

Міністерство освіти і науки України
Одеський національний технологічний університет
Кафедра комп'ютерної інженерії



**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ**

на тему Створення віртуального світу
(назва кваліфікаційної роботи згідно наказу ОНТУ)
в Unreal Engine

Здобувача Крисань І.С.
(прізвище, ініціали)

4 курсу 542a групи

Керівники: к.т.н., доц. Шестопалов С.В.
(посада, прізвище та ініціали)

ст. викл. Жуковецька С.Л.
(посада, прізвище та ініціали)

Консультанти: _____
(посада, прізвище та ініціали)

д.е.н., проф. Басюркіна Н.Й.
(посада, прізвище та ініціали)

Кваліфікаційна робота допускається до захисту

Рішення кафедри від 05.06 2024 р., протокол № 8

Завідувач кафедри комп. інженерії _____ Сергій АРТЕМЕНКО
(назва кафедри) (підпис) (Ім'я ПРІЗВИЩЕ)

Одеса – 2024 рік

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерної інженерії, програмування та кіберзахисту
Кафедра комп'ютерної інженерії
Ступінь вищої освіти бакалавр
Спеціальність 123 «Комп'ютерна інженерія»
Освітня програма Розробка ігор та інтерактивних медіа у віртуальній
реальності

ЗАТВЕРДЖУЮ

Зав. кафедри комп'ютерної інженерії
Сергій АРТЕМЕНКО
« 30 » серпня 2023 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Крисань Івана Сергійовича

1. Тема роботи Створення віртуального світу в Unreal Engine

Затверджена наказом університету від « 30 » серпня 2023 р., наказ № 442-03

2 Термін здачі здобувачем закінченої роботи 28 травня 2024 р.

3. Вихідні дані роботи

1. Програма для 3D моделювання «Blender». 2. Unreal Engine 5. 3. Ассети для Unreal Engine 5. 4. Текстовий редактор Microsoft Word. 5. Редактор презентації Microsoft PowerPoint. 6. Програма для текстурування «Substance painter»

4. Перелік питань, які потрібно розробити

1. Вступ. 2. Збір та аналіз інформації. 3. Проектування
4. Розробка сцени. 5. Економічні розрахунки.
6. Охорона праці. 7. Загальні висновки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Слайд 2. Вступ. Слайд 5. Невірна геометрія. Слайд 4. Несумісність форматів
Слайд 5. Невірна геометрія. Слайд 6. Невідповідність масштабів.
Слайд 9. Модель меча у Unreal Engine 5. Слайд 10. Висновок

6. Консультанти по роботі, із зазначенням розділів роботи, що стосуються їх

| Розділ | Консультант | Підпис, дата | |
|----------------------|-------------------------------------|----------------|------------------|
| | | Завдання видав | Завдання прийняв |
| <i>Економіка</i> | <i>Phd, ст. викл. Богданов О.О.</i> | | |
| <i>Охорона праці</i> | <i>ст. викл. Жуковецька С.Л.</i> | | |
| <i>Нормоконтроль</i> | <i>ст. викл. Жуковецька С.Л.</i> | | |

7. Дата видачі завдання 30.08.2023

Керівники

Сергій ШЕСТОПАЛОВ

Світлана ЖУКОВЕЦЬКА

Завдання прийняв до виконання

Іван КРИСАНЬ

КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів кваліфікаційної роботи | Термін виконання етапів роботи | Примітка |
|----|---|--------------------------------|----------|
| 1. | <i>Дослідження предметної області</i> | <i>26.10.2023</i> | |
| 2. | <i>Дослідження існуючих аналогів</i> | <i>30.11.2023</i> | |
| 3. | <i>Дослідження методів створення віртуального світу</i> | <i>28.01.2023</i> | |
| 4. | <i>Проектування</i> | <i>15.02.2024</i> | |
| 5. | <i>Розробка демонстраційної версії ПЗ</i> | <i>27.03.2024</i> | |
| 6. | <i>Підготовка техніко-економічної частини</i> | <i>15.04.2024</i> | |
| 7. | <i>Підготовка розділу охорони праці</i> | <i>15.04.2024</i> | |
| 8. | <i>Оформлення пояснювальної записки</i> | <i>27.05.2024</i> | |
| 9. | <i>Оформлення графічної частини та лістингу</i> | <i>27.05.2024</i> | |

Здобувач-дипломник Іван КРИСАНЬ

Керівники роботи Сергій ШЕСТОПАЛОВ

Світлана ЖУКОВЕЦЬКА

Несу відповідальність за ідентичність електронного та друкованого варіантів кваліфікаційної роботи, даю згоду на обробку персональних даних та не заперечую проти розміщення кваліфікаційної роботи на офіційних web-ресурсах ОНТУ.

Підтверджую, що в кваліфікаційній роботі відсутні порушення норм академічної доброчесності.

Здобувач-дипломник Іван КРИСАНЬ

АНОТАЦІЯ

Кваліфікаційна робота присвячена створенню сцени віртуального світу в ігровому рушії *Unreal Engine*. Сцена «меч у камені» включає озеро, у центрі якого розташований меч, вбудований у камінь. Для досягнення високоякісного візуального результату та реалістичності оточення було використано різні інструменти та технології.

У першому розділі розглянуто особливості розроблення віртуальних світів, а також проведено аналіз наявних аналогів і методів їхньої реалізації.

У другому розділі сформовано технічне завдання та розроблено пайплайн роботи, включно з було розібрано технологію моделювання, текстурювання та створення віртуального світу в *Unreal Engine*.

У третьому розділі обґрунтовано вибір інструментів, їхній опис і розписано процес розробки. Модель меча було створено у *Blender*, текстуровано в *Substance Painter* та інтегровано в *Unreal Engine*, а потім зроблено сцену на рушії.

У четвертому розділі проведено оцінку ефективності створення цієї сцени, а також розглянуто маркетинговий, науково-технічний, економічний, соціальний та екологічний ефекти від розробки проекту. У п'ятому розділі розглянуто питання про охорону праці.

Результатом роботи є демонстраційна версія сцени в *Unreal Engine*, що включає озеро, меч у камені, туман, створений за допомогою *Blueprint*, і дощ, реалізований з використанням системи *Niagara*.

Ключові слова: рушії *Unreal Engine*, віртуальний світ, *Blender*, *Substance Painter*, *Blueprint*, *Niagara*, сцена «меча в камені».

