги і т.д. Важливо сформуванати каталоги і в рамках рівня, а також шари, що містять всі види об'єктів, які потрібні нам в ігровому світі [20]. Оскільки на етапі проєктування вже можна бачити чітку картину додатку, підготувати все це не складатиме труднощів.

Наступною задачею при роботі з проектом є підготовка та експорт 3D моделей, для чого потрібно виконати певні дії:

1. Перевірити, чи у всіх моделей опорні точки Pivot Point знаходяться в потрібному місці.
2. Триангулювати всі моделі, так як графічна карта працює з трикутними полігонами.
3. Створити відповідні UV розгортки текстур для кожної моделі.
4. Створити карту нормалей для високо- та низько-полігональних моделей.
5. Створити колізії для об'єктів, що цього потребують, оскільки колізія дуже важлива для оптимізації прорахунку фізики в процесі гри.
6. Експортувати всі необхідні об'єкти в формат FBX з відповідними налаштуваннями.

Кожна гра з сюжетом не може обійтись без діалогів, тому в нашому проекті вони також будуть присутні. Іноді вони можуть бути дуже узагальненими, іноді більш конкретними – на цьому етапі потрібно детально працювати над текстом. Після визначення персонажів потрібно додати невеликий опис і розпочинати роботу над самим текстом діалогу.

В наступному етапі потрібно імпортувати попередньо підготовлені моделі ігрової локації у сцену, розмістити їх відповідно задуму квесту. У рідких випадках потрібно буде взаємодіяти з камерою, анімацією, мімікою. Звичайно вираховується кілька стандартних ракурсів, а в подальшому моделі редагуються попередньо зробленими налаштуваннями [26]**Ошибка! Источник ссылки не найден..**

Коли готова базова структура, в грі присутні всі персонажі і об'єкти, можна приступати до безпосередньої роботи над рівнем. Щоб створити цікавий світ, основні завдання гравець повинен знаходити в різних частинах локації. На цьому етапі розробляється інтерфейс додатку, створюються квест-завдання, визначається та програмується інтерактивна взаємодія між об'єктами додатку.

Коли готовий прообраз квесту, починається його ретельне тестування. В кінцевому підсумку, коли вся послідовність подій проходить вірно, все працює гладко і завдання квесту досить цікаве, роботу над ним можна вважати завершеною. Але

якщо на цьому етапі виникають проблеми, очевидно, необхідно ще попрацювати. Зрештою завданню виставляють оцінку по наступним пунктам:

- як завдання вписується в гру;
- чи цікаво проходити квест;
- чи вірно він збалансований.

Зазвичай трапляється кілька помилок, іноді ми додаємо новий сюжетний поворот або ідею, яка працює краще. В такому випадку квест ще якийсь час переробляється, поки не буде готовий до остаточного поданням.

Як тільки все працює правильно, діалоги передаються на доопрацювання. Тут в черговий раз щось може змінитися. Іноді, поки ми працюємо над сюжетом, з'являється зовсім нова ідея, і ми може впустити її в хід. Звичайно, іноді зміни доводиться відкидати назад; буває так, що потрібно внести зміни в уже зроблений діалог і завдання змінюється відповідно.

Коли все готово, проект виходить на новий рівень, він ретельно перевіряється тестуванням з паралельним внесенням виправлень. Він знову отримує оцінку. Кожне додане в гру завдання тестується, оцінюється і допрацьовується. До фінальної реалізації повинні доживати лише найкращі ідеї.

3.2 Структурно-функціональний аналіз проекту в нотації IDEF0

Найкращим способом графічно зобразити систему проекту є методологія IDEF0, оскільки вона є простою і достатньо інформативною мовою опису бізнес-процесів. За допомогою цього стандарту можливо передавати інформацію між усіма учасниками проекту, або демонструвати етапи розробки фахівцям.

Стандарт IDEF0 зроблений максимально ретельно та задовольняє всі потреби при проектуванні процесів в проекті. Для роботи з ним існує безліч інструментів, наприклад, VISIO, BPWIN, ERWIN, Bussines studio, для реалізації даного проекту було обрано програмний продукт компанії Computer Associates – Bpwin [27].

За допомогою цієї методології було проведено структурно-функціональний аналіз використання додатку, представлений контекстною діаграмою (рис. 3.2).

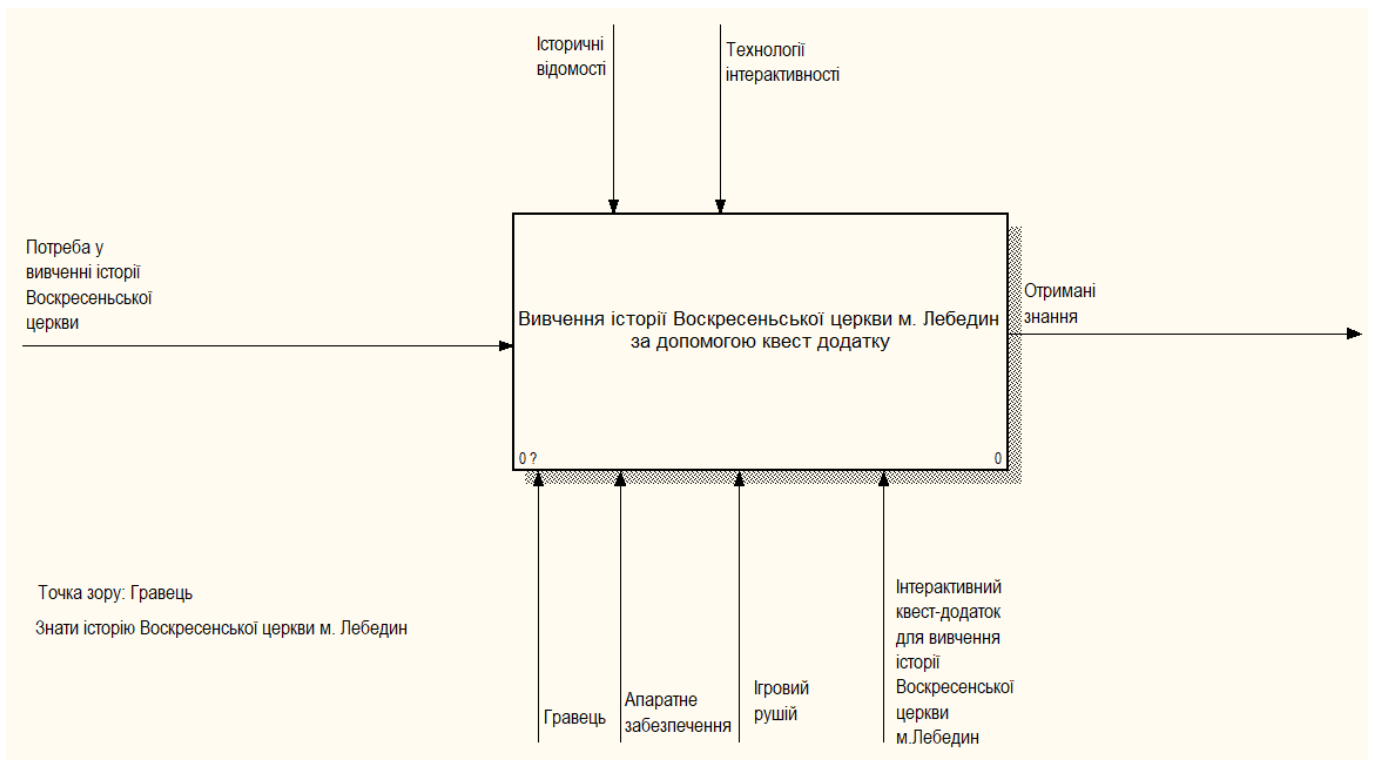


Рисунок 3.2 – Контекстна діаграма

Згідно методології, на контекстній діаграмі присутня вхідна інформація, а саме «Потреба у вивченні історії Воскресеньської церкви», відповідно на виході результатом процесу будуть «Отримані знання».

Керуватись процес використання додатку буде «Історичними відомостями» та «Технологіями інтерактивності», а механізмами виконання виступатиме «Гравець», «Апаратне забезпечення», «Ігровий рушій» та сам «Інтерактивний квест додаток для вивчення історії Воскресеньської церкви (м.Лебедин)».

На першому рівні декомпозиції процесу діаграма розбивається на наступні блоки, представлені на рисунку 3.3:

- ознайомитись з ігровою локацією;
- виконати квест завдання;
- отримати винагороду.

Вхідною стрілкою до блоку А1 «Ознайомлення з ігровою локацією» є потреба у вивченні історії Воскресенської церкви, механізмами до цього блоку будуть виступати гравець, ігровий рушій, відповідно сам додаток та апаратне забезпечення, а керуючим елементом технології інтерактивності.

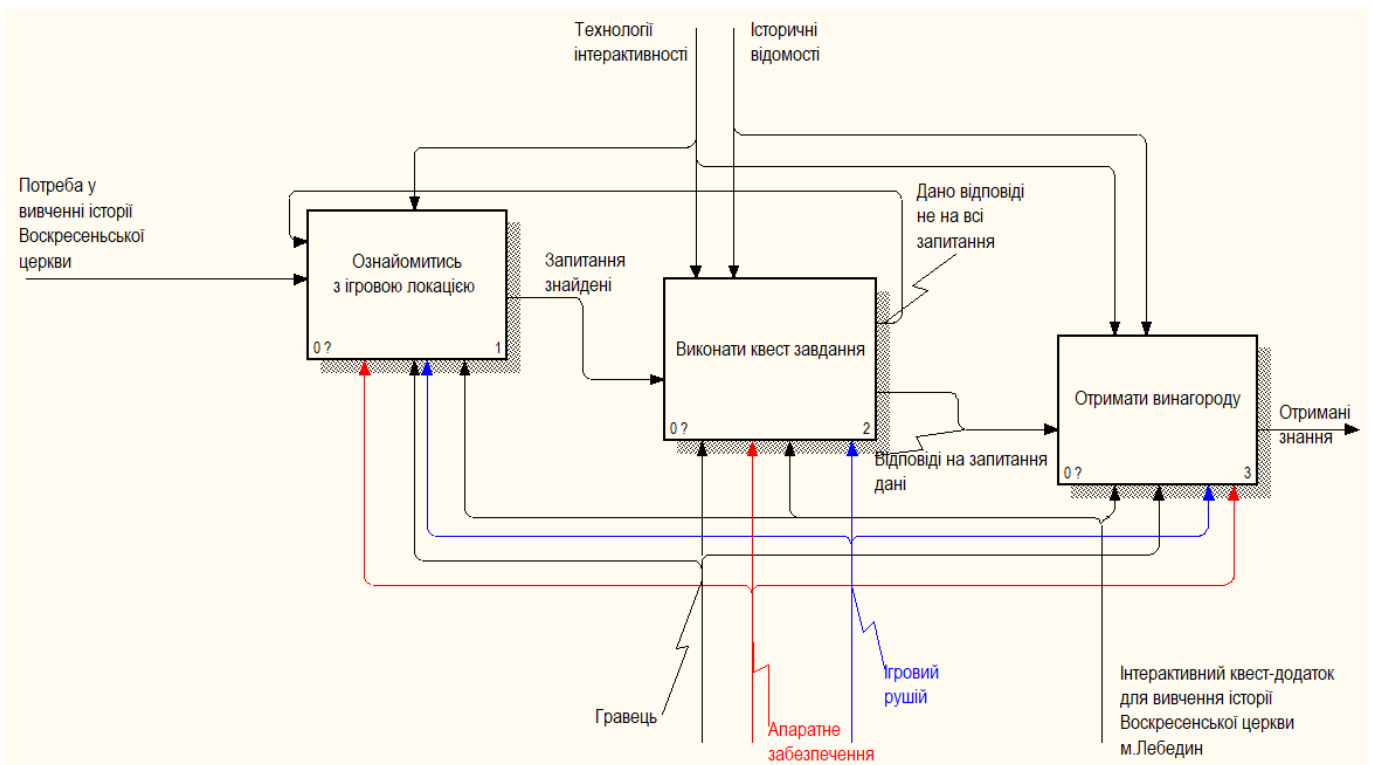


Рисунок 3.3 – Діаграма декомпозиції 1 рівня

Після знаходження запитань в квест додатку потрібно буде виконати їх, що і показує блок А2 «Виконати квест завдання». Для нього існує стрілка зворотного зв'язку, яка показує, що дано відповіді не на всі запитання. Механізмами для нього також будуть виступати гравець, апаратне забезпечення, ігровий квест-додаток та рушій, проте до керуючих елементів додаються ще історичні відомості щодо Воскресенської церкви.

За блоком виконання завдання у випадку надання вірних відповідей слідує блок А3 «Отримати винагороду». Блок «Отримати винагороду» показує кінцевий процес використання додатку, вихідною стрілкою якого є отримані знання.

В діаграмі існує 3 рівня декомпозиції. На рисунку 3.4 наведено другий рівень декомпозиції блоку А1 «Ознайомитись з ігровою» локацією, де показано основні процеси, які буде виконувати гравець перед відповідями на запитання. Другий рівень декомпозиції цього блоку представлений процесом ознайомлення з відкритими для гравця зонами та пошуком місцезнаходження квест завдань. Блок «Ознайомитись з відкритою ігровою зоною» на виході представлений стрілкою, яка позначає, що ігрову локацію обстежено.

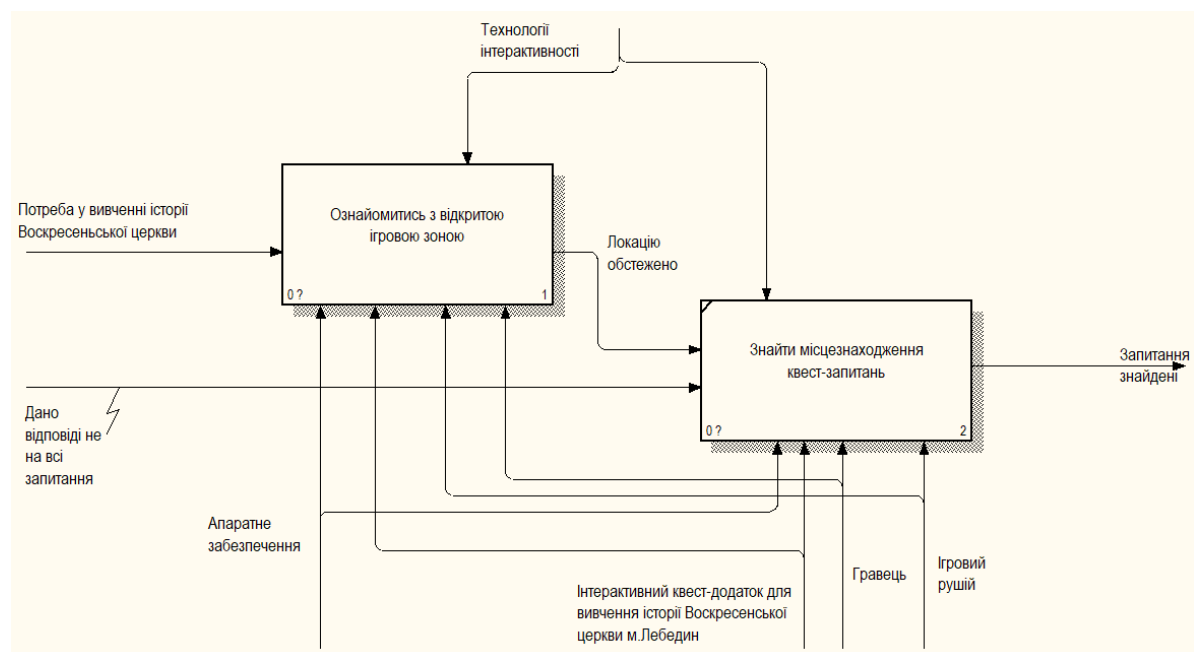


Рисунок 3.4 – Діаграма декомпозиції блоку «Ознайомитись з ігровою локацією»

Блок «Ознайомитись з відкритою ігровою зоною» також має третій рівень декомпозиції (рис. 3.5), а саме блоки «Оглянути доступний периметр» та «Знайти перехід до наступної локації».