ABSTRACT

The qualifying work is devoted to creating a virtual world scene in the Unreal Engine game engine. The "sword in the stone" scene involves a lake in the center of which is a sword embedded in the stone. Various tools and technologies were used to achieve high-quality visual results and realistic environments.

The first chapter discusses the features of the development of virtual worlds, as well as an analysis of existing analogues and methods for their implementation.

In the second section, a technical task was formed and a work pipeline was developed, including technology for modeling, texturing and creating a virtual world in Unreal Engine.

The third chapter justifies the choice of tools, their description and outlines the development process. The sword model was created in Blender, textured in Substance Painter and integrated into Unreal Engine, and then the scene was made in engine.

The fourth chapter assesses the effectiveness of creating this scene, and also examines the marketing, scientific, technical, economic, social and environmental effects of the development of the project. The fifth chapter discusses the issue of labor protection.

The result is a demo scene in Unreal Engine, including a lake, a sword in the stone, fog created using Blueprint, and rain implemented using the Niagara system.

Keywords: *Unreal Engine, virtual world, Blender, Substance Painter, Blueprint, Niagara, "sword in the stone" scene.*

ЗМІСТ

| | стор. |
|---|-------|
| ВСТУП | 7 |
| РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ | 12 |
| 1.1 Дослідження віртуальних світів та їх класифікація | 12 |
| 1.1.1 Поняття віртуального світу..... | 12 |
| 1.1.2 Класифікація віртуальних світів..... | 14 |
| 1.1.3 Інструмент створення віртуального світу..... | 15 |
| 1.1.4 Спосіб взаємодії..... | 18 |
| 1.2 Приклади використання віртуальних світів | 19 |
| 1.2.1 У фільмах..... | 20 |
| 1.2.2 У іграх..... | 21 |
| 1.2.3 Інше..... | 24 |
| 1.3 Постановка завдання..... | 25 |
| Висновок до першого розділу..... | 25 |
| РОЗДІЛ 2 ПРОЕКТУВАННЯ | 26 |
| 2.1 Розробка пайплайну | 26 |
| 2.2 Проектування віртуального світу..... | 28 |
| 2.2.1 Формування концепії..... | 28 |
| 2.2.2 Розробка коцепії меча..... | 28 |
| 2.2.3 Розробка концепії ландшафту..... | 30 |
| 2.2.4 Розробка концепії наповнення сцени..... | 31 |
| 2.2.5 Пошук референсів..... | 32 |
| 2.3 Огляд технології створення об'єктів..... | 33 |
| 2.3.1 Технологія моделювання..... | 33 |
| 2.3.2 Технологія текстурування..... | 36 |

| | | | | | | | | |
|---------------|-------------|---------------------|---------------|-------------|---|----------------------|-------------|----------------|
| | | | | | КРБ.КІ.1.442-03.1.10 | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | Створення віртуального світу в Unreal Engine | Літ. | Арк. | Акрушів |
| Розробив | | Іван КРИСАНЬ | | | | | 6 | 116 |
| Перевірів | | Світлана ЖУКОВЕЦЬКА | | | | | | |
| Рецензент | | Денис СНИГУР | | | | | | |
| Нормоконтроль | | Світлана ЖУКОВЕЦЬКА | | | | | | |
| Затвердив | | Сергій АРТЕМЕНКО | | | | | | |
| | | | | | | гр. 542, ОНТУ | | |

| | | |
|-----------------------------------|--|----|
| 2.4 | Технологія створення віртуальних світів у <i>Unreal Engine</i> | 39 |
| 2.4.1 | Можливості <i>Unreal Engine</i> | 39 |
| 2.4.2 | Використання систем частинок..... | 43 |
| 2.4.3 | Використання блюпринтів | 45 |
| | Висновок до другого розділу | 47 |
| РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОЕКТУ | | 48 |
| 3.1 | Вибір програмного забезпечення | 48 |
| 3.1.1 | Інструмент моделювання | 48 |
| 3.1.2 | Інструмент текстурування..... | 50 |
| 3.2 | Моделювання меча | 53 |
| 3.3 | Текстурування меча | 60 |
| 3.4 | Побудова сцени у <i>Unreal Engine</i> | 64 |
| 3.4.1 | Імпорт моделей..... | 66 |
| 3.4.2 | Використання системи частинок..... | 67 |
| 3.4.3 | Використання блюпринтів | 68 |
| 3.4.3 | Використання освітлення..... | 70 |
| 3.5 | Створення відео..... | 71 |
| 3.5.1 | Додавання камери і її анімація | 71 |
| 3.5.2 | Візуалізація і постобробка | 73 |
| | Висновок до третього розділу..... | 74 |
| РОЗДІЛ 4 ЕКОНОМІЧНА ЧАСТИНА | | 76 |
| 4.1 | Організаційно-економічне та маркетингове обґрунтування проект | 76 |
| 4.1.1 | Порівняльний аналіз пропонованого проекту | 76 |
| 4.1.2 | Основні положення..... | 81 |
| 4.1.3 | Маркетингове обґрунтування проекту | 84 |
| 4.2 | Розрахунки ціни програмного продукту | 87 |
| 4.2.1 | Розрахунок капітальних витрат | 92 |
| 4.2.2 | Розрахунок показників економічної ефективності проекту..... | 93 |
| 4.3 | Бізнес план стартап-проекту..... | 94 |
| | Висновок до четвертого розділу..... | 96 |

| | |
|---|-----|
| РОЗДІЛ 5 ОХОРОНА ПРАЦІ | 97 |
| 5.1 Основні положення охорони праці | 97 |
| 5.2 Недоліки та умови роботи за комп'ютером | 98 |
| 5.3 Електробезпека..... | 99 |
| 5.4 Пожежна безпека при роботі з комп'ютером..... | 101 |
| Висновок до п'ятого розділу | 102 |
| ЗАГАЛЬНІ ВИСНОВКИ | 103 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 105 |
| ДОДАТКИ..... | 107 |

| | | | | | | |
|-------------|-------------|-----------------|---------------|------------|-----------------------------|-------------|
| | | | | | <i>КРБ.КІ.1.442-03.1.10</i> | <i>Арк.</i> |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дат</i> | | 8 |

ВСТУП

Віртуальні світи – це цифрові простори, створені за допомогою спеціалізованих технологій і програмного забезпечення, що дозволяють користувачам взаємодіяти один з одним і з віртуальним середовищем. Вони можуть імітувати реальність або бути повністю вигаданими, і є важливою частиною сучасної технології.