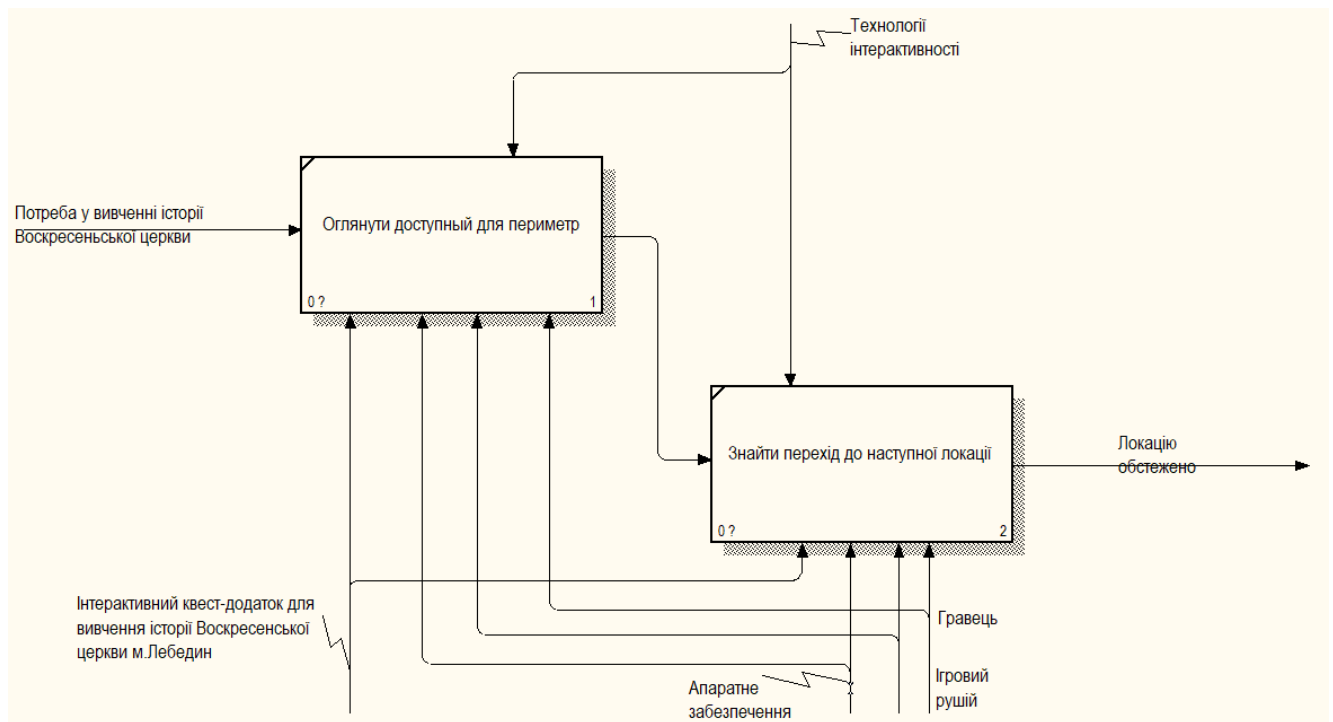


Рисунок 3.5 – Діаграма декомпозиції блоку «Ознайомитись з відкритою ігровою зоною»

Діаграму декомпозиції процесу «Виконати квест завдання» показано на рисунку 3.6. Вона представлена трьома блоками «Прочитати запитання», «Дати відповідь згідно наданим варіантам», «Перейти до іншої частини локації». Механізмами для всіх трьох блоків виступають «Гравець», «Апаратне забезпечення», «Інтерактивний квест-додаток для вивчення історії Воскресеньської церкви (м. Лебедин)» та «Ігровий рушій», а засобами керування «Технології інтерактивності» та «Історичні відомості». Також на цій діаграмі приведено приклад зворотного зв'язку, який представлений стрілками «Дано невірну відповідь на запитання» та «Дано відповіді не на всі запитання».

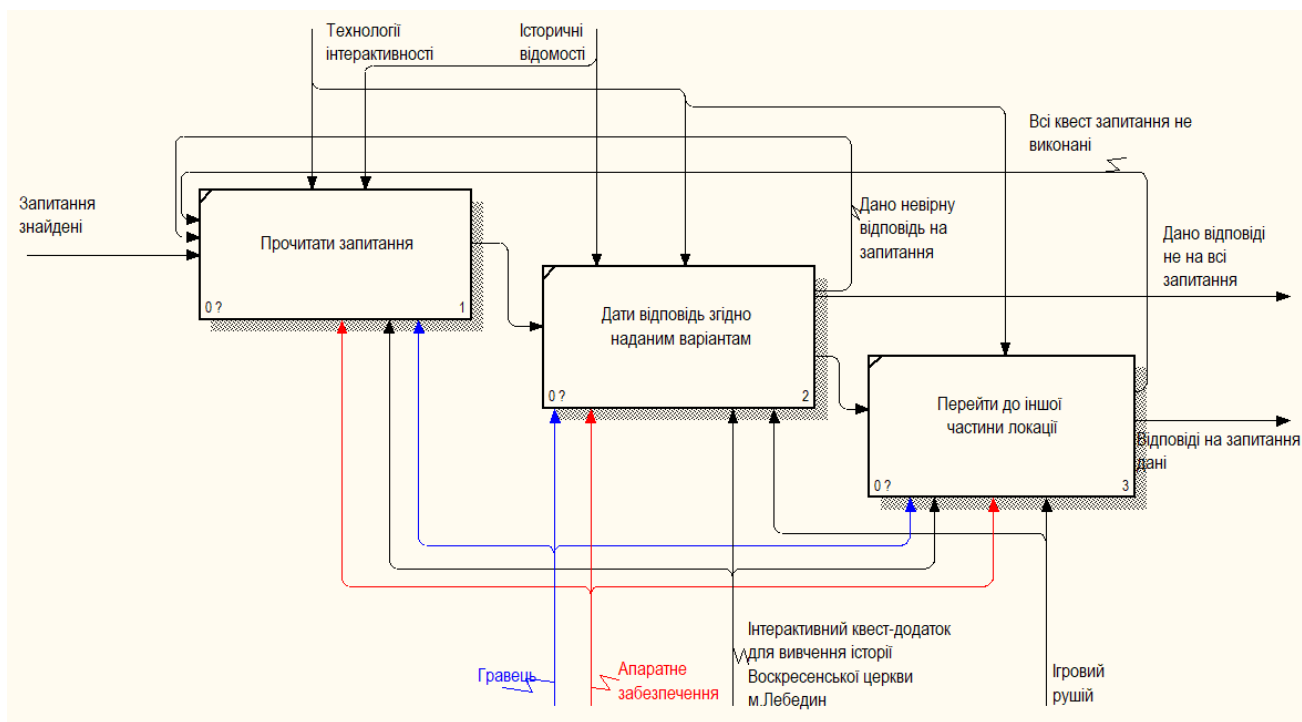


Рисунок 2.6 – Діаграма декомпозиції блоку «Виконати квест завдання»

Останнім блоком для декомпозиції є «Отримати винагороду» (рис. 3.7). Цей блок відображає процес завершення квесту й керується історичними відомостями, технологіями інтерактивності, на виході з блоку ми отримуємо знання щодо історії Воскресенської церкви (м. Лебедин).

Таким чином, в результаті проведеного структурно-функціонального аналізу на основі методології IDEF0 визначені основні процеси, що проходять у додатку, показано механізми управління та керування цими процесами, розписано вхідні дані до кожного блоку декомпозиції та показані зворотні зв'язки між процесами, а вихідні стрілки чітко показують отримувані результати.

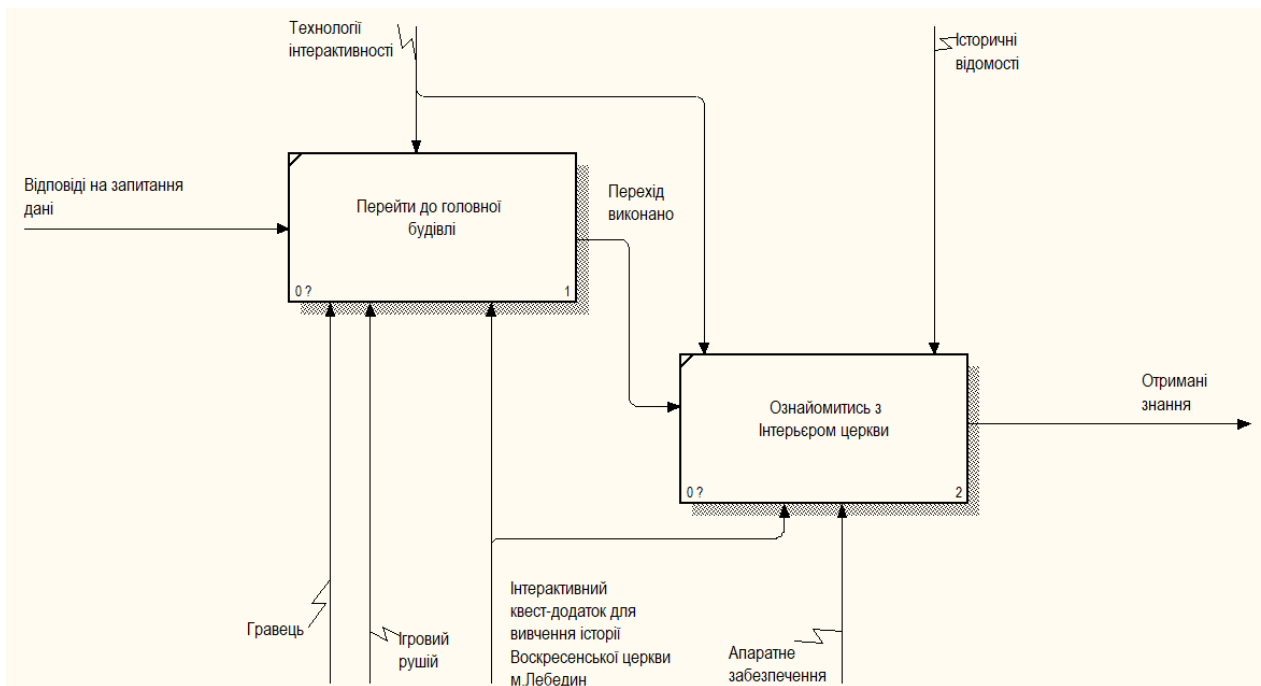


Рисунок 3.7 – Діаграма декомпозиції блоку «Отримати винагороду»

3.3 Створення діаграми варіантів використання

Діаграмами варіантів використання називають діаграмами поведінки. Вони використовуються для опису певних дій, котрі деякі системи (або підсистеми) можуть виконувати у співпраці з зовнішніми користувачами системи, яких називають акторами. Кожен випадок використання повинен дати певний спостережуваний та цінний результат для суб'єктів або інших зацікавлених сторін системи [29].

Перед графічним зображенням діаграми потрібно визначити акторів та варіанти використання. В діаграмі був визначений актор, який безпосередньо буде взаємодіяти з додатком. Цим актором є гравець, він через «Запуск головного меню» та «Виконання завдань» буде взаємодіяти з іншими варіантами використання.

У моделі реалізовані зв'язки *include* та *extends*. Зв'язок *include* показує, що саме використовує базовий варіант для виконання функції, а зв'язок *extend* вказує на можливість використання, відмінну від стандартного варіанту [30]. Прецеденти (варіанти використання) представлені наступними блоками:

- виконання завдань;
- запуск головного меню;
- вихід;
- початок гри;
- отримання винагороди;
- відкриття частини локації;
- перевірка відповідей на запитання;

Розроблена діаграма представлена на рисунку 3.8.

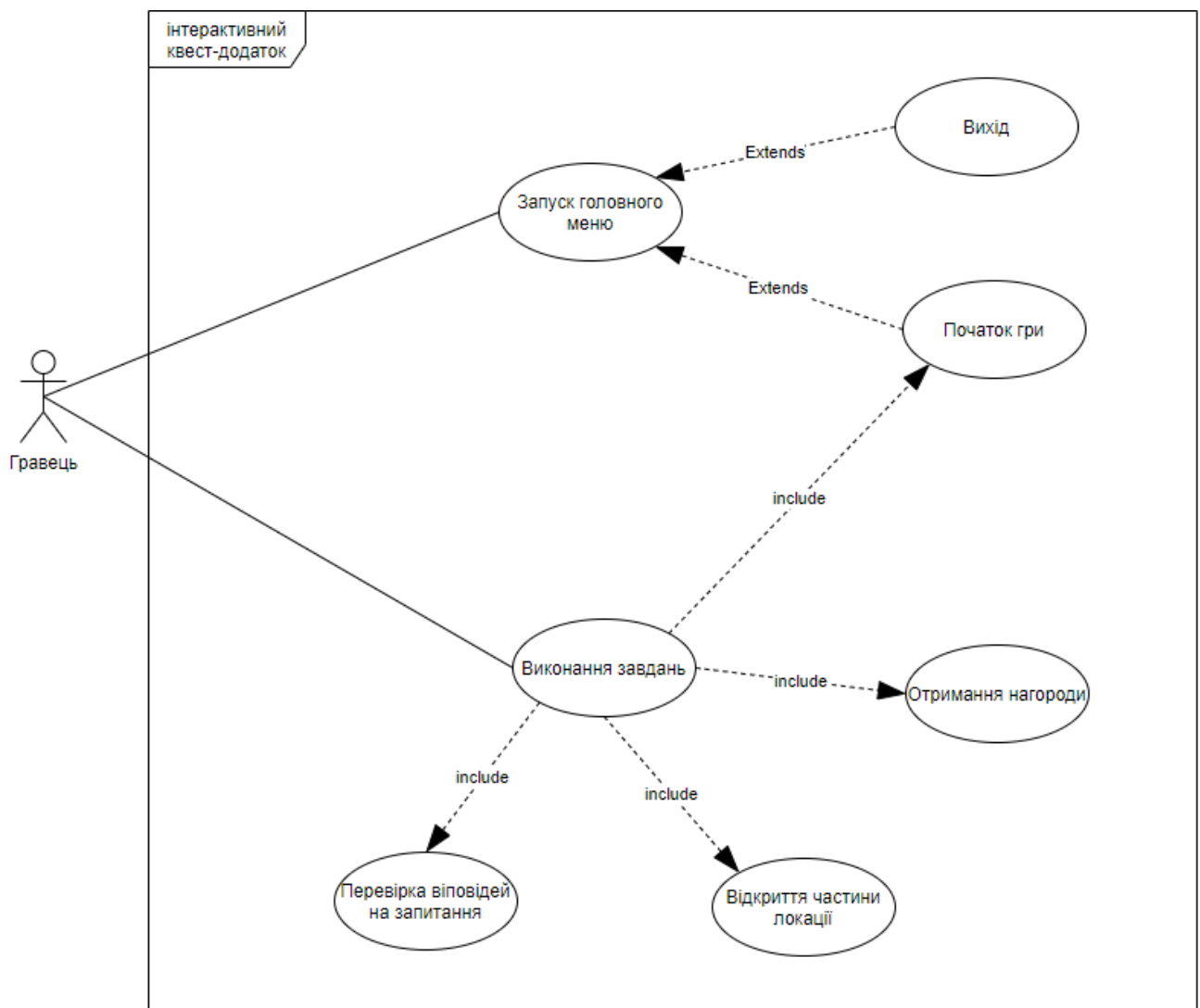


Рисунок 3.8 – Діаграма варіантів використання

4 РЕАЛІЗАЦІЯ ІНТЕРАКТИВНОГО ДОДАТКУ

4.1 Підготовка 3D моделей до імпорту

Першим етапом у реалізації додатку була підготовка 3D моделей до експорту. Потрібно оптимізувати частини моделей, які завадили б правильному відображенню текстур в додатку. Для експорту обрано формат FBX, оскільки він забезпечує підтримку всіх конструкцій сцени, включаючи анімацію, системи кісток, об'єкти морфинга і анімації. У ньому також передбачена можливість вставки текстур або їх перетворення в зображення формату TIF.

Крім того, за допомогою цього формату можна експортувати і імпортувати вибрані одиниці виміру і координати. У форматі FBX можна фільтрувати спеціальні об'єкти: діалогові вікна з параметрами імпорту та експорту даних в файлах формату FBX. Щоб визначити об'єкти сцени, які потребують оптимізації, 3D модель було переведено у режим каркасного представлення.

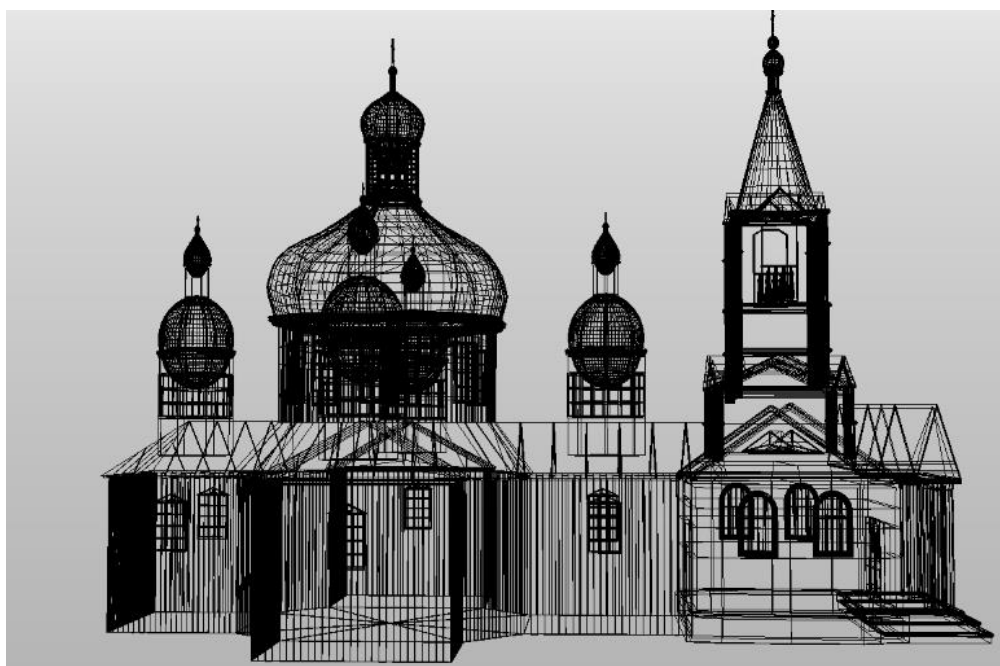


Рисунок 4.1 – Каркасный вигляд моделі

Після оптимізації моделі загальна кількість полігонів стала 144646, а вершин 88912, що задовольняє потреби для експорту.

У діалоговому вікні експорту налаштовуємо необхідні параметри для коректного відображення моделі в ігровому рушії (рис.4.2). При експорті моделі головним було збереження геометрії, тому в підменю Geometry вибрано параметри TurboSmooth, Convert Deforming Dummies to Bones, Triangulate, Preserve edge orientation. Після чого модель було експортовано у файл формату FBX.