Віртуальні світи мають значний вплив на сучасне суспільство, відкриваючи нові можливості для освіти, розваг, соціальної взаємодії та бізнесу. Забезпечують інноваційні методи навчання, дозволяючи студентам занурюватися у вивчення предметів через інтерактивні симуляції та віртуальні лабораторії. Створюють нові форми соціальних взаємодій, дозволяючи людям спілкуватися та співпрацювати у віртуальних середовищах, незважаючи на географічні бар'єри. Вони також відкривають нові економічні можливості для бізнесу та підприємництва, включаючи віртуальні ринки та нові форми реклами і маркетингу. Однак використання віртуальних світів викликає також ряд етичних та соціальних питань, таких як захист персональних даних та віртуальна приватність.

Майбутнє віртуальних світів виглядає дуже перспективним завдяки розвитку технологій. Інтеграція елементів віртуальних світів у реальний світ через технології розширеної реальності відкриває нові можливості для взаємодії та навчання. Використання штучного інтелекту дозволяє створювати розумні, адаптивні віртуальні середовища, що підлаштовуються під потреби та поведінку користувачів. Ідея створення єдиного, взаємопов'язаного віртуального світу, який поєднує різні платформи та сервіси, стає все більш реальною завдяки розширенню технології [25].

Актуальність теми – це пов'язано зі стрімким розвитком технологій, популярністю віртуальних середовищ, їхнім різноманітним застосуванням та високим попитом на кваліфікованих спеціалістів у цій галузі.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 9 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

Таким чином, віртуальні світи надають унікальну можливість для створення та дослідження нових просторів, які не пов'язані з реальними фізичними законами. Вони стали важливою частиною інформаційного світу і продовжують розвиватися, відкриваючи нові горизонти для творчості, навчання та взаємодії.

Метою роботи є створення віртуального світа у Unreal Engine 5.

Об'єктом дослідження є процеси створення віртуального світа у Unreal Engine 5.

Предметом дослідження є створення віртуального світа у Unreal Engine 5.

Основні задачі, які потрібно вирішити в ході роботи:

1. Дослідити інші віртуальні світи та способи створення.
2. Проаналізувати існуючі варіанти віртуальних світів.
3. Проаналізувати можливості двигуна Unreal Engine 5.
4. Створити модель віртуального світу.
5. На підставі моделі розробити відеоролик.

Методи розробки. Використовуються методи комп'ютерної графіки для створення 3D-моделей. Також використовуються методи алгоритмічні методи для налаштування різних елементів сцени, як ефекти частинок і освітлення, і методи інформатики, що включає в себе оптимізацію продуктивності сцени для плавної роботи.

Наукова новизна кваліфікаційної роботи полягає у комплексному підході до створення віртуального світу з використанням сучасних інструментів та технологій.

Практичне значення одержаних результатів. Практичне значення отриманих результатів. Розроблено сцену в ігровому рушії, також створено модель у 3D-редакторі та затекстуровано у спеціальній програмі. Розроблено демонстраційний відеоролик для сцени з використанням впроваджених методів. Сцена та модель можуть бути використані в навчальному процесі здобувачів освіти університету, а також для продажу через спеціалізовані онлайн-майданчики в якості курсів.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 10 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

Апробація результатів роботи і публікація. Крисань І.С. дослідження проблем переносу моделі з 3D-редактора в ігровий рушій. Крисань І.С., Жуковецька С.Л. // Матеріали ХХІV Всеукраїнської науково-технічної конференції молодих вчених, аспірантів і студентів, Одеса, 18-19 квітня 2024 р. – Одеса, Видавництво ОНТУ, 2024 р. – С. 402 – 403.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 11 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

РОЗДІЛ 1

ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Дослідження віртуальних світів та їх класифікація

1.1.1 Поняття віртуального світу

Перша згадка про віртуальну реальність відноситься до 1950-х років, коли американський математик і вчений Норберт Вінер представив концепцію "кібернетичного простору". Це був віртуальний світ, створений за допомогою комп'ютерних систем, який міг імітувати реальне середовище і давав змогу людям взаємодіяти з ним.

Однак на той час технології створення повноцінних віртуальних світів ще не існувало. Перші пристрої віртуальної реальності, як шоломи з датчиками руху і рукавички, що відстежують рухи рук, були розроблені тільки в 1960-х роках. Ці пристрої використовували переважно для військових тренувань і медичних досліджень.

У 1970-х роках з'явилися перші комп'ютерні ігри, що стали ще однією віхою у створенні віртуальних світів. Ці ігри давали змогу гравцеві взаємодіяти з віртуальним середовищем і керувати екранним персонажем.

У 1980-х роках поява потужніших персональних комп'ютерів та ігрових консолей дала змогу створювати віртуальні світи для широких мас. Одним із перших таких прикладів стала «*Ultima Underworld The Stygian Abyss*», що вийшла 1992 року [7].

Це була одна з перших ігор, у якій використовувалися текстури і 3D-графіка для створення захопливого віртуального світу.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 12 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |



Рис. 1.1 – Гра «Ultima Underworld The Stygian Abyss»

З появою потужніших технологій віртуальні світи в іграх стали більш реалістичними та деталізованими. Прикладом тому може слугувати «Crysis», що вийшла в 2007 році [7]. У цій грі гравцям належало досліджувати величезний тропічний острів, наповнений дикою природою і ворожими солдатами.

Завдяки передовому ігровому рушію *CryEngine* гра пропонувала приголомшливу графіку і фізику, а також інноваційні механізми взаємодії з навколишнім середовищем, такі як руйнування об'єктів і динамічна рослинність.

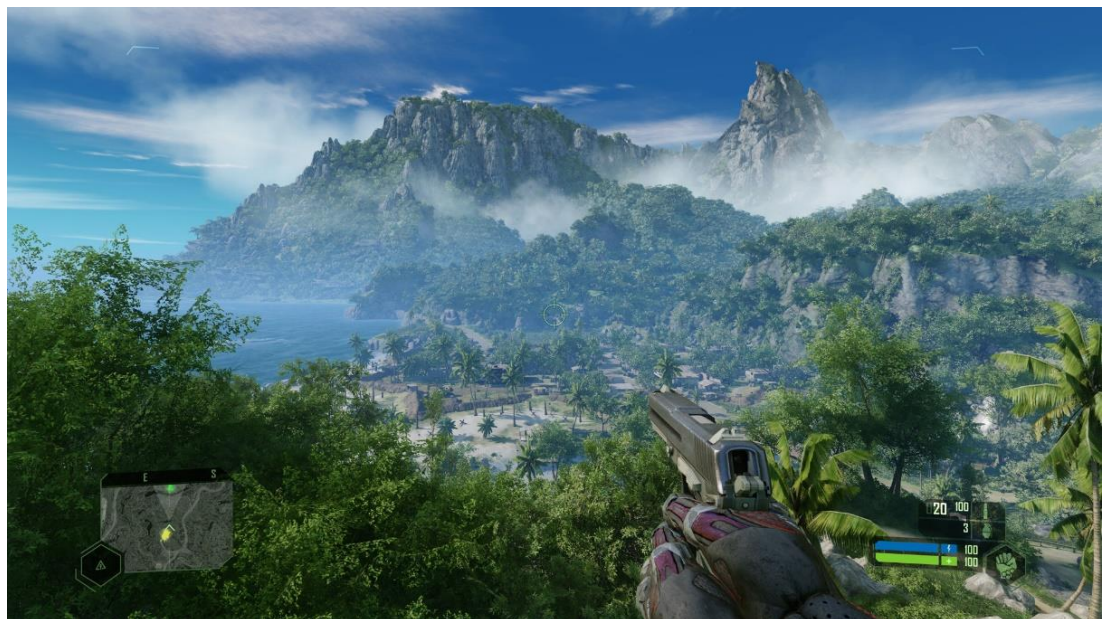


Рис. 1.2 – Гра «Crysis»

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дат | | 13 |

1.1.2 Класифікація віртуальних світів

Насправді віртуальні світи – це не тільки ігри та соціальні мережі. Вони належать до будь-якого згенерованого комп'ютером середовища, де користувачі можуть взаємодіяти з об'єктами та іншими користувачами. Серед неігрових прикладів – віртуальні музеї, навчальні симулятори, віртуальні тури містами та країнами, віртуальні виставки та віртуальні конференції.

Цікаво, що в майбутньому межі між реальним і віртуальним світом можуть стати ще більш розмитими. З розвитком таких технологій, як доповнена реальність (AR) і віртуальна реальність (VR), віртуальні світи все більше проникатимуть у наше повсякденне життя [24].

Якщо продовжувати розмірковувати на цю тему, то можна сказати, що віртуальний світ має безліч переваг перед реальним. Наприклад, вони пропонують можливості, які неможливі в реальності. У віртуальних світах люди можуть літати, мандрувати в часі та створювати будь-які предмети і світи, які тільки можна собі уявити.

Існують різні підходи до класифікації віртуальних світів. Один із них полягає в класифікації віртуальних світів за ступенем занурення користувача у віртуальне середовище [24]:

1. Неімерсійні віртуальні світи – користувач взаємодіє з віртуальним середовищем за допомогою стандартних пристроїв введення/виведення (клавіатура, миша, монітор). Приклади: комп'ютерні ігри, соціальні мережі, віртуальні виставки.

2. Напівімерсійні віртуальні світи – користувачі частково занурюються у віртуальне середовище за допомогою спеціальних пристроїв, таких як шоломи віртуальної реальності або 3D-окуляри. Приклади: симулятори віртуальної реальності, віртуальні тури.

3. Імерсійні віртуальні світи – користувач повністю занурюється у віртуальне середовище за допомогою спеціалізованих пристроїв, таких як рукавички для роботи з даними або костюми віртуальної реальності. Приклади: віртуальна реальність в ігровій індустрії; віртуальні лабораторії.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 14 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

1.1.3 Інструмент створення віртуального світу

Game engines – ігрові рушії являють собою комплекс програмного забезпечення, призначеного для створення комп'ютерних ігор та інших інтерактивних додатків. Вони надають широкий спектр інструментів для створення віртуального світу, включно з редакторами рівнів, системами анімації, фізичним рушієм тощо. Деякі з найпопулярніших ігрових рушіїв включають [24]:

1. *Unity* – це популярний ігровий рушії, який використовується для створення ігор та інших інтерактивних додатків для різних платформ, включно з ПК, консолями, мобільними пристроями та веб. *Unity* підтримує широкий спектр програмних мов, зокрема *C#*, *JavaScript* і *Python*.

2. *Unreal Engine* – це високопродуктивний ігровий рушії, який використовується для створення ігор та інших інтерактивних додатків для різних платформ, включно з ПК, консолями, мобільними пристроями та віртуальною реальністю. Підтримує програмну мову *C++*.

3. *CryEngine* – інший варіант ігрового рушія для створення ігор та інших інтерактивних додатків, доступний на різних платформах, включаючи ПК, консолі та віртуальну реальність. У своїй роботі *CryEngine* використовує мову програмування *C++*.

3D-моделювання – програми використовуються для створення *3D*-об'єктів і персонажів, які можна використовувати у віртуальних світах. Однією з найважливіших особливостей програм для *3D*-моделювання є можливість створення текстур, які можна накладати на *3D*-об'єкти для надання їм більш реалістичного вигляду. Текстури можуть включати кольори, візерунки та інші деталі, які допомагають створити правдоподібний віртуальний світ.

Ще одна важлива особливість програм для *3D*-моделювання – можливість створення анімації. Анімація може використовуватися для створення руху персонажів, транспортних засобів та інших об'єктів у віртуальному світі. Це

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дат | | 15 |

допомагає створити більш динамічне та інтерактивне середовище для користувача [7].

Програми для 3D-моделювання такі як:

1. *3dsMax* – популярна програма для 3D-моделювання, використовується для створення 3D-об'єктів і персонажів для ігор, фільмів та інших інтерактивних додатків. Вона пропонує потужні інструменти для моделювання, анімації та рендерингу.

2. *Maya* – професійна програма для моделювання, що використовується для створення об'єктів і персонажів для ігор, фільмів та інших інтерактивних речей. Широко використовується в кіноіндустрії та ігровій індустрії завдяки своїм потужним можливостям для анімації та симуляцій.