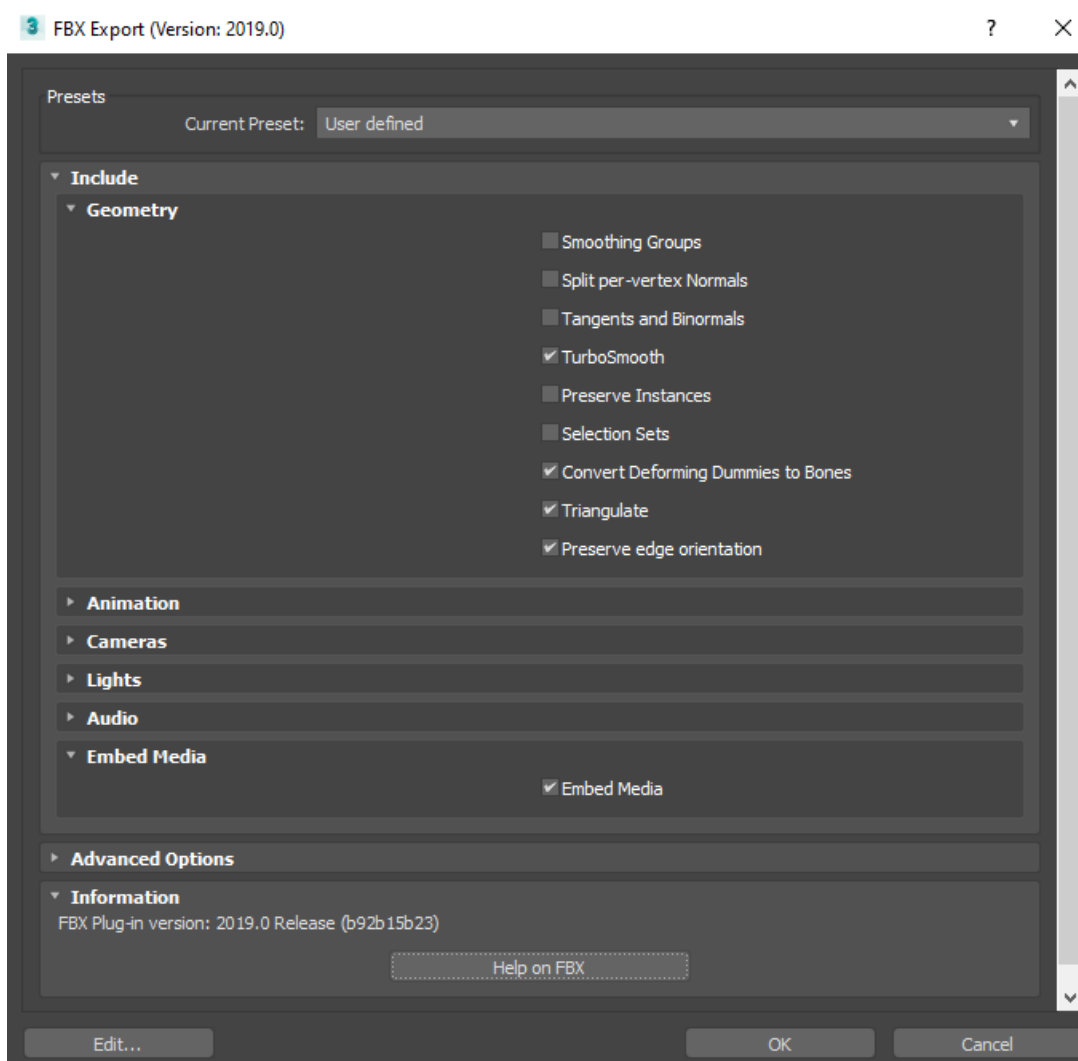


Рисунок 4.2 – Налаштування експорту моделі

Оскільки експорт відбувався в програмному забезпеченні 3Ds max 2019, тому плагін FBX був використаний відповідної версії – 2019.0.

4.2 Розробка ігрової локації у рушії

Для створення проекту використано шаблон проекту з видом від третьої особи. Такий режим камери надасть гравцю оптимальну можливість взаємодії з ігровою локацією. В налаштуваннях відповідно було вибрано Desktope, оскільки додаток розробляється під ПК, також в якості додаткових налаштувань було вибрано Maximum Quality, та додано стартовий контент, який спрощує наповнення сцени.

Unreal Engine 4 надає можливість програмувати процедури ігрового сценарію за допомогою візуально представлених блоків – нодів, що надає чітке представлення взаємодії між кожними об'єктом сцени.

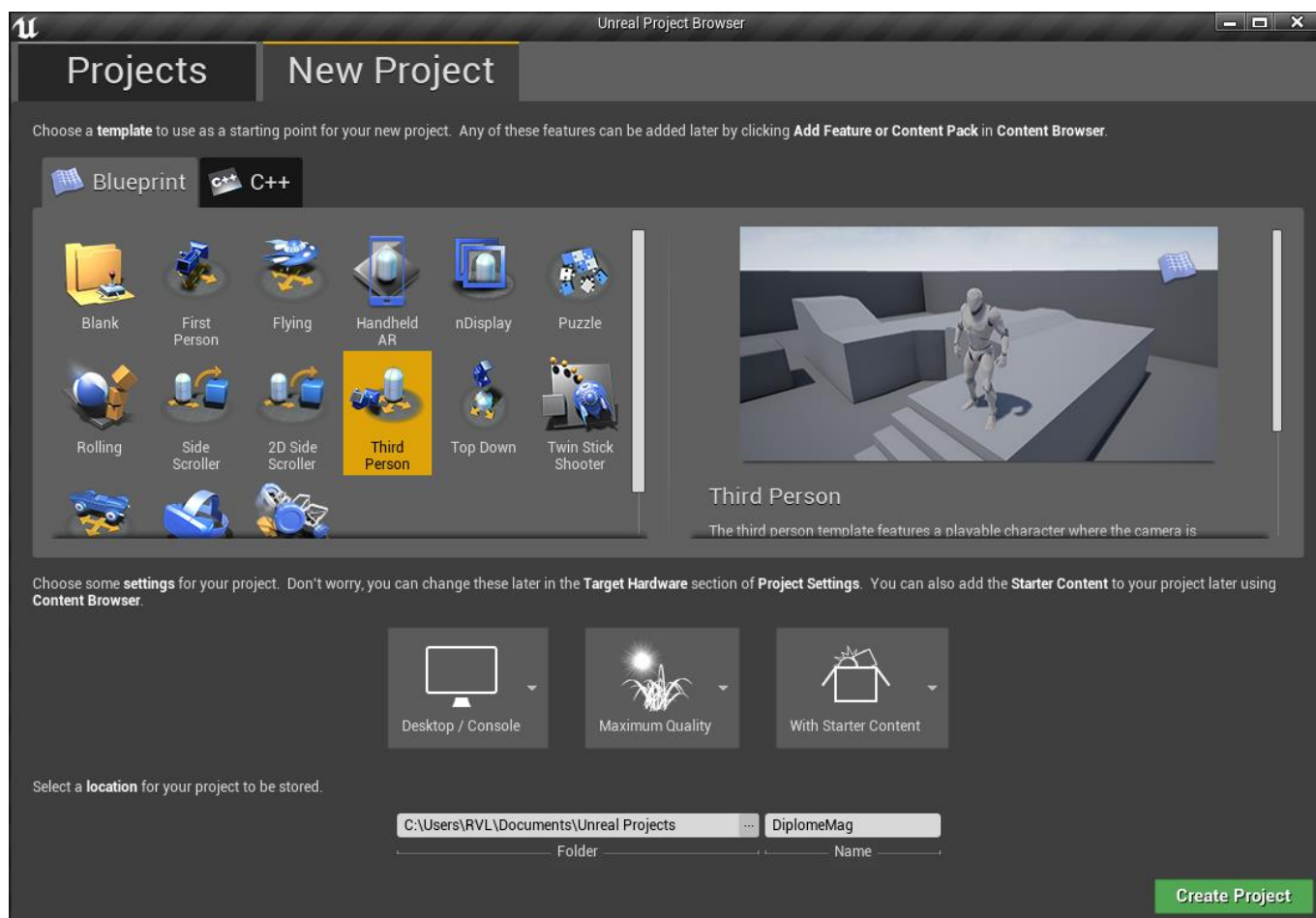


Рисунок 4.3 – Вікно створення проекту

Після того як проект створився, необхідно використати інструменти Unreal Engine (UE4) для налаштування локації, а саме створення ландшафту. Для цього в UE4 в меню обрано пункт ландшафт Landscape, у вкладці Modes.



Рисунок 4.4 – Вікно створення ландшафту

Оскільки накладення текстур для ландшафту буде відбуватись вже після його створення, параметр Material нам не потрібен. Дизайн локації не потребує нахилу відносно горизонту тому поля Location Rotation, будуть відповідати значенню нуль, а поле Scale буде дорівнювати 100, оскільки цей параметр регулює лише масштаб, а не фактичний розмір ландшафту. Section Size встановлюється розміром 31 на 31 квадрат, що задовольняє потреби проекту. Інші параметри залишаємо за замовчуванням.

Після створення ландшафту у вікні перегляду сцени буде відображено плоский прямокутник без текстур. Завдяки відображенню прямокутників на поверхні, зрозуміло де саме знаходиться кожна секція ландшафту (рис.4.5), що в свою чергу значно полегшує створення нерівностей.

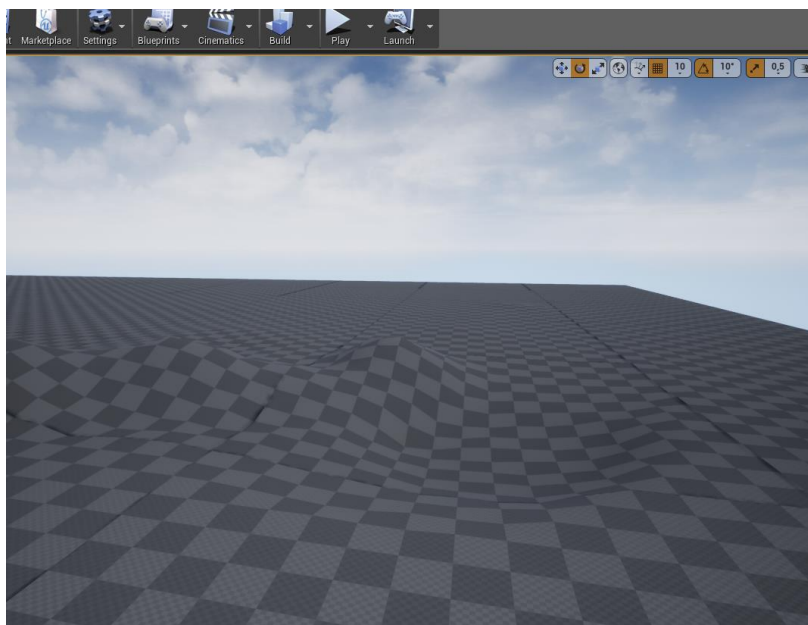


Рисунок 4.5 – Створений ландшафт

Для надання ландшафту більш реалістичної форми використовувались інструменти Sculpt, Smooth, Erosion. Зокрема Sculpt був основним інструментом при створенні висот в локації, та розділенню її на частини. Такий метод редагування ландшафту є доволі зручним та швидким, а результат роботи гарантує чітке розуміння гравцю, як саме йому потрібно переміщатись локацією.

В результаті роботи з цими інструментами сцену було поділено на коридорні локації, обмежені пагорбами та горами (рис 4.6).

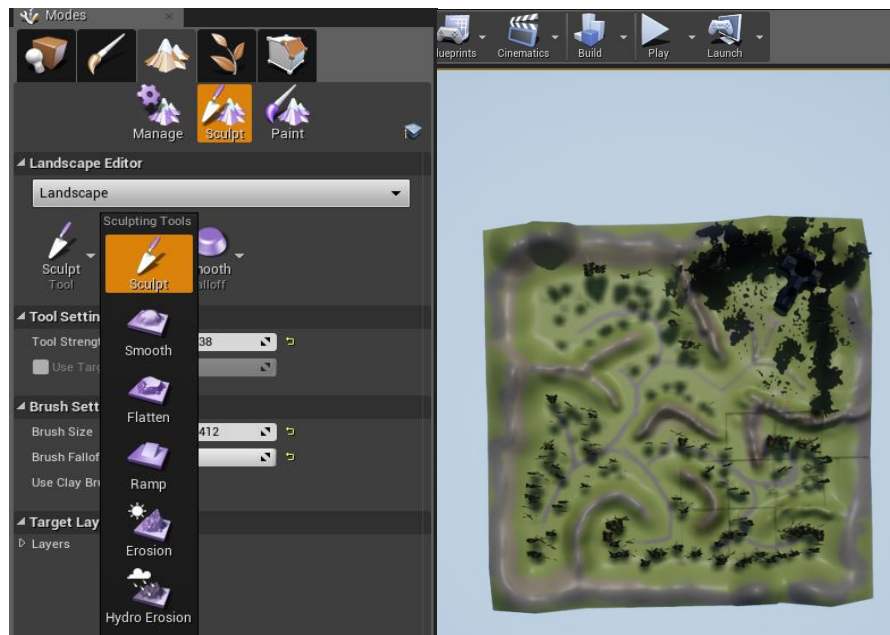


Рисунок 4.6 – Результат роботи з ландшафтом

Коли роботу над висотами завершено, потрібно створити та призначити матеріал на готову поверхню. Для цього в UE4 використовується інструмент Paint, для роботи якого потрібно створити спеціальний матеріал, що містить шари.

Шар - це частина призначеного ландшафту матеріалу, яку необхідно намалювати на ландшафті, щоб змінити його вигляд. Ці шари визначаються, як текстура, що застосовується до актору.

Ландшафт також може використовувати кілька шарів з різними текстурами, масштабуванням, обертанням та панорамним переміщенням, щоб створити остаточну текстуровану місцевість.

Шари, визначені в альбомному матеріалі, автоматично заповнюють список цільових шарів у інструменті Paint. Під час створення такого матеріалу використовується нода Layer Blend, в якій зазначаються параметри для підключення текстур (рис.4.7).

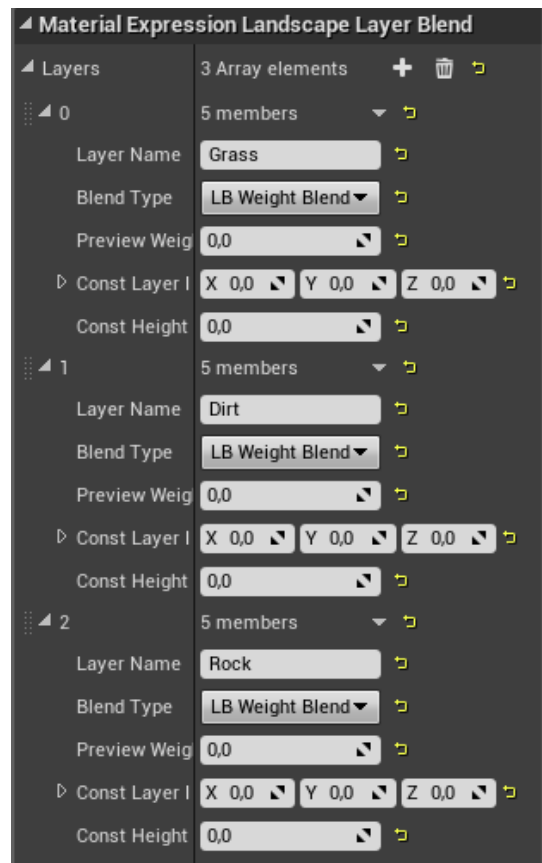


Рисунок 4.7 – Налаштування ноди Layer Blend

Кожен шар названий відповідно текстури, яка до нього підключається. У використаному матеріалі буде 3 текстури, а саме трава, бруд та каміння. У всіх шарів стоїть налаштування LB Weight Blend, що є оптимальним режимом змішування, коли не важливий порядок шарів.

Далі було додано відповідно текстури з папки Starter Content. Для більш реалістичного зображення в матеріалі також було додано карту нормалей, що зробило відображення матеріалу більш реалістичним (рис.4.8).

Після налаштування всіх шарів було розфарбовано ландшафт відповідно до висоти. До низин і нижніх частин пагорбів примінялась трава, до вершин пагорбів примінялось каміння, щоб створити перехідну лінію між травою та ґрунтом, було використано текстуру бруду. Таким чином було отримано реалістичне представлення щодо зміни типу покриття землі під час переходу від низин до пагорбів (рис.4.9).

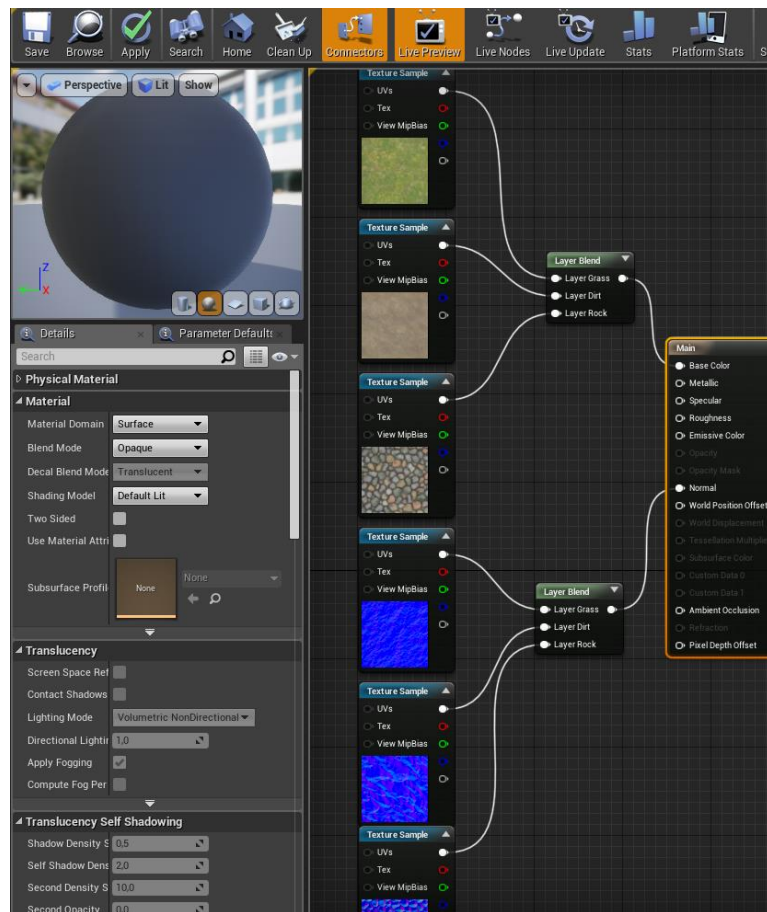


Рисунок 4.8 – Готовий матеріал для фарбування ландшафту

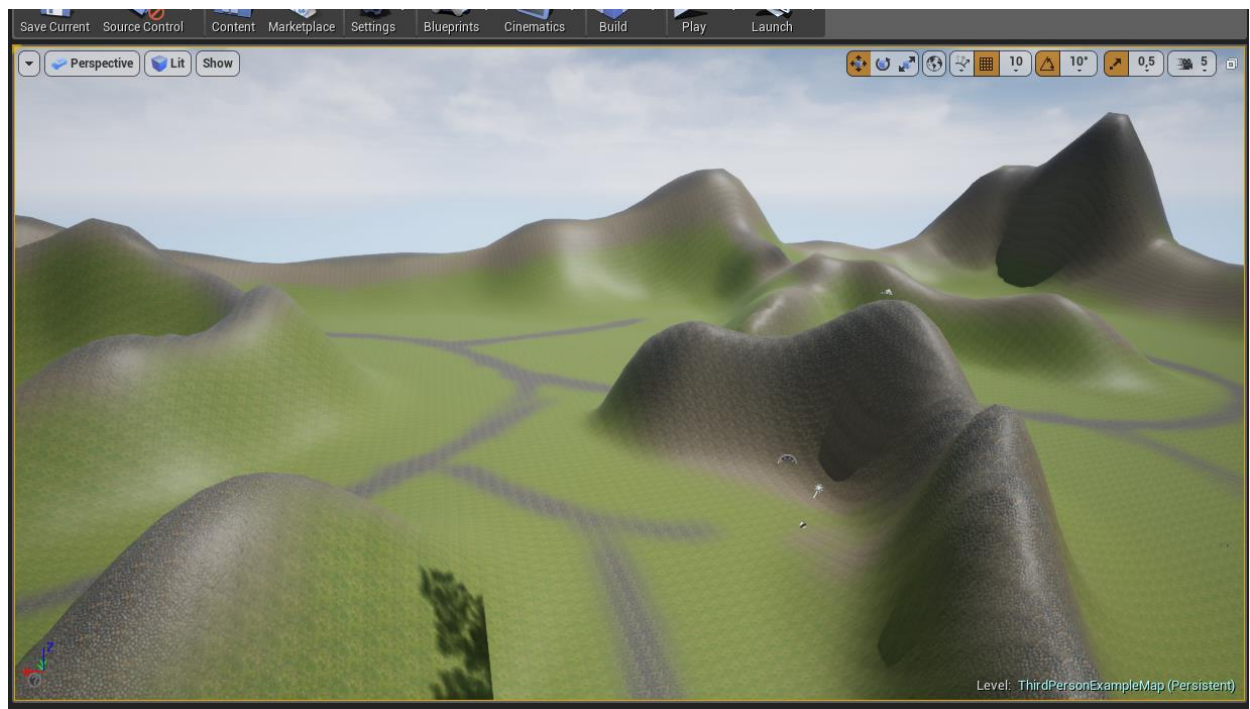


Рисунок 4.9 – Результат застосування матеріалу до ландшафту

Створення ландшафту в сцені дозволяє перейти до імпорту моделі церкви, яку було попередньо збережено в форматі fbx, та її розташування. Для цього попередньо створено структуру файлів проекту, яка дозволяє швидко орієнтуватись в файлах під час роботи над додатком. Назва кожної папки відповідає розташованим в ній фалам, а можливість зміни кольору папки дозволяє позначати, де знаходяться важливі файли (рис.4.10).

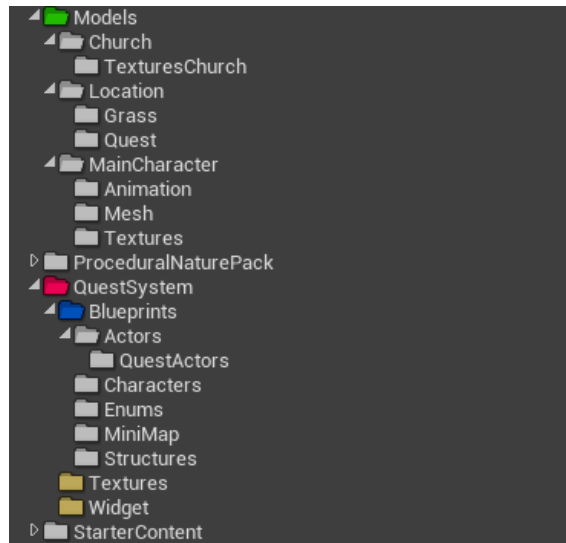


Рисунок 4.10 – Структура файлів проекту

Під час імпорту моделей необхідно встановити певні налаштування, а саме виключити автоматично створювану колізію, прослідити щоб координати при імпорті дорівнювали нулям, а масштаб був рівний одиниці (рис.4.11). Це допоможе правильно імпортувати геометрію моделей та матеріали.

Колізію для об'єкту буде встановлено за допомогою інструментів Unreal Engine 4. Після імпорту у проект були додані додаткові елементи, матеріали та текстури, які потім розміщені відповідно дизайну локації (рис.4.12). До певних об'єктів було примінено матеріали зі стартового набору Unreal Engine.

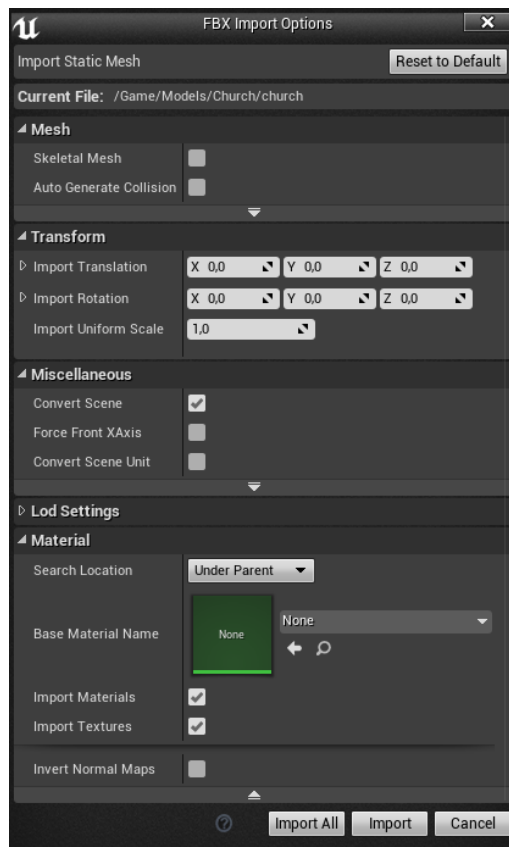


Рисунок 4.11 – Налаштування імпорту моделей в рушій



Рисунок 4.12 – Модель церкви в ігровій локації без матеріалу

Оскільки ігрова локація вимагає відповідного оформлення й не повинна бути пустою, вона була наповнена додатково моделями дерев, кущів та трави із бібліотек рушій. Для зручного розміщення рослинності в сцені використано інструмент Paint з підменю Foliage.

Для роботи з цим інструментом потрібно додати об'єкти типу meshes для дерев, трави та кущів, після чого налаштувати розмір пензля, інтенсивність розміщення та щільність розміщення. Завдяки зручному перемиканню між доданою рослинністю, можна окремо додавати в сцену траву чи дерева, а змінюючи розміри інструменту розміщення – розташування рослин в сцені стає більш детальним та наближеним до реальності (рис.4.13).

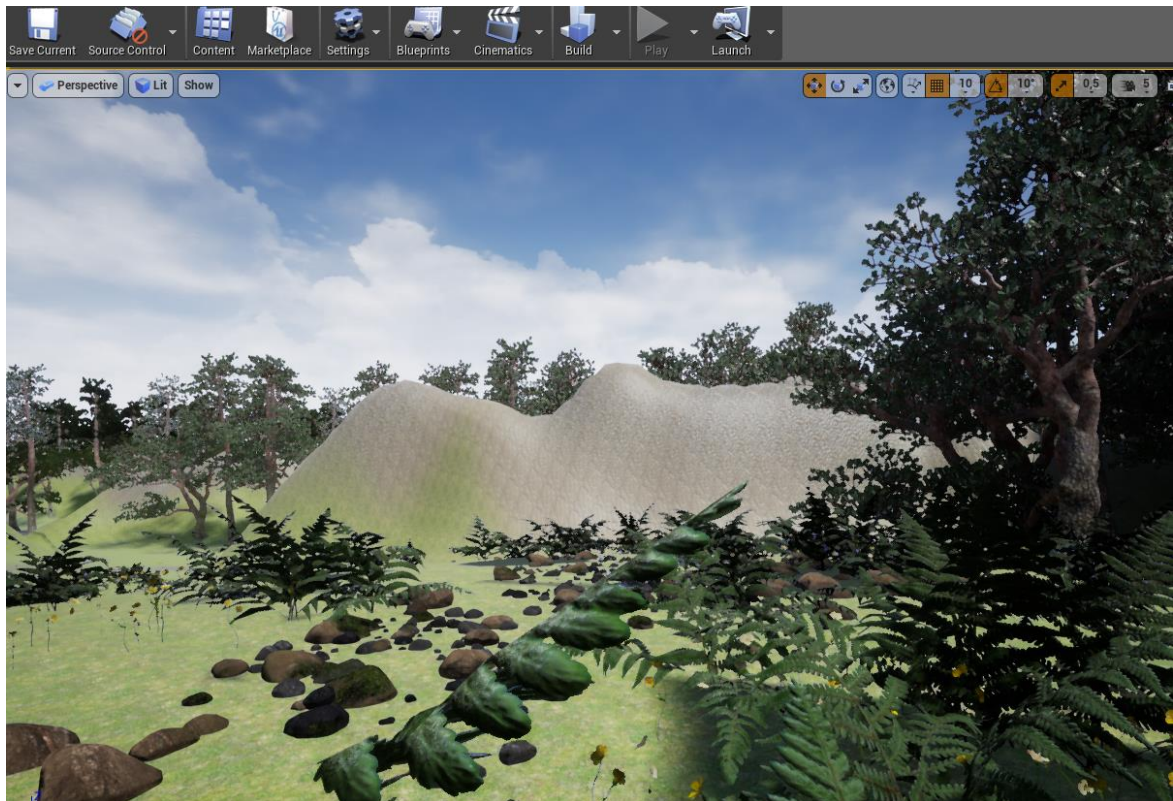


Рисунок 4.13 – Результат додавання рослинності до ландшафту

Після використання доступних в Unreal Engine 4 інструментів, сцена має необхідне оформлення для подальшого додавання персонажа та налаштування програмних зв'язків між об'єктами.

4.3 Налаштування ігрового персонажу

Стандартну модель ігрового персонажу було вирішено замінити на більш доцільну для даного додатку модель. З web ресурсу Міхато [31] було імпортовано модель персонажу, яка по своєму дизайну задовольняє потреби проекту (рис.4.14). Для моделі відповідно підбиралася анімація бігу, анімація очікування та анімація стрибку.

Після модель було експортовано в форматі fbx з налаштуваннями, які підійдуть для роботи в додатку (рис.4.15). Кожна анімація була імпортована лише зі скелетом, завдяки чому вдалося заощадити місце на жорсткому диску та швидко призначити кожену анімацію одній моделі.



Рисунок 4.14 – Головний персонаж на Міхато

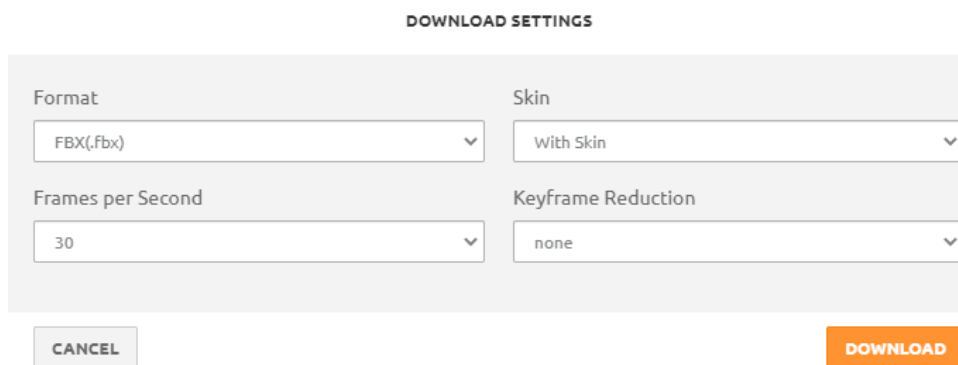


Рисунок 4.15 – Параметри експорту моделі головного персонажу

Далі були розроблені анімаційні Blueprint, де відбувалося безпосереднє налаштування анімації руху, а саме створювались «режими» переміщення – біг, швидкий біг, ходіння. Для кожного задавалась швидкість та призначалася відповідна анімація з файлів проекту (рис.4.16)

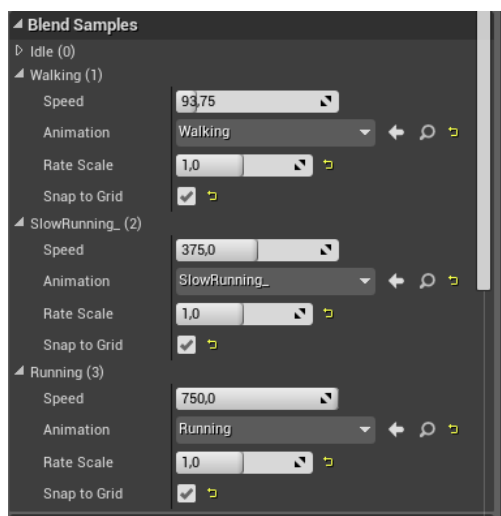


Рисунок 4.16 – Налаштування анімації персонажу

В проекті було реалізовано 3 режими переміщення: ходьба, швидкий біг та звичайний біг. Такий вибір був зроблений для зручності переміщення гравця по ігровій локації. Перемикання між режимами переміщення реалізовано за допомогою клавіш Left Shift, Left Ctrl (рис.4.17).

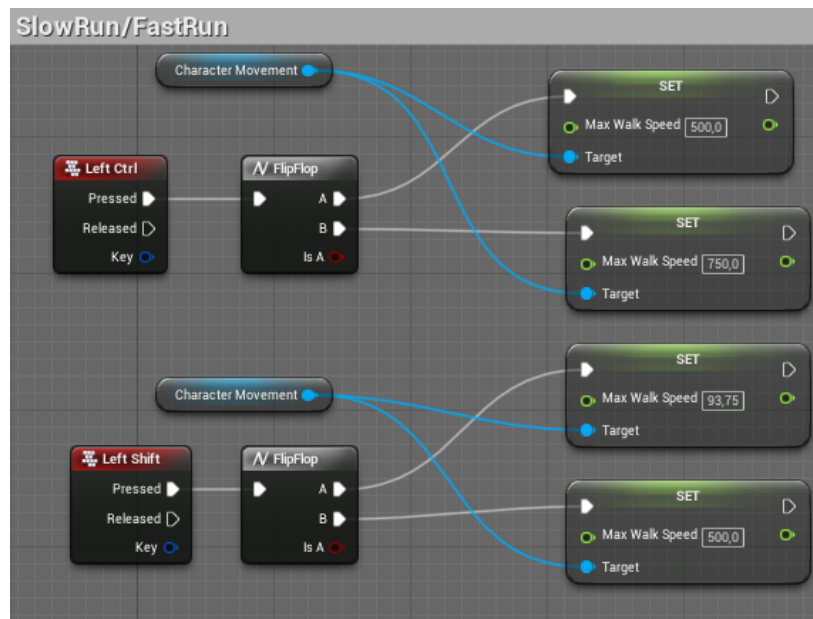


Рисунок 4.17 – Blueprint процедура переключення ходьби на біг

Оскільки гравець повинен мати повноцінну зможу переміщатися в ігровому просторі, було додано та налаштовано його анімації стрибка (за допомогою змінних перевірялося, де саме знаходиться персонаж і вмикалась потрібна анімація) (рис.4.18).

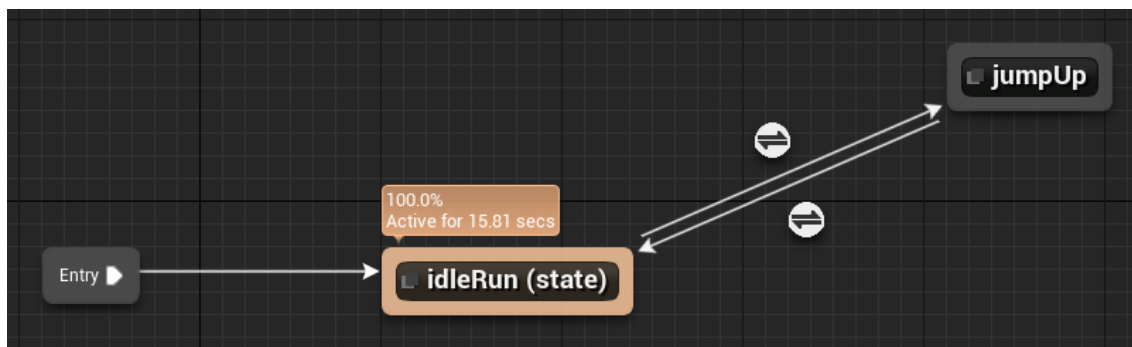


Рисунок 4.18– Анімаційний граф персонажу

Після закінчення роботи над анімацією персонажа, проєкт було протестовано на виявлення помилок під час анімації очікування, режимів переміщення, стрибків та взаємодії моделі зі скелетом тощо. Під час тестування подібних помилок виявлено не було, що дозволило перейти до створення інтерфейсів.

4.4 Розробка інтерфейсу додатку

Для повноцінного використання додатку було розроблено та реалізовано зрозумілий користувачу інтерфейс, який надає інформацію щодо квестових завдань та допомагає орієнтуватись в ігровій локації, а саме було створено ігрове меню з такими пунктами:

- Почати гру
- Правила
- Вихід

Кожний пункт меню максимально чітко доносить інформацію до гравця щодо функцій, які він виконує (рис.4.19). Далі було реалізовано меню паузи, яке має такі самі пункти, як і головне меню, але відрізняється дизайном (назва додатку була прибрана, а фон розміщення меню зроблений прозорим, щоб гравець міг бачити, де саме він знаходиться в ігровій сцені (рис.4.20).