3. *Blender* – безкоштовна програма для моделювання, для створення об'єктів і персонажів для ігор, фільмів. Є популярним вибором для незалежних розробників завдяки своїй безкоштовності та потужним можливостям для моделювання, анімації та рендерингу.

Крім *3dsMax*, *Maya* і *Blender*, існує безліч інших програм для 3D-моделювання, таких як *Cinema4D*, *Modo* і *ZBrush* [15]. У кожній програмі є свої переваги та особливості, які роблять її більш придатною для певних типів проектів.

Загалом програми для 3D-моделювання відіграють важливу роль у створенні віртуальних світів і середовищ.

Віртуальна реальність – технології використовуються для створення повноцінного занурення у віртуальний світ. Вони надають користувачеві можливість взаємодіяти з об'єктами і персонажами у віртуальному світі за допомогою спеціальних контролерів і шоломів віртуальної реальності.

Веб-технології – такі як *HTML5* і *WebGL*, використовуються для створення віртуальних світів у браузері. Це дає змогу створювати віртуальні світи, доступ

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дат | | 16 |

до яких можна отримати з будь-якого пристрою, що має підключення до Інтернету.

Деякі з найпоширеніших інструментів, що використовуються для створення віртуальних світів за допомогою веб-технологій, такі як [7]:

1. *A-Frame* – безкоштовний фреймворк для створення віртуальної реальності з використанням *HTML5* і *WebGL*.

2. *Three.js* – безкоштовна бібліотека *JavaScript* для створення 3D-графіки в браузері.

Веб-технології також можна використовувати для створення віртуальних світів в іграх. Прикладом може слугувати гра *Agar.io*, що вийшла 2015 року. Це багатокористувацька онлайн-гра, в якій гравці керують клітинами і поглинають інші клітини, щоб рости і виживати. Гра побудована з використанням веб-технологій *HTML5* і *JavaScript*.

Інший приклад – гра "*Minecraft Classic*" [7], також побудована на веб-технологіях. *Minecraft Classic* – це безкоштовна версія оригінальної *Minecraft*, яка дає змогу користувачам створювати свій власний віртуальний світ із блоків. Гру було створено за допомогою мови програмування *Java*.

Веб-технології також використовуються для створення віртуальних турів і презентацій. Наприклад, технологія *WebVR* [24] може використовуватися для створення віртуальних турів музеями та виставками, які можна переглядати в браузері комп'ютера або смартфона.

Перевага використання веб-технологій для створення віртуальних світів полягає в тому, що користувачам не потрібно встановлювати додаткове програмне забезпечення або купувати дороге обладнання. Усе, що їм потрібно – це браузер і підключення до Інтернету.

Крім того, веб-технології дають змогу створювати кросплатформні додатки, що працюють на різних пристроях та операційних системах.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 17 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

1.1.4 Спосіб взаємодії

Навігація віртуальним світом дає змогу користувачеві переміщатися віртуальним світом, досліджувати його та взаємодіяти з об'єктами. Навігація може бути реалізована за допомогою різних пристроїв введення, таких як клавіатура, миша, джойстик, контролер руху тощо. Класифікація навігації може містити такі параметри, як тип пристрою введення, тип переміщення (піший, транспортний, телепортація тощо), рівень свободи переміщення (відкритий світ, лінійний шлях, обмежена зона тощо).

Взаємодія з об'єктами у віртуальному світі дає змогу користувачеві маніпулювати об'єктами у віртуальному світі, виконувати різні дії з ними, такі як захоплення, переміщення, обертання, зміна розміру і т.д.

Взаємодія з об'єктами може бути реалізована за допомогою різних пристроїв введення, таких як миша, клавіатура, контролер руху, шолом віртуальної реальності тощо.

Класифікація взаємодії з об'єктами може містити такі параметри, як тип пристрою введення, тип об'єкта (статичний, динамічний, інтерактивний тощо), тип дії (захоплення, переміщення, обертання тощо) [7-24].



Рис. 1.7 – Гра «Minecraft» відкриття дверей як спосіб взаємодії з віртуальним світом

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дат | | 18 |

Одним із прикладів використання віртуального світу у фільмах є фільм «Аватар» Джеймса Кемерона [25]. У цьому фільмі було створено цілий віртуальний світ Пандори, населений фантастичними істотами і рослинами. За допомогою технології захоплення руху актори могли взаємодіяти з віртуальними об'єктами і персонажами, що надавало фільму більшої реалістичності.



Рис. 1.9 – Кадри з фільму "Avatar"

Іншим прикладом використання віртуального світу у фільмах є фільм «Гравітація» Альфонсо Куарона [25]. У цьому фільмі було створено віртуальне середовище космосу, в якому відбувалася дія. За допомогою комп'ютерної графіки було створено реалістичні зображення космічних кораблів, астронавтів і космічного простору.

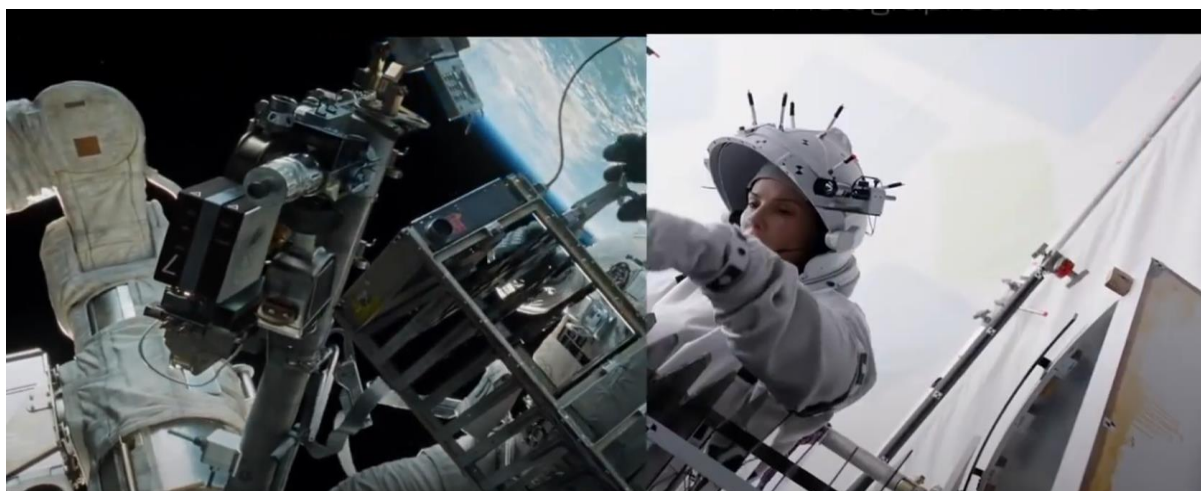


Рис. 1.10 — Кадри з фільму "Гравітація"