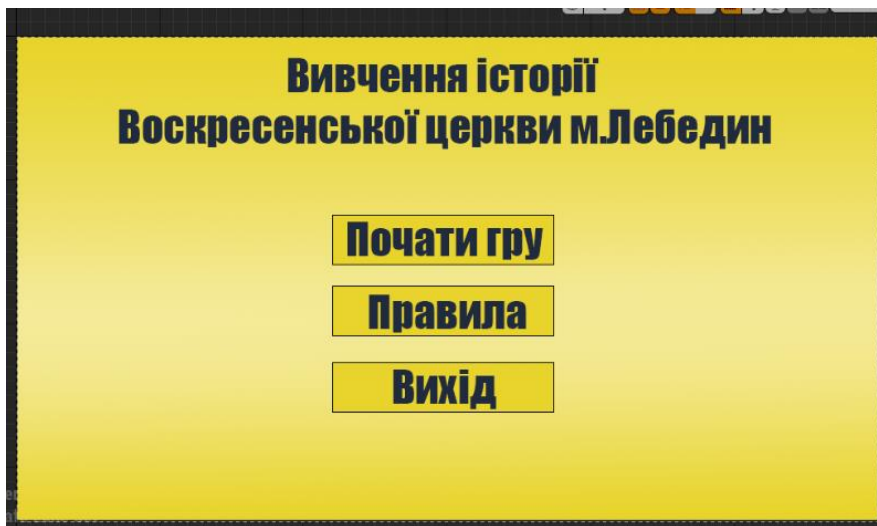


Рисунок 4.19 – Вигляд головного меню

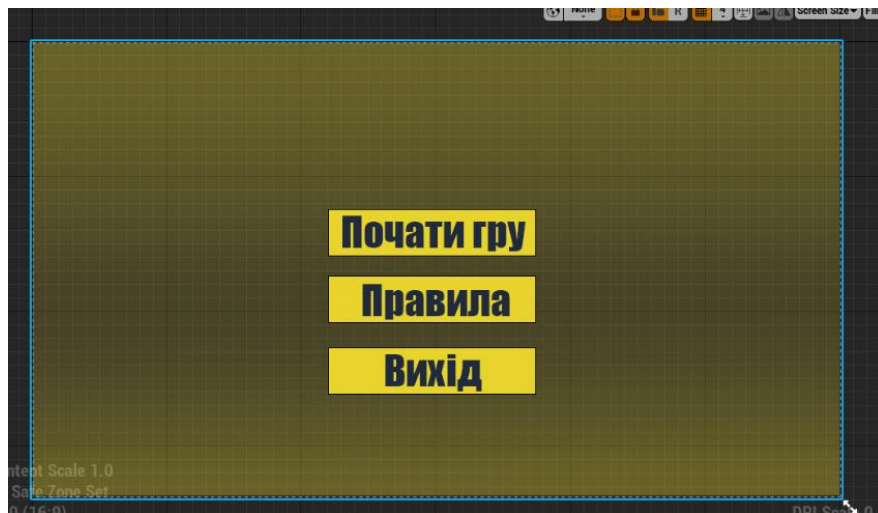


Рисунок 4.20 – Вигляд меню паузи

Створення елементів інтерфейсу відбувалося в проєкті за допомогою файлів Widget Blueprint та налаштування взаємодії миші з елементами інтерфейсу. Зокрема для всіх активних елементів інтерфейсу повинен бути ввімкнений параметр `is Variable`.

Щоб гравець мав змогу краще орієнтуватись в ігровій локації, було створено мінімапу (рис.4.21), на якій зображується гравець та елементи сцени. Мінімапу було реалізовано за допомогою віджету з додаванням матеріалу до Image, розміщеного на фоні інтерфейсу. Матеріал для мінімапи було створено з використанням допоміжної камери, розміщення її на потрібній висоті над персонажем та передачі зображення до текстури Render Target, яка в свою чергу задіяна в створеному матеріалі.



Рисунок 4.21 – Вигляд мінімапи

Щоб гравець мав змогу бачити кількість часу, що у нього залишилася, а також кількість квест-запитань, на які потрібно дати відповідь, було створено таймер зворотного відліку (рис.4.22). Ці елементи інтерфейсу були реалізовані за допомогою спеціальних функцій перекладу значень внутрішньо-програмних змінних у текстовий графічний вигляд із оновленням значень за допомогою відповідних процедур Blueprints.

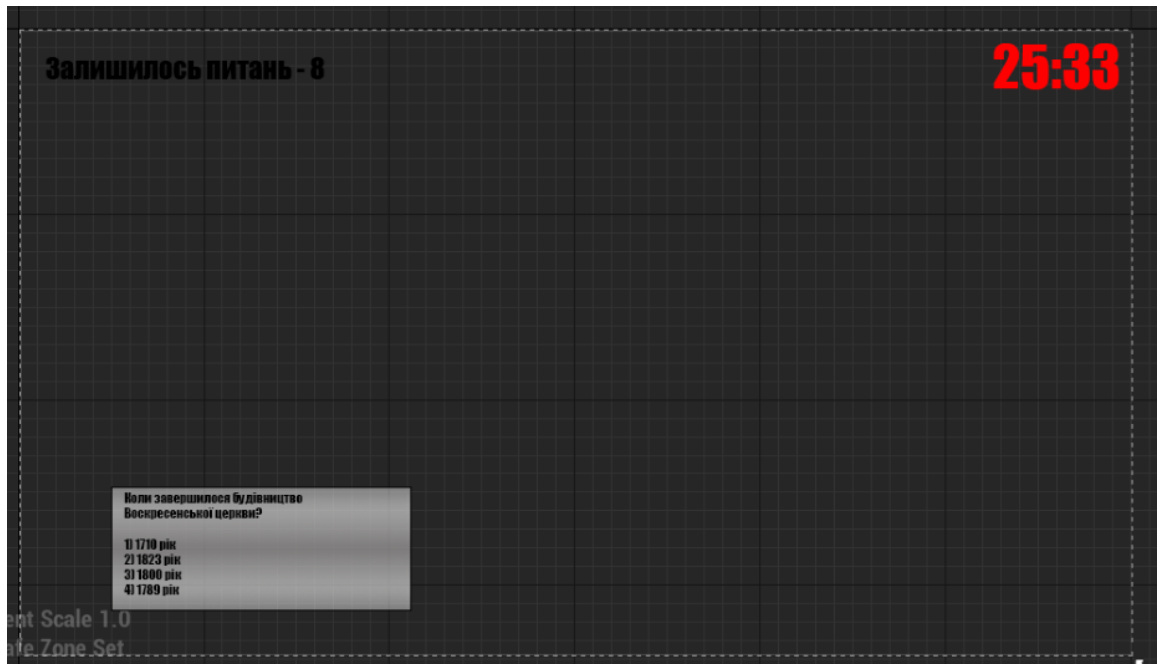


Рисунок 4.22 – Widget Blueprint таймеру та запитань

4.5 Реалізація квест-завдань

Для початку роботи над реалізацією квестових завдань потрібен файл Blueprint Enumeration для створення категорій подій (рис. 4.23). Було створено 3 основних типи подій в додатку:

1. Main quest – головне завдання додатку, яке показує скільки відповідей на запитання залишилось дати гравцеві.
2. Side Quest – поточне запитання, на яке гравець дає відповідь.

3. Event – подія, що відбувається після виконання всіх завдань (відкриття дверей до моделі церкви, доступ до наступних частин ігрової локації тощо).

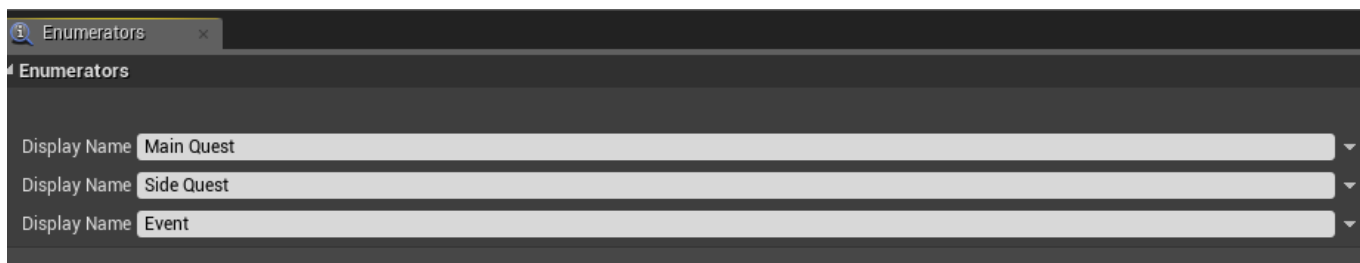


Рисунок 4.23 – Категорій подій

Також створюємо Blueprint Enumeration для поділу локацій на регіони (рис.4.24). Після виконання завдань в кожному регіоні буде відкриватись доступ до наступного.

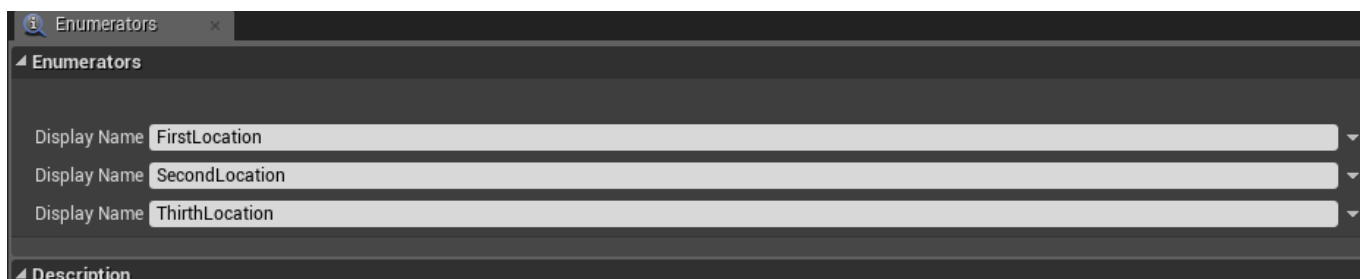


Рисунок 4.24 – Розділення локацій

Наступним кроком створюємо структуру, яка буде відповідати за перевірку й відображення квесту на мінімапі. Для цього відповідно створюємо дві змінних – HasLocation і Location. Змінна HasLocation типу Boolean, перевірятиме чи знаходиться квест в полі зору мінімапи для відображення. Друга змінна Location типу Vector, вона буде безпосередньо нести в собі інформацію про місцезнаходження квесту (рис. 4.25).

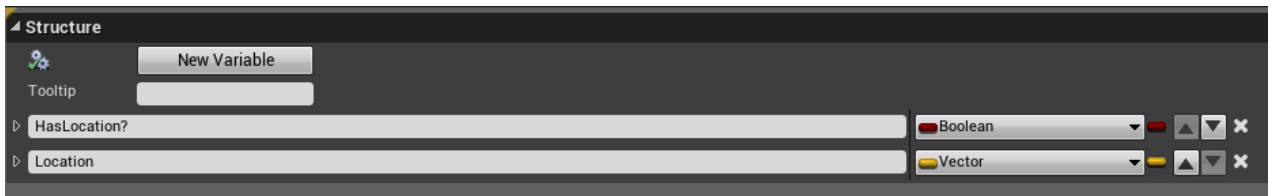


Рисунок 4.25 – Структура відображення квестів на мінімапі

Також потрібно створити структуру із загальною інформацією про завдання в додатку (рис 4.26). В ній створенні змінні, в яких буде зберігатись інформація про виконані квест-завдання, тобто текстова інформація, що виводиться на екран, інформація про розташування квесту, змінні, які перевіряють виконаний квест чи ні та змінні короткого опису квесту.

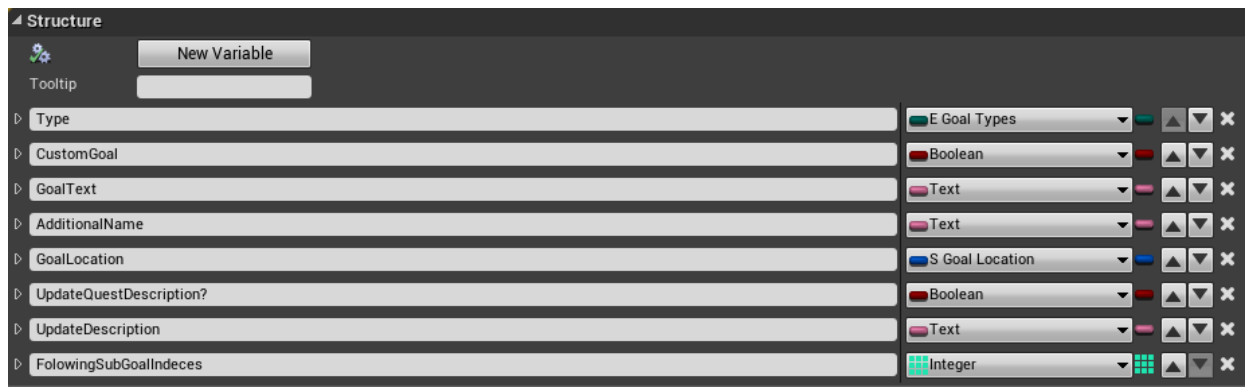


Рисунок 4.26 – Структура з інформацією про квест завдання

Останнім файлом Blueprints Structure буде QuestInfo. В ньому вказуватимуться інформація безпосередньо про кожен квест (рис. 4.27), його назву (Name), категорію (Category), опис (Description), інформацію про місцезнаходження квесту (Region), тип складності (Difficulty) тощо.

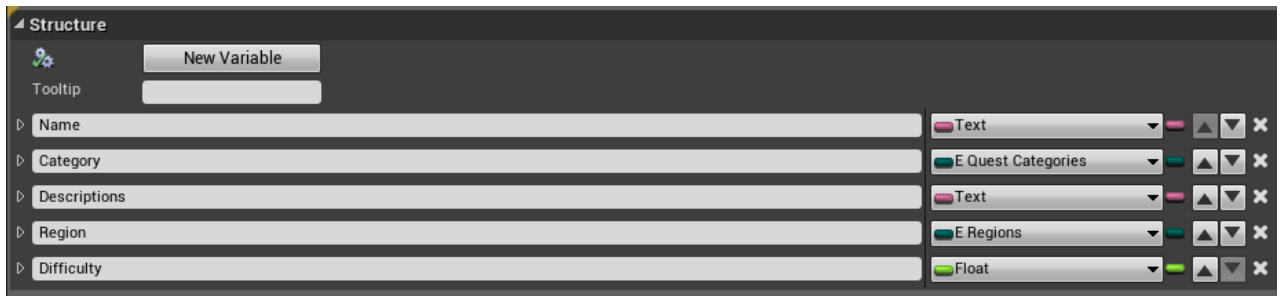


Рисунок 4.27 – Структура з інформацією про квест завдання

Для налаштування взаємодії між всіма структурами спочатку було створено новий Blueprint class Actor з назвою BP_MasterQuest, в якому створено та розділено на категорії нові змінні (рис. 4.28). Категорія Do Not Touch була створена для змінних, які не можна змінювати, оскільки вони потрібні для нащадків класу.

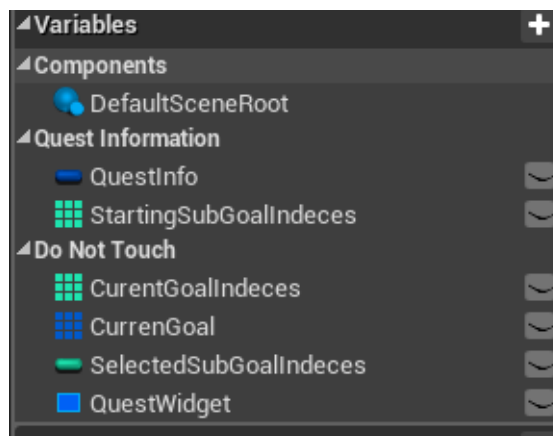


Рисунок 4.28 – Змінні файлу BP_MasterQuest

Після цього в файлі потрібно створити необхідні для роботи квестових завдань функції (рис. 4.29). Першою функцією даного класу буде UpdateSubGoals. В ній ми беремо поточний CurrentGoal, потім за допомогою нодів очищаємо змінну, підключаємо QuestInfo, а з неї вже через Break витягуємо SubGoals. Для отримання інформації з масиву CurrentGoalIndices використовуємо ноду ForEachLoop й додаємо до CurrentGoal через ADD отриманий індекс та інформацію з QuestInfo. Для

завершення налаштування функції на Completed ForEachLoop потрібно поставити Return (рис. 4.30). Створена функція дозволяє оновлювати інформацію щодо виконаних завдань.

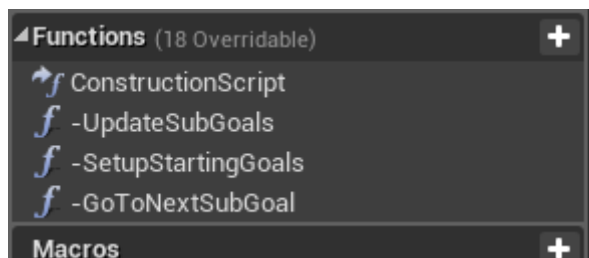


Рисунок 4.29 – Функції файлу BP_MasterQuest

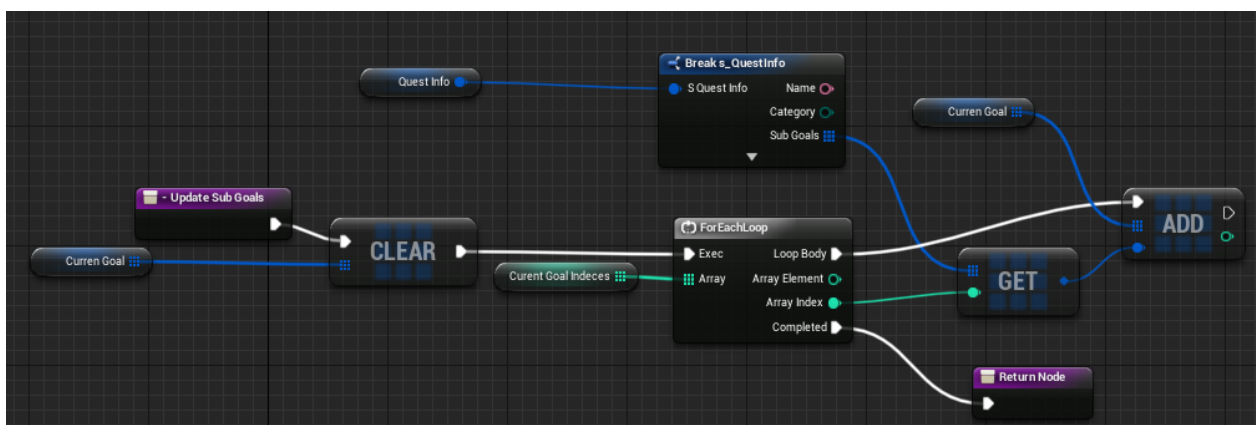


Рисунок 4.30 – Функція UpDateSubGoals

Далі створено функцію SetupStartingGoals для занесення інформації щодо нових квестів через функцію UpdateSubGoals. Для цього взято масив CurentGoalIndices, очищено інформацію в ньому через Clear та записано в нього StartingSubGoalIndices, після цього буде виконуватись попередньо створена функція UpdeteSubGoals (рис.4.31).

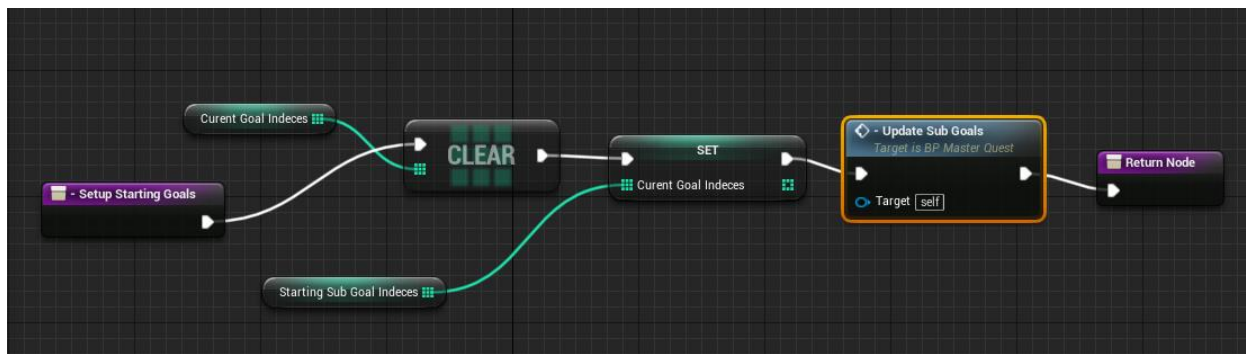


Рисунок 4.31 – Функція SetupStartingGoal

Останньою функцією в `BP_MasterQuest` є `GoToNextSubGoal` (рис.4.32). Для початку в неї додано `CurentGoalIndices`, з якого було взято змінну з найбільшим значенням, й порівняно з `QuestInfo`, а саме було взято поточний індекс цілі й перевірено чи більше він загальної кількості підцілей в `QuestInfo`. Якщо кількість менше або така сама кількість цілей, то `Branch` буде `False`, якщо ж більше – то інформація в `CurentGoalIndices` буде оновлена за допомогою виклику функції `UpdateSubGoals`.

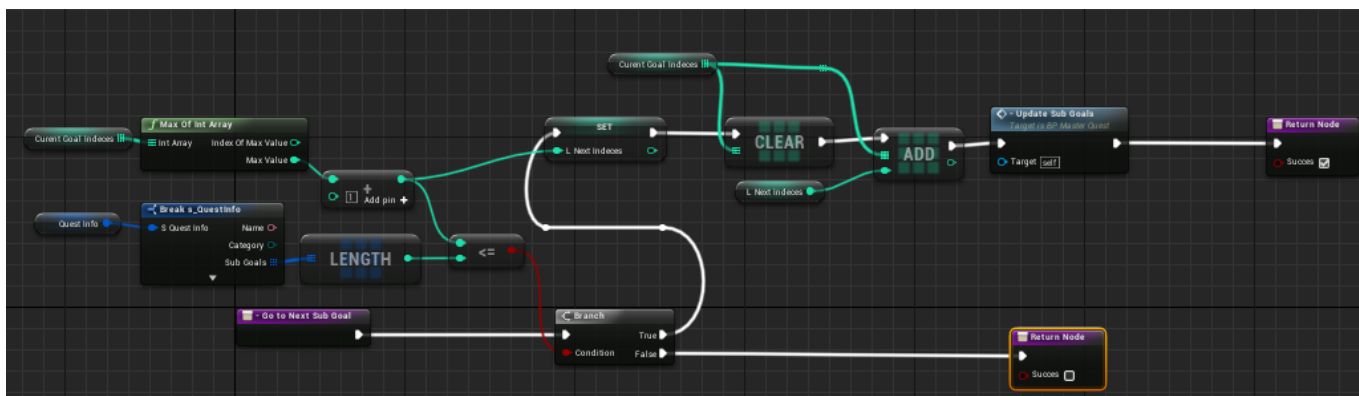


Рисунок 4.32 – Функція GoToNextSubGoal

Далі було створено новий Blueprint class Actor з назвою `BP_QuestManager` – завдяки даному класу в додатку здійснюється керування квестами. Спочатку було додано змінну `ThirdPersonChar`. Для неї встановлено параметри `Instance Editable Expose on Spawn`, так як коли буде використовуватись цей клас, потрібно буде

знаходити дані про персонажа. Наступною змінною додано MainWidget, через яку відбувається взаємодія з головним екраном. З аналогічними налаштуваннями в класі створювались інші необхідні для взаємодії з віджетами змінні, відмінним був лише тип цих змінних (рис.4.33).

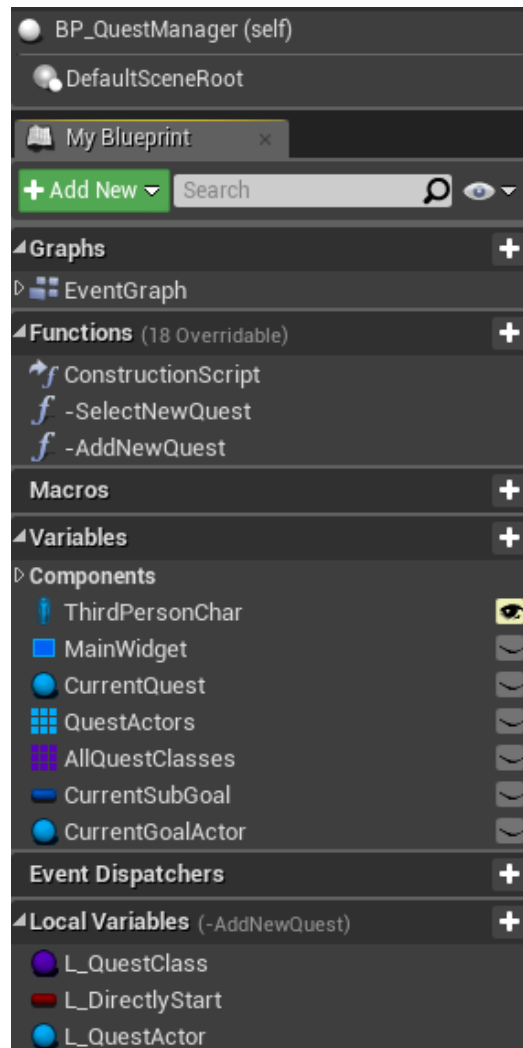
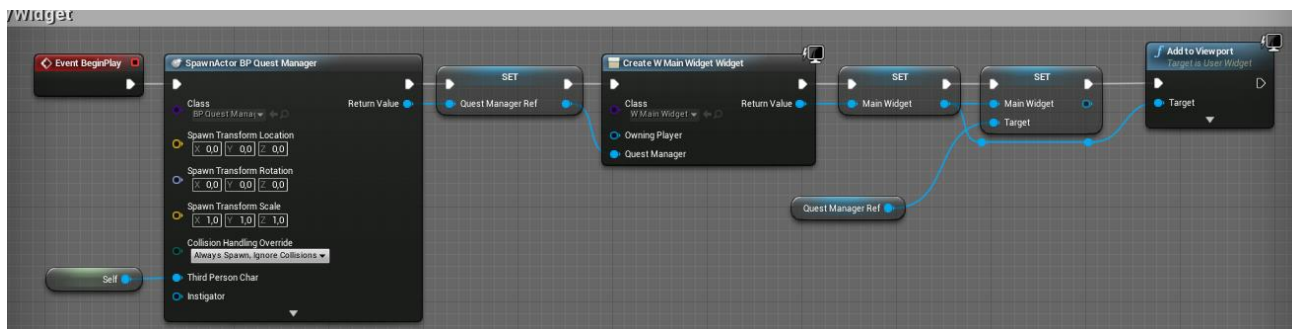


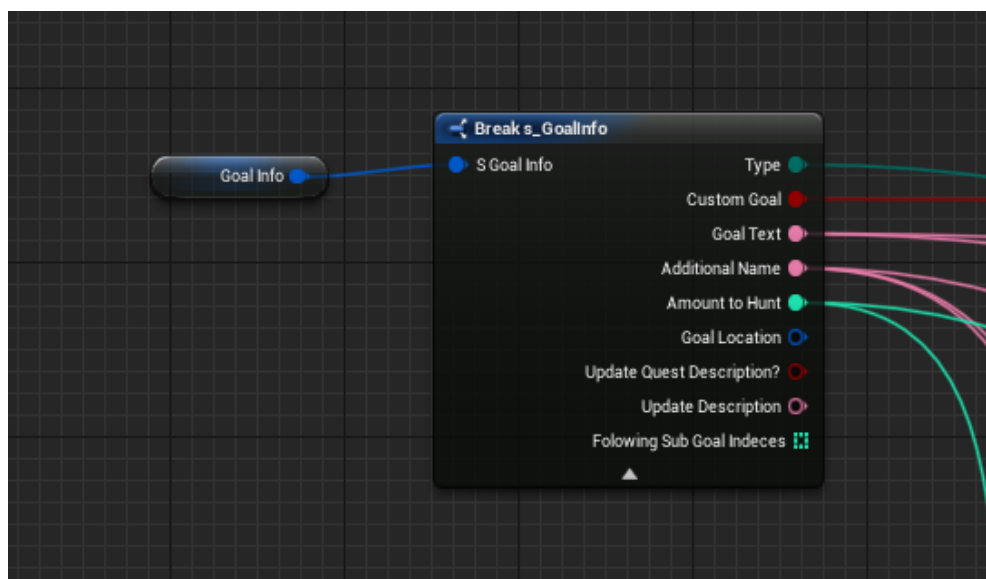
Рисунок 4.33 – Змінні та функції BP_QuestManager

Для відображення квестових завдань через віджет на екрані гравця було створено в файлі головного гравця (ThirdPersonCharacter) зв'язку нодів з Event BeginPlay, де виконується функція SpawnActor BP QuestManager з параметрами трансформації рівними нулям. Далі створено за допомогою ноди віджет Create MainWidget, підключено і реалізовано вивід на екран через Add to Viewport (рис.4.34).



4.34 – Blueprint реалізація показу на екрані квест завдань

Наступним кроком була реалізація за допомогою Blueprint зв'язків для відображення в SubGoalWidget. В функції Update відбувається перевірка на можливість оновлення, розбиття змінних на складові (рис.4.35) для витягування текстової інформації, та занесення інформації в Indexes.



4.35 – Реалізація отримання інформації з змінної GoalInfo

Для взаємодії гравця з квестами було налагоджено зв'язки між раніше створеними функціями та віджетами з безпосередньою взаємодією з гравцем (рис.4.36). Тому в файлі Blueprint w_QuestWidget було створено наступні функції:

1. GenerateSubWidget – створює новий віджет відповідно поточному завданню (рис.4.38).
2. UpdateQuest – перевіряє чи виконав гравець завдання й оновлює інформацію щодо відповіді гравця на поточне завдання.
3. CurentQuest? – функція перевіряє чи не намагається гравець повторно взяти вже виконаний квест.
4. SelectSubGoal – функція зчитує інформацію, які саме варіанти відповідей надає гравець.
5. -OnQuestSelected – робить неактивними завдання, які вже були взяті гравцем.

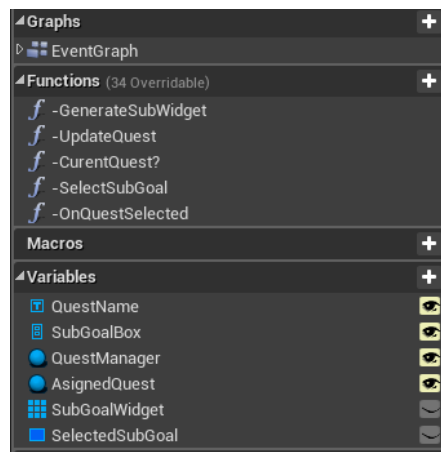


Рисунок 4.37 – Функції та змінні файлу w_QuestWidget

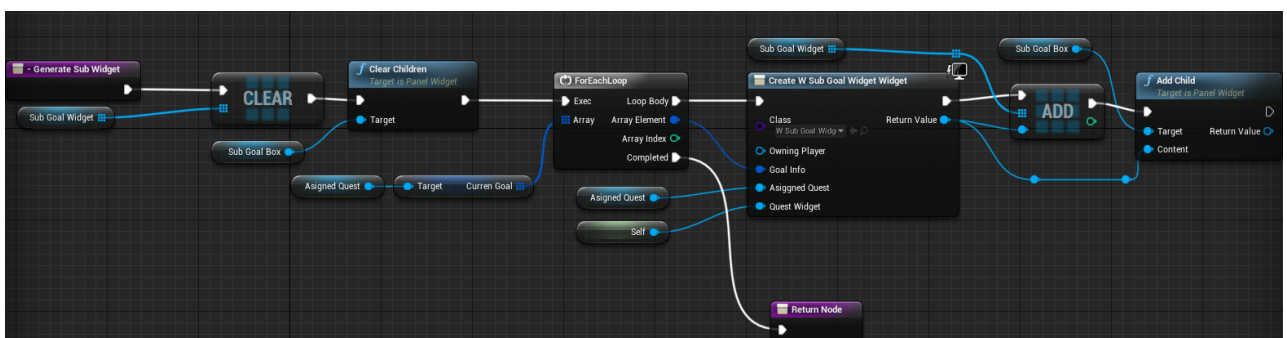


Рисунок 4.38 – Blueprint вигляд функції GenerateSubWidget

4.6 Тестування роботи додатку

Після закінчення роботи над всіма етапами розробки додаток було протестовано на предмет помилок. Після запуску потрібного файлу з назвою «ChurchTest» (рис.4.39) було показано головне меню додатку (рис.4.40).

Имя	Дата изменения
ChurchTest	09.12.2020 05:55
Engine	09.12.2020 05:55
ChurchTest.exe	09.12.2020 06:42
Manifest_NomUFSFiles_Win64.txt	09.12.2020 06:42

Рисунок 4.39 – Файли програми

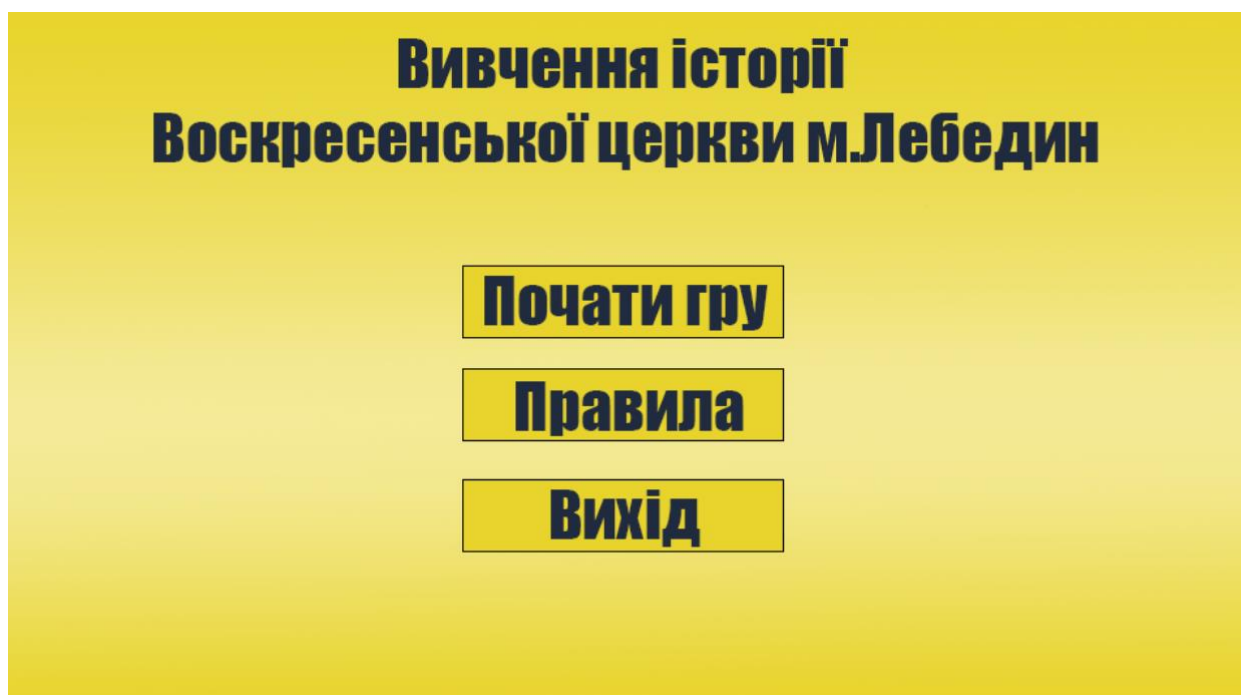


Рисунок 4.40 – Головне меню додатку

Для тестування було створено список можливих проблем, з якими може зіткнутися гравець під час використання ігрового додатку:

1. Проблеми з відображенням анімації стрибку.