1.2.2 У іграх

Віртуальний світ в іграх використовується для створення ігрового процесу та оточення, у якому відбувається дія. Гравці можуть взаємодіяти з віртуальними об'єктами та персонажами, виконувати завдання та місії, досліджувати ігровий світ і розвивати свого персонажа. Одним із прикладів використання віртуального світу в іграх є гра «*World of Warcraft*». У цій грі створено цілий віртуальний світ Азерот, населений фантастичними істотами та персонажами. Гравці можуть обирати свою расу, клас і професію, виконувати завдання та місії, досліджувати ігровий світ і розвивати свого персонажа.



Рис. 1.11 – Карта Азерота



Рис. 1.12 – Пейзажі Азерота

Ще одним із найкращих прикладів віртуального світу є гра "*Red Dead Redemption 2*". У мешканців міста чи селища є свій графік роботи та відпочинку.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дат | | 21 |

Наприклад, власники магазинів відчиняють свої крамниці вранці та зачиняють увечері.

У диких тварин також є свої особливості поведінки протягом дня. Наприклад, хижаки, або вовки та пуми, полюють уночі, а травоїдні, олені та антилопи, пасуться вдень. У риб у річках і озерах теж є свої особливості поведінки.

Деякі види риб більш активні вдень, інші – вночі. У неігрових персонажів можуть бути свої щоденні маршрути пересування. Наприклад, стражники, які патрулюють місто, можуть слідувати за певним маршрутом, а мандрівники переміщатися між різними таборами. Погода також впливає на поведінку тварин і неігрових персонажів. У сиру погоду деякі тварини ховаються в укриттях, а неігрові персонажі захищаються від дощу.



Рис. 1.8 – Пейзажі гри "Red Dead Redemption 2"

Не менш важливо пам'ятати і про 2D-ігри. Як і 3D-віртуальні світи, вони являють собою штучно створені цифрові середовища, в яких користувачі можуть взаємодіяти один з одним і навколишнім середовищем у рамках певних правил і обмежень.

| | | | | | | |
|------|------|----------|--------|-----|-----------------------------|------|
| | | | | | <i>КРБ.КІ.1.442-03.1.10</i> | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дат | | 22 |

Прикладами 2D віртуальних світів є:

1. «*Stardew Valley*» – популярна відеогра, в якій гравці виконують різноманітні завдання, ловлять рибу і взагалі стежать за фермою.



Рис. 1.9 – Гемплей з гри " Stardew Valley "

2. «*Club Penguin*» – онлайн-гра, у якій гравці можуть створити власного пінгвіна, блукати віртуальним островом, грати в міні-ігри та взаємодіяти з іншими гравцями.

3. «*Terraria*» – гра-пісочниця, у якій гравці можуть будувати й ламати блоки, створюючи різноманітні конструкції, і досліджувати нескінченний процедурно генерований світ.



Рис. 1.10 – Гемплей з гри "Terraria"

4. «*Habbo*» – браузерна гра, в якій користувачі можуть створювати персонажів, оформляти віртуальні кімнати, спілкуватися і грати в міні-ігри з іншими користувачами.

1.2.3 Інше

Окрім ігор та фільм, віртуальний світ використовуються у [7-24]:

1. *Louvre VR* – віртуальна версія Лувру, де користувачі можуть досліджувати експонати і виставки, не виходячи з дому.

2. *Google Arts & Culture* – ця платформа надає доступ до тисяч музеїв і галерей по всьому світу, дозволяючи користувачам переглядати експонати у високій якості.

3. *AltspaceVR* – платформа для проведення віртуальних лекцій, семінарів і конференцій, яка дозволяє студентам і викладачам взаємодіяти у віртуальних класах.

4. *Ossor VR* – віртуальна платформа для навчання хірургів, яка дозволяє практикувати операції у безпечному і контрольованому середовищі.

5. *SimX* – медичний симулятор, який використовується для навчання студентів-медиків та медичних працівників, дозволяючи моделювати різні клінічні сценарії.

6. *VirBELA* – платформа, яка створює віртуальні офіси та кампуси, дозволяючи компаніям проводити зустрічі, тренінги та інші корпоративні заходи у віртуальному середовищі.

7. *Engage* – освітня платформа, яка дозволяє створювати та проводити інтерактивні уроки, тренінги та презентації у віртуальному середовищі.

8. *Virtual Reality Exposure Therapy (VRET)* – використання віртуальної реальності для психотерапії, де пацієнти можуть безпечно опрацьовувати свої страхи та тривоги у контрольованому середовищі.

9. *Spatial* – платформа для проведення віртуальних зустрічей та співпраці, що дозволяє користувачам спілкуватися та працювати разом у 3D-просторі.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 24 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

10. *Mozilla Hubs* – Відкрита платформа для створення та спільного використання віртуальних просторів для зустрічей, презентацій та творчої співпраці.

1.3 Постановка завдання

Завданням дипломної роботи є розробка сцени у рушії у 3D-всесвіту:

1. Загальні вимоги. Зробити віртуальний неімерсійний і неінтерактивний світ. Так само відтворити сцену меча в камені, при цьому відповідаючи критеріям відповідності стилю проекту. А також, зокрема, зробити меч у 3D-програмі.

2. Вимоги до об'єктів. Модель низькополігональна, роздільна здатність текстури та коректне накладення на об'єкт. Так само об'єкти сцени доповнюються з бібліотеки.

3. Вимоги до візуальних ефектів. Потрібно реалізувати ефекти туману, дощу, світіння.

4. Вимоги до використовуваних технологій. Застосувати програми для 3D-моделювання, текстурингу, ігровий рушій. Так само, зокрема використання технології системи *Niagara* для частинок. А, для туману технологію систему *Blueprint*.

5. Вимоги до результату. Результатом роботи має бути демонстраційна ролик у форматі mp4, тривалістю до 2 хв.