2. Працездатність перемикання ходьби на біг і назад.
3. Працездатність віджетів виводити інформацію на екран.
4. Перехід між ігровими локаціями.

При натисканні клавіші Space у персонажа, як і потрібно, вмикається анімація стрибку. Під час анімації стрибку персонажа було оглянуто зі сторін, до яких камера має доступ. Проблем з текстурами, некоректною роботою SkeletalMesh виявлено не було (рис.4.41).

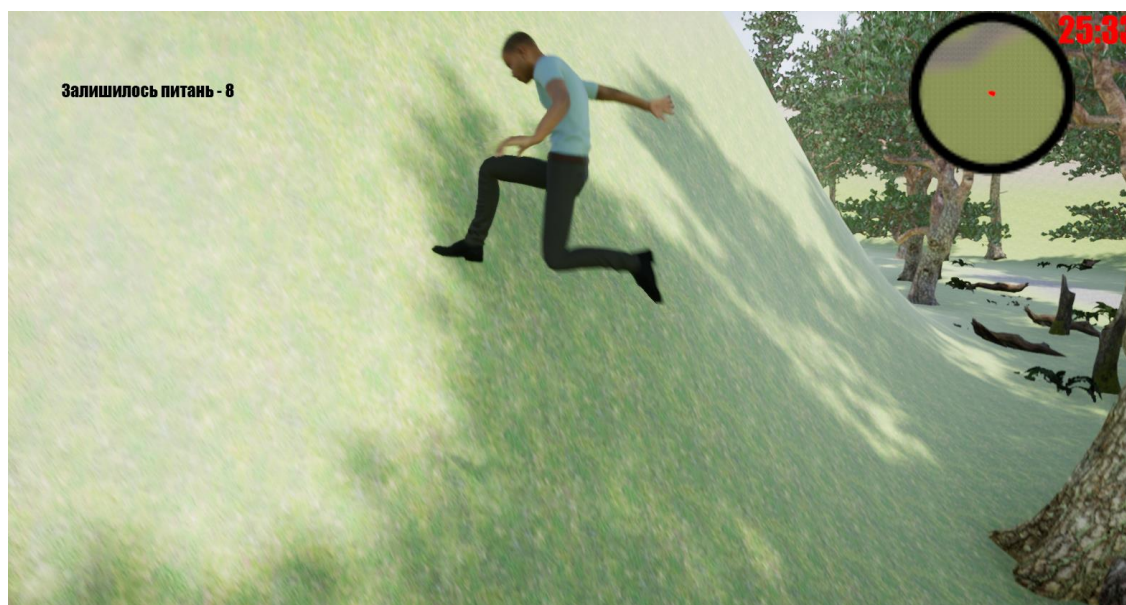


Рисунок 4.41 – Тестування анімації стрибку

Оскільки в додатку було реалізовано декілька анімацій переміщення ігрового персонажу, потрібно було впевнитись, що робота з ними була проведена коректно і не має критичних помилок. Після численних спроб створення нестандартних для анімацій бігу ситуацій (спроби застрягання в стовбурах дерев, забігання на пагорби і т.д.) було виявлено, що перемикання режимів переміщення працює правильно і помилок не виникає (рис.4.42).



Рисунок 4.42 – Перевірка анімації бігу

Тестування віджетів було успішним, всі елементи інтерфейсу працюють згідно завданню поставленому над проектом (рис.4.43).

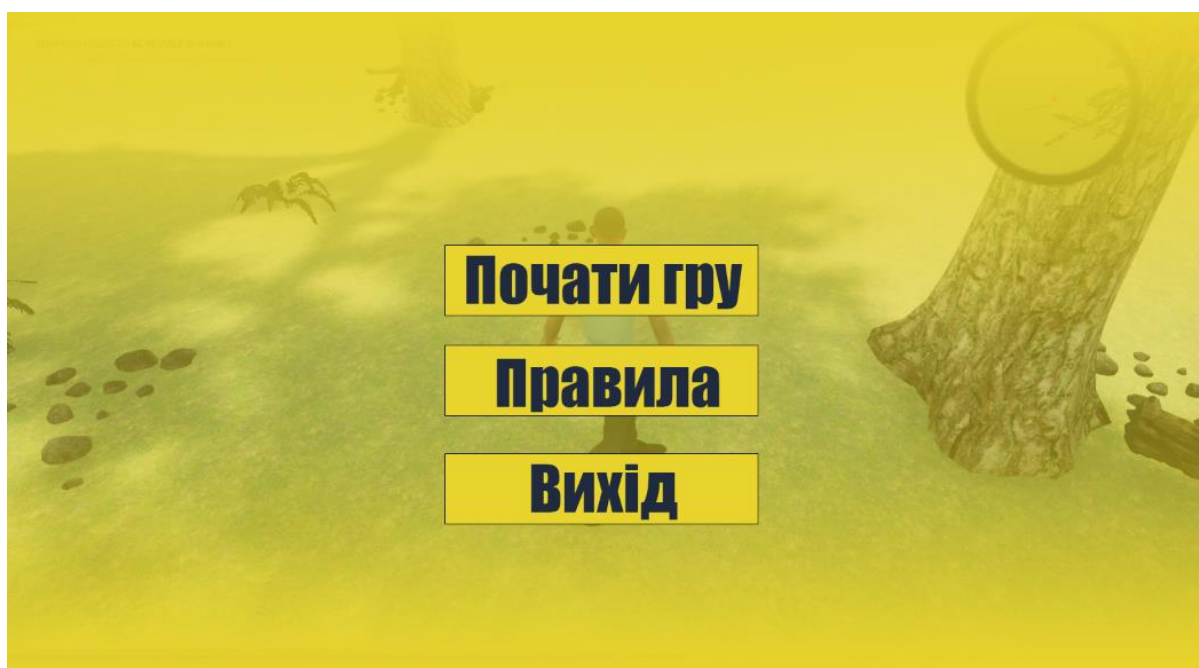


Рисунок 4.43 – Відображення меню паузи під час гри

Останнім етапом тестування є відкриття доступу до наступних частин локації після виконання завдань. Як показано на зображенні (рис.4.44) туман, що заважав переходу, поступово зникає під текстури відкриваючи шлях.



Рисунок 4.44 – Відкриття переходу до іншої частини локації

Таким чином можна зробити висновок, що поставлена задача виконана – у підсумку розроблено ігровий квест-додаток, реалізовані визначені при постановці задачі функції. Dodatok protestovano na відсутність помилок.

ВИСНОВКИ

Під час роботи над кваліфікаційною роботою було сформульовано мету роботи та визначено перелік задач для її досягнення.

Проведено аналіз предметної області, визначено, що для реалізації поставленої мети актуальним буде розробка проєкту у вигляді інтерактивного квест-додатку. Розглянуто аналогічні проєкти, визначено їх переваги та недоліки, на які слід звернути увагу при створенні власного проєкту.

В результаті аналізу інструментів, які могли б задовольнити всі аспекти роботи над проєктом було вирішено використовувати ігровий рушій Unreal Engine 4. Це дозволить створити додаток, який поєднає в собі інформаційний довідник та цікаву квест-гру, дозволить розвивати розумові здібності і одночасно отримувати нову інформацію.

Було розроблено структурно-функціональну модель роботи додатку, визначено варіанти використання додатку актором-гравцем, проведено планування роботи, розроблено календарний план реалізації проєкту.

Визначені етапи реалізації додатку, згідно з якими підготовлено 3D моделі до імпорту в рушій та сплановано розміщення сцени в рушії. Проведено налаштування необхідних матеріалів до моделей та використаних об'єктів рослинності.

В проєкті задіяно персонажа для реалізації інтерактивності та виконання квестових завдань, налаштовано управління персонажем та анімацію його рухів у додатку. Розроблено інтерфейс додатку, реалізовано логіку його роботи за допомогою Blueprint процедур.

Розроблений інтерактивний квест-додаток буде корисним при вивченні історичних фактів про Воскресенську церкву м.Лебедина і дозволить зацікавити користувачів до ознайомлення з історією рідного краю.

Квест-додатки мають величезний потенціал, так як націлені на розвиток індивідуальності, самостійності, ініціативності, пошукової активності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Нікіфоров О. В. Історичний квест як досвід освоєння національної культури / Олег Володимирович Нікіфоров // Вчені записки Новгородського державного університету імені Ярослава Мудрого / Олег Володимирович Нікіфоров. – Новгород, 2018.
2. Шевченко Л. О. Сюжетно-рольова гра [Електронний ресурс] / Л. О. Шевченко // Сонечко. – 2016. – Режим доступу до ресурсу: <http://dnz1.kupyansk.info/info/page/3680>.
3. Бутенко В. Г. Психолого-педагогічні основи ігрової діяльності дітей старшого дошкільного віку / В. Г. Бутенко // Збірник наукових праць [Херсонського державного університету]. Педагогічні науки. - 2016. - Вип. 72(1). - С. 51-56.
4. Урунтаєва, Г. А. Психологія дошкільного віку: підручник / Г. А. Урунтаєва. - М. : Академія, 2014.
5. Кічерова М. Н. Освітні квести як креативна педагогічна технологія для студентів нового покоління / М. Н. Кічерова, Г. З. Єфімова. // Світ Науки. - 2016. - №5. - С. 10.
6. Гончарова Н.О. Професійна компетентність учителя в системі навчання STEM// Наукові записки Малої академії наук України: зб. наук. праць. –К. : Інститут обдарованої дитини НАПН України, 2015. –Вип.7. –С. 141–148.
7. Дубова С. В. Використання квест-ігор в освітньому процесі, як спосіб мотивації учнів початкових класів [Електронний ресурс] / С. В. Дубова. – 2019. – Режим доступу до ресурсу: <https://vseosvita.ua/library/vikoristanna-kvest-igor-v-osvitnomu-procesi-ak-sposib-motivacii-uchniv-pocatkovih-klasiv-126773.html>.
8. Кпілова А. К. Квест-гра, або як йти в ногу з часом [Електронний ресурс] / А. К. Кпілова. - 2017. - Режим доступу до ресурсу: <http://io.nios.ru/articles2/87/9/kvest-igra-ili-kak-idti-v-nogu-so-vremenem>.

9. Осяк С. А. Освітній квест – сучасна інтерактивна технологія [Електронний ресурс] / С. А. Осяк, Т. В. Захарова, В. Н. Яковлева. - 2015. - Режим доступу до ресурсу: <https://www.science-education.ru/ru/article/view?id=20247>
10. FAULKNER J. Outer Wilds Review | Out of this world [Електронний ресурс] / JASON FAULKNER. – 2019. – URL: <https://www.gamerevolution.com/review/544997-outer-wilds-review>.
11. Рудь Д. Космічна пригода Outer Wilds долетить до Steam в червні [Електронний ресурс] / Д. Рудь. – 2020. – Режим доступу до ресурсу: <https://3dnews.ru/1006734>.
12. Coman D. Pro și contra ConQUIZtador [Електронний ресурс] / Dan Coman // Monitorul – URL: <https://www.monitorulsv.ro/Local/2008-01-15/Pro-si-contra-ConQUIZtador>.
13. Флэш (Flash). Почему не следует использовать на сайтах [Електронний ресурс]. – 2018. – URL: <https://ipipe.ru/info/flash>.
14. Обзор игры Quest Hunter [Електронний ресурс] // mmo13. – 2019. – URL: <https://mmo13.ru/games/quest-hunter>.
15. Unreal Engine vs Unity: Which is Better? [Електронний ресурс]. – 2020. – URL: <https://www.gamedesigning.org/engines/unity-vs-unreal/#:~:text=Unity%20is%20a%20very%20versatile,well%2Dmade%20that%20game%20was>.
16. Бауэр Е. К. Мобильные приложения в современном образовании [Електронний ресурс] / Екатерина Кондратьевна Бауэр. – 2016. – URL: <https://www.it-pedagog.ru/prilozheniya-v-sovremennom-obrazova>.
17. Блинцов О. Unreal Engine 4 для инди и прототипирования [Електронний ресурс] / Олександр Блинцов. – 2018. – URL: <https://vc.ru/pixonic/51306-ue4-guide>.
18. Бевза А. Unity3D или Unreal Engine 4 [Електронний ресурс] / Антон Бевза. – 2015. – URL: <https://stfalcon.com/ru/blog/post/unity3d-vs-unreal-engine-4>.
19. Unreal Engine 4 Terminology [Електронний ресурс]. – 2020. – URL: <https://docs.unrealengine.com/en-US/GettingStarted/Terminology/index.html>.
20. Создание редактора квестов и диалогов для Unreal engine: [Електронний ресурс] / deema35. – 2016. – URL: <http://www.gamedev.ru/code/forum/?id=221826>.

21. Murray J. W. C# Game Programming Cookbook for Unity 3D / Jeff Murray., 2014. – 452 с.
22. Хокинг Д. Unity в действии. Мультиплатформенная разработка на C# / Джозеф Хокинг. – Санкт-Петербург: Питер, 2019. – 352 с.
23. Romero M. Blueprints Visual Scripting for Unreal Engine / M. Romero, S. Brenden., 2019. – 380 с. – (Second; 2).
24. Cookson A. Unreal Engine 4 Game Development in 24 Hours, Sams Teach Yourself / A. Cookson, R. Dowlingsoka, C. Crumpler., 2016. – 496 с. – (First).
25. Sewell B. Blueprints Visual Scripting for Unreal Engine / Brenden Sewell., 2015. – 188 с.
26. Дизайн квестов в The Witcher 3: Wild Hunt [Электронный ресурс]. – 2016. – Режим доступа до ресурсу: <https://dplayer.ru/dizayn-kvestov-v-the-witcher-3-wild-hunt/>.
27. Хохлов В. Диалоги в играх: Программные средства и дизайнерские идеи [Электронный ресурс] / Владимир Хохлов. – 2017. – Режим доступа до ресурсу: <https://www.progamer.ru/dev/videogame-dialogues-tools-ideas.htm>.
28. Остроух А. В. Проектирование информационных систем / А. В. Остроух, Н. Е. Суркова., 2019. – 164 с.
29. Srivastava A. UML Modelling for Business Analysts: With Illustrated Examples / A. Srivastava, S. Desai, 2017. – 295 с.
30. Pilone D. UML 2.0 Pocket Reference: UML Syntax and Usage / Dan Pilone, 2006. – 138 с.
31. Mixamo Character Base [Электронный ресурс] // Adobe Systems Incorporated. – 2020. – Режим доступа до ресурсу: <https://www.mixamo.com/#/?page=1&type=Character>.

ДОДАТОК А. ПЛАНУВАННЯ РОБІТ

А.1 Деталізація мети проекту методом SMART

Продуктом дипломного проекту є інтерактивний квест-додаток для вивчення історії Воскресенської церкви (м. Лебедин). Результати деталізації методом SMART розміщені у таблиці А.1.

Таблиця А.1 – Деталізація мети методом SMART

Specific (конкретна)	Додаток надає змогу урізноманітнити навчальний процес, завдяки введенню методології квест-гри для вивчення історії. Така методологія дозволяє значно краще донести знання до учнів за рахунок того що надає змогу досягнути розглянути певний історичний аспект більш детально. Що в цілому сприяє гарному засвоєнню матеріалу.
Measurable (вимірювана)	При використанні інтерактивного додатку вивчення історичних відомостей щодо Воскресенської церкви (м. Лебедин) покращується на 35%, відносно звичайного процесу вивчення матеріалу.
Achievable (досяжна)	Для отримання готового додатку потрібно отримати представлення про схожі за функціоналом додатки, визначити баланс між інтерактивністю та ефективністю засвоєння матеріалу.
Relevant (реалістична)	Для виконання проекту потрібно створити додаток відповідно до реальної історичної будівлі.
Time-framed (обмежена у часі)	Проект виконується згідно наданого часу і буде завершений відповідно до встановлених термінів.

A.2 Планування структури робіт

Для загального розуміння обсягу проекту була розроблена структура розподілу робіт (WBS). Завдяки цьому ієрархічному опису роботи, було отримано представлення щодо кожного етапу проекту, оскільки кожен рівень в WBS представляє дедалі детальніший опис результатів.

Правильно розроблена WBS діаграма описує заплановані результати, а не заплановані дії, а завдяки ієрархічній структурі робота над кожним результатом стає значно простішою, особливо для інтерактивних додатків, в розробці яких присутні непов'язані процеси (рис.А.1).