6. На основі заздалегідь розробленого плану проводять роботу з використанням аналізу різних алгоритмів, методів і заходів щодо реалізації. Проміжні результати зберігаються на додаткових носіях для подальшого використання.

Висновки до першого розділу

1. У результаті аналізу було досліджені віртуальні світи. Зокрема поняття класифікації, інструменти, способи взаємодії та приклади використання.

2. На підставі дослідження була зроблена постановка завдань.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 25 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

РОЗДІЛ 2 ПРОЕКТУВАННЯ

2.1 Розробка пайплайну

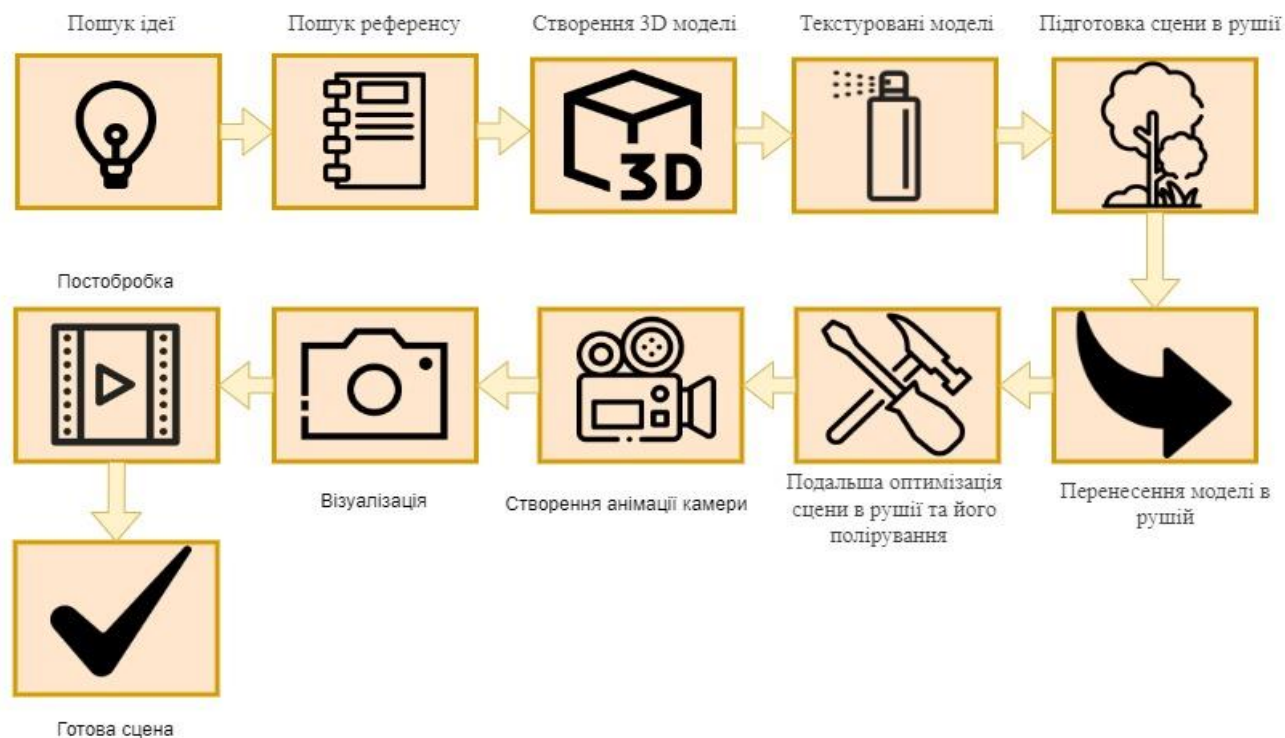


Рис. 2.1 – Пайплайн

1. Пошук ідеї – на цьому етапі визначається концепція і загальна тема майбутньої сцени. Це може включати роздум'я , обговорення ідей з командою та їх аналіз.

2. Референси – проводиться збір референсів і матеріалів, які допоможуть візуалізувати ідею. Це можуть бути фотографії, креслення, ілюстрації, що стануть основою для створення 3D-моделей.

3. Створення 3D-моделі – на основі зібраних референсів починається процес моделювання об'єктів у програмному забезпеченні. Цей етап включає створення геометрії, деталізацію і перевірку топології моделей.

4. Текстурування моделі – після створення моделі додаються текстури, щоб надати об'єктам реалістичного вигляду. Це включає створення карт

| | | | | | | |
|------|------|----------|--------|-----|-----------------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дат | | 26 |

нормалей, карт кольору, карт відбиття та інших типів текстур у спеціалізованих програмах.

5. Підготовка сцени в рушії – розробка і підготовка сцени у графічному рушії. На цьому етапі створюється оточення, розташовуються моделі, налаштовується освітлення та інші параметри сцени.

6. Перенесення моделі в рушій – експорт моделей з 3D-програмного забезпечення і імпорт їх у графічний рушій. Тут важливо стежити за коректністю перенесення текстур і матеріалів.

7. Подальша оптимізація сцени в рушії та його полірування – проводиться оптимізація сцени для забезпечення плавного рендерингу. Це може включати налаштування рівнів деталізації (*LOD*), зменшення кількості полігонів, оптимізацію текстур і матеріалів.

8. Створення анімації камери – на цьому етапі налаштовується анімація камери для створення руху в сцені. Це може включати налаштування траєкторії руху камери, швидкості і плавності переходів. Використовуються інструменти анімації камери у графічному рушії, щоб забезпечити бажаний ефект і кінематографічний вигляд сцени.

9. Візуалізація – Проводиться остаточний рендер сцени для створення зображень або відео. Важливо налаштувати параметри рендерингу, такі як якість зображення, роздільна здатність та інші налаштування.

10. Постобробка – після рендерингу сцена проходить постобробку, де додаються фінальні штрихи, такі як корекція кольору, додавання ефектів (наприклад, блиску, туману, глибини різкості) та інші покращення.

11. Готова сцена – завершальна стадія, коли сцена готова до показу або інтеграції у проект. Перевіряється якість, відповідність технічним вимогам і готовність до використання за призначенням.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 27 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

2.2 Проектування віртуального світу

2.2.1 Формування концепції

Під час створення 3D-моделі слід враховувати низку важливих аспектів, які визначають її якість, реалізм і функціональність. Першочергово, важливо мати чітке уявлення про те, що саме ви хочете створити, чи то архітектурний об'єкт, чи то предмет інтер'єру, персонаж для гри, чи то щось іще [10].