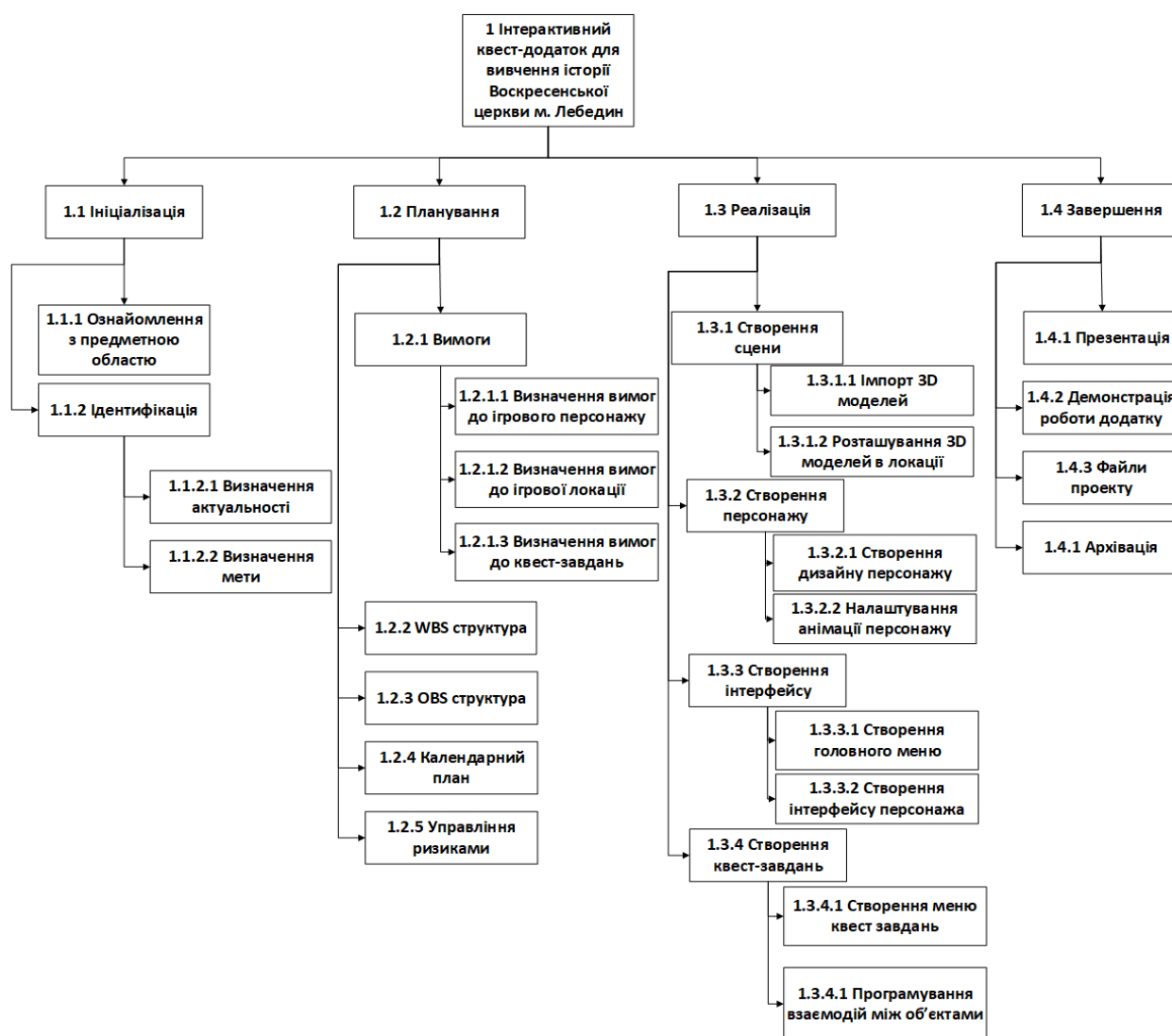


Рисунок А.1 – WBS. Структура робіт проекту

Після побудови WBS було створено структуру виконавців OBS. Організаційна структура проекту спирається безпосередньо на учасників проекту, які працюють в певній частині роботи над додатком.

Організаційну структуру підрозділу створено з використанням діаграм для чіткої демонстрації учасників проекту. Для роботи над проектом було створено OBS, що за своєю структурою нагадувала дерево; кожна гілка якого вказує на робочі відносини та відповідальність. Чим ближче до вершини OBS, тим більше обов'язків має людина.

Щоб отримати чітке розуміння робочого процесу, було розроблено детальну структуру робіт, в якій блоки діаграми є зручним інструментом розподілу робіт між учасниками проекту. Увагу було зосереджено на досягненні індивідуальних, керованих цілей і прагненню до ефективної та результативної роботи.

Діаграма OBS зображена на рисунку А.2.

Список виконавців, що функціонують в проекті, наведено в таблиці А.2.

Таблиця А.2

Роль	Ім'я	Проектна роль
Проектувальник	Дементієнко Д.С.	Підготовлює 3D моделі до імпорту в ігровий рушій. Створює дизайн персонажу. Реалізовує зв'язки між об'єктами сцени та програмну частину проекту.
Консультант проекту	Баранова І.В.	Формує завдання на розробку проекту. Надає уточнення по технічному завданню проекту.
Менеджер проекту	Дементієнко Д.С.	Відповідає за терміном роботи над проектом планом виконання задач. Створює документацію проекту.

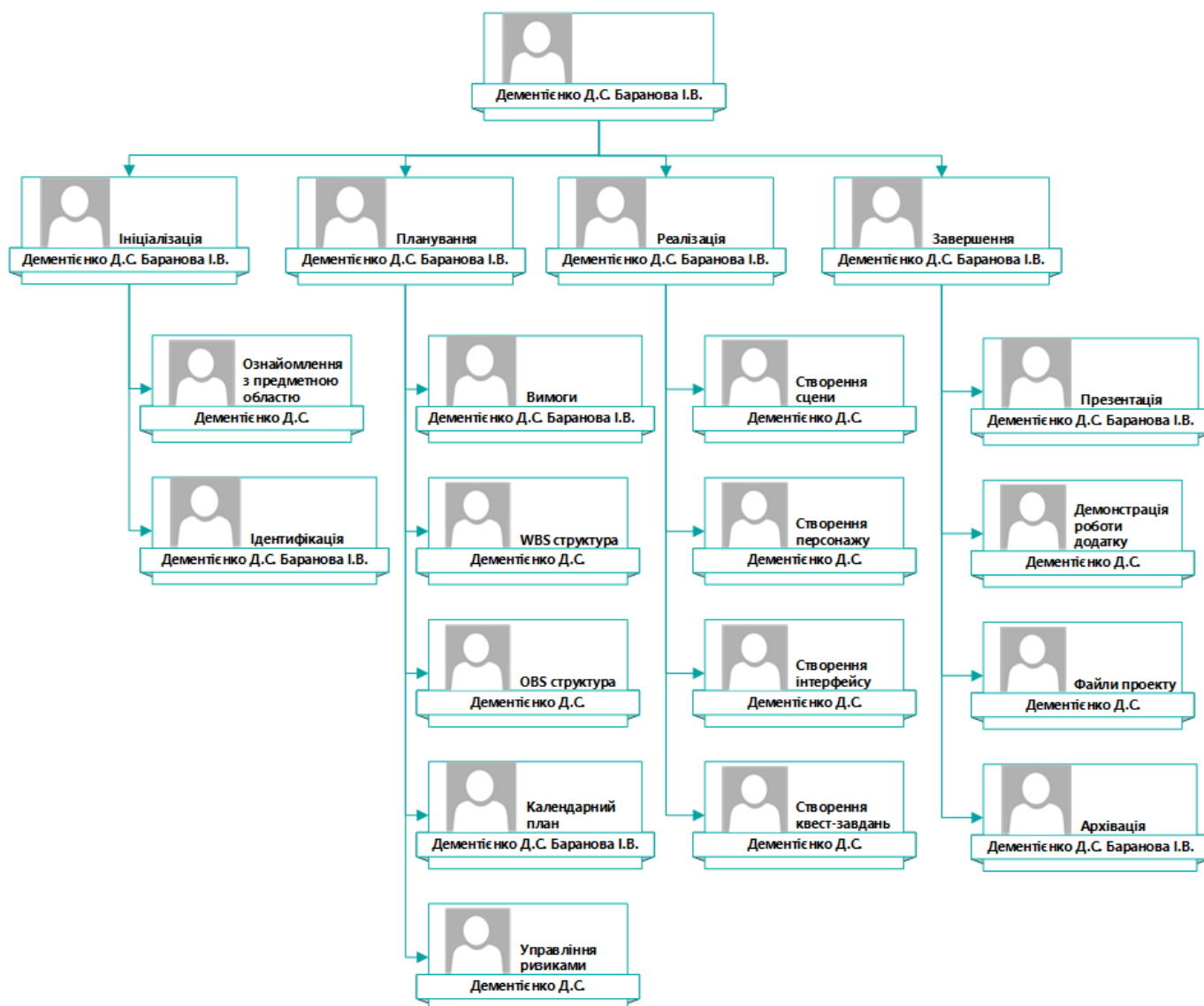


Рисунок А.2 – OBS Структура робіт проекту

А.3 Побудова календарного графіка виконання ІТ проекту

Для підвищення ефективності командної роботи над проектом було створено діаграму Ганта, яка також прискорить виконання робочих процесів та забезпечить своєчасне виконання завдань. Крім цього, діаграма Ганта є наочним способом відображення запланованих завдань, ілюструючи хід цілого проекту та всіх його частин.

Завдяки зручному вигляду горизонтальних графів, діаграму можна використовувати для проектів з достатньою кількістю процесів, яким і є створюваний проект. Діаграми досить зручні, щоб показати, яку роботу потрібно виконати в певний

день та час. Вони також допомагають команді відстежувати дати початку та етапів проекту для правильного розподілу часу.

Результат побудови діаграми Ганта представлений на рисунку 1.3. Відповідно до тривалості етапів проекту будуються прямокутники, які схематично зображають тривалість проекту, зв'язки між проектами зображуються лініями.

А.4 Оцінка ризиків проекту

Будь який проект при розробці схильний до ризику. Для прогнозування та мінімізації ризиків виконується їх аналіз та управління. До головних ризиків створення інтерактивного квест-додатку для вивчення історії Воскресенської церкви (м. Лебедин) відноситься:

- відставання від графіку робіт;
- проблеми з програмним забезпеченням;
- помилки під час проектування;
- відсутність резервних копій даних;
- недостатня кваліфікація розробника;
- зростання вимог до проекту.

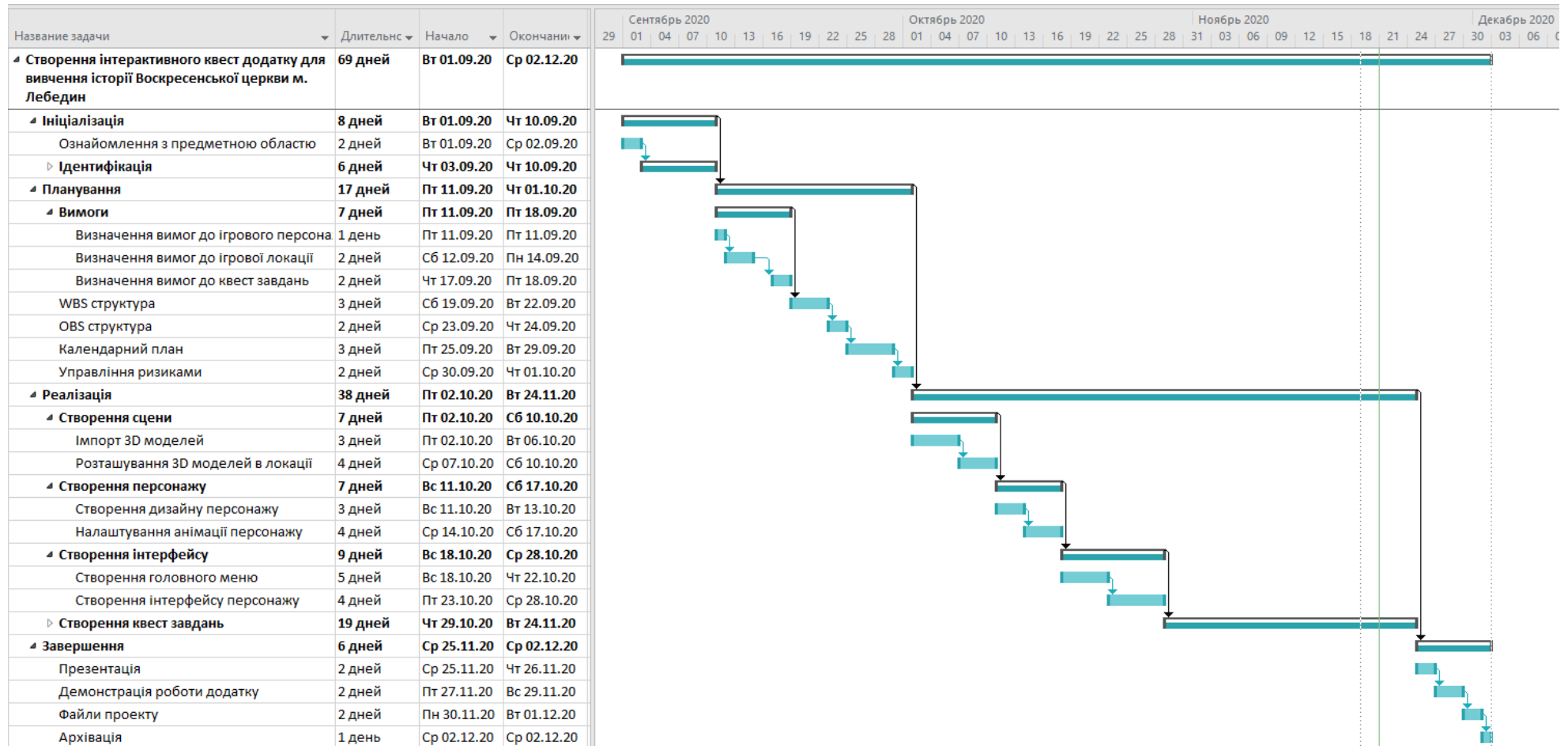


Рисунок А.3 – Діаграма Ганта

Таблиця А.3 – Ймовірність виникнення і величина ризику

№	Ризики	Виникнення	Втрати
1	відставання від графіку робіт	3	2
2	проблеми з програмним забезпеченням	2	3
3	помилки під час проектування	2	2
4	відсутність резервних копій даних	1	3
5	недостатня кваліфікація розробника	3	3
6	зростання вимог до проекту	1	2

Проаналізувавши ризики за ймовірністю виникнення та втратами, можна розділити їх за рівнем впливу на:

- прийнятні:
 - зростання вимог до проекту;
- виправдані:
 - помилки під час проектування;
 - відсутність резервних копій даних;
 - відставання від графіку робіт;
 - проблеми з програмним забезпеченням;
- неприпустимі:
 - недостатня кваліфікація розробника.

В результаті аналізу ризиків було визначено ймовірність їх виникнення та можливі втрати. Згідно двох визначених параметрів для проекту було створено матрицю «Ймовірність – Втрати», завдяки якій вдалося розділити ризики на рівні впливу (рис.А.4).

Виникнення	Відставання від графіку робіт недостатня кваліфікація розробника	3	3	6	9
	Проблеми з програмним забезпеченням	2	2	4	6
	Помилки під час проектування Відсутність резервних копій даних	1	1	2	3
	Зростання вимог до проекту		1	2	3

Зростання вимог до проекту
Відставання від графіку робіт
Помилки під час проектування
Відсутність резервних копій даних
Проблеми з програмним забезпеченням
Недостатня кваліфікація розробника

Втрати

Рисунок А.4 – Матриця «Ймовірність – Втрати»

Більшість ризиків виявилася виправданими і втрати при їх виникненні не нанесуть суттєвої шкоди проекту. До рівня «Недопустимі» вийшло віднести лише один ризик – «Недостатня кваліфікація розробника», але завдяки аналізу розробник проекту матиме змогу мінімізувати вплив можливого виникнення такого ризику.

ДОДАТОК Б. ДЕМОНСТРАЦІЯ РОБОТИ ДОДАТКУ

Приклад роботи додатку з проходженням квестових завдань та взаємодією ігровими меню наведено на рис.Б.1 – Б.4.



Рисунок Б.1 - Надання відповіді на квест-запитання

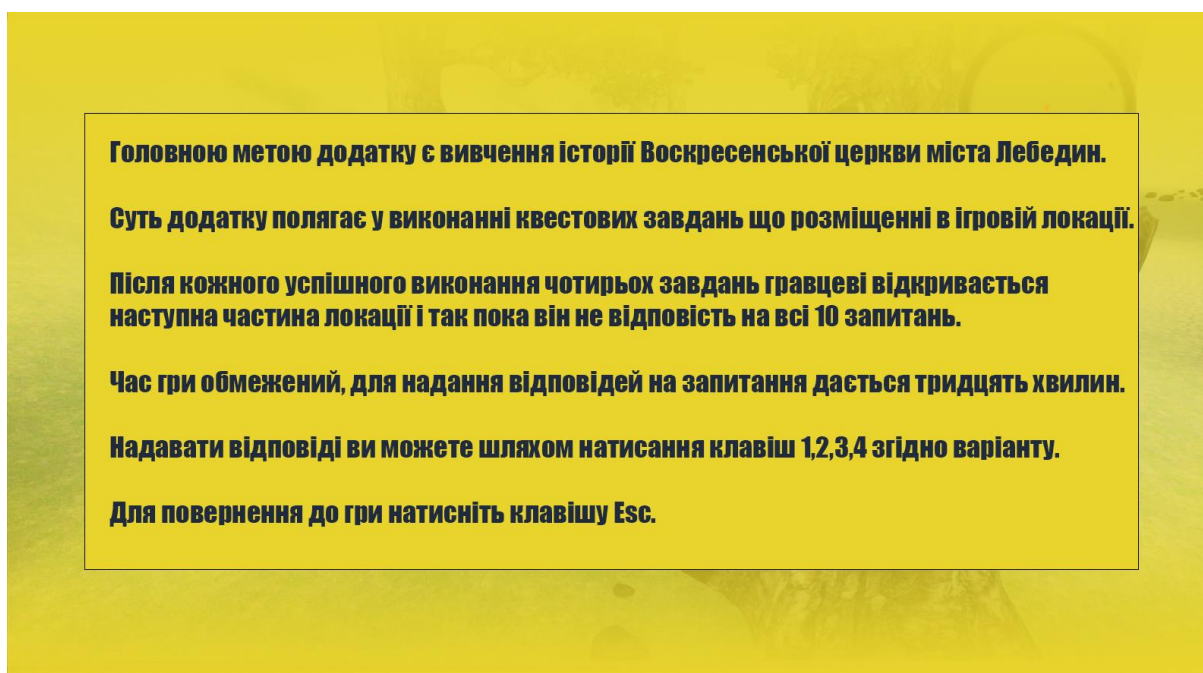


Рисунок Б.2 – Використання пункту меню «Правила»



Рисунок Б.3 – Надання відповіді на квест запитання



Рисунок Б.4 – Фінальний бонус за проходження квесту гравець (Огляд архітектури будівлі з висоти)