Наприклад, знання анатомії меча і його складових частин, таких як клинок, руків'я і гарда, допоможе створити модель, яка матиме переконливий і автентичний вигляд.

Крім того, при створенні 3D-моделі меча важливо враховувати, для яких цілей вона буде використовуватися. Якщо модель призначена для візуалізації в рамках кінематографа або відеоігор, то вона має бути оптимізована з урахуванням вимог до продуктивності та анімації.

Далі, при створенні сцени на рушії варто так само враховувати деталі

1. Мета – першим кроком має бути визначення цілі проекту і того, яку роль відіграватиме сцена в досягненні цих цілей. Наприклад, якщо це кінематографічна сцена, то вона має бути спроектована так, щоб важливі деталі залишалися в центрі екрана.

2. Естетика й атмосфера – візуальне оформлення сцени відіграє величезну роль у створенні атмосфери. Архітектурний дизайн, освітлення, текстури та об'єкти зобов'язані відповідати загальному настрою.

2.2.2 Розробка концепції меча

Цей меч, встромлений у камінь, має уособлювати таємничість і мудрість віків. Він не просто зброя, а й частина самої землі, частина душі лісу. Тому, цей меч має бути водночас простим і величним. Його лезо має бути вкрите «чистим металом» і вигравірувано деякі ієрогліфи, що світлитимуться, це надасть гарну таємничість. А просте рукоятка додасть йому мудрості.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 28 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

Яскравий приклад використання концепту меча в камені у фільмах-мультфільмах:

1. «Мерлін» (*Merlin*, 1998) – у цьому телефільмі меч, витягнутий із каменю, зображується як величний клинок із різьбленим руків'ям, оздобленим дорогоцінним камінням або символами магії. Лезо меча може бути прикрашене рунами або іншими містичними символами, що відображають його зв'язок із магичною силою.



Рис. 2.2 – Кадри с фільма «Мерлін»

2. «Король Артур» (*King Arthur*, 2004) – у цьому фільмі меч Екскалібур, витягнутий із каменя, зображується як могутній клинок із блискучим лезом, прикрашений складними гравіюванням і символами. Рукоять меча може бути обтягнута шкірою або прикрашена дорогоцінним камінням, а на гарді можуть бути вирізьблені символи стародавньої магії.

Приклади з ігор:

1. *The Legend of Zelda: Skyward Sword* – у цій грі меч, витягнутий з каменю, називається «Меч Зефіра». Він має витончений, майже ефірний вигляд, з гравіюванням на клинку, що нагадують хмари або вихори. Рукоять меча може бути прикрашена символами стародавньої магії або світяться кристалами, що відображають його зв'язок з духовними силами.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 29 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

2. *Fable* – у цій грі меч у камені може бути представлений у різних варіантах, залежно від того, який конкретно меч гравець знаходить. Це може бути як старовинний і шляхетний клинок із гравіюванням та візерунками, так і простий, але могутній меч із практичним дизайном і міцною конструкцією.

Загальний підсумок концепту меча в камені можна охарактеризувати як поєднання простоти і могутності. Незалежно від того, як його зображують у фільмах, мультфільмах або іграх, меч у камені завжди представлений як предмет простої, природної краси і форми, але при цьому він має величезну силу і значущість.

Візуально він може бути зображений з прямим лезом, простим руків'ям і без зайвих декоративних елементів, що підкреслює його надійність і практичність. Водночас, його могутня символіка виражається в його зв'язку з каменем, який є символом твердості, довговічності та сили.

2.2.3 Розробка концепції ландшафту

Ландшафт не тільки визначає візуальну естетику, а й відіграє ключову роль у посиленні наративу та емоційного впливу на аудиторію.

Одним із найпотужніших і найзнаковіших сюжетів, де ландшафт стає невід'ємним елементом оповіді, є сцена витягування меча з каменю.

Цей міфічний момент, що втілює ідеї обраності, магії та героїзму, вимагає від творців ретельного підходу до розроблення довкілля, щоб повністю передати його значення і магічну атмосферу.

Приклад ландшафту з фільмів, мультфільмів та ігор:

1. «Мерлін» (*Merlin*, 1998) – меч у камені розташований у містичному лісі, оточеному стародавніми деревами, вкритими мохом і лишайниками. Навколо панує чарівна атмосфера: повітря наповнене легким туманом, пробиваються промені сонця, і чуються тихі звуки природи. Камінь, з якого стирчить меч, стоїть на невеличкій галявині, Досвітлений м'яким світлом, що створює відчуття, що це місце священне та наповнене магією.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 30 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

2. «Меч у камені» (*The Sword in the Stone*, 1963) – меч у камені розташований на центральній площі невеликого середньовічного міста. Площа оточена кам'яними будівлями, критими черепицею, і брукованими вулицями. Навколо каменю зібрався натовп людей, здивованих і очікуючи побачити диво.



Рис. 2.3 – З мультфільму «Меч у камені»

3. *The Legend of Zelda: Skyward Sword* – меч Зефіра витягнуто з каменю на вершині високого острова, що ширяє в хмарах. Острів оточений небесними пейзажами: білі хмари, блакитне небо і мерехтливі світлом повітряні потоки. Навколо каменю ростуть чудернацькі рослини, що випромінюють світло, і чути мелодійний звук вітру, що надає місцю чарівного і духовного характеру.

2.2.4 Розробка концепції наповнення сцени

Створення захопливих і символічно насичених сцен у кіно та відеоіграх вимагає ретельного розроблення концепції наповненості сцени. Наповненість сцени охоплює організацію всіх візуальних, звукових і атмосферних елементів таким чином, щоб вони разом ефективно передавали бажані емоції, теми і смисли. Наповненість сцени включають [25]:

1. Декорації та об'єкти – розробка концепції починається з вибору і розташування всіх об'єктів у сцені, включно з декораціями, меблями, реквізитом і природними елементами. Ці об'єкти мають відповідати тематиці та часу дії сцени.

2. Освітлення – світло відіграє ключову роль у створенні атмосфери. Правильне освітлення може підкреслити настрій сцени, виділити важливі елементи і створити глибину кадру. У сцені виймання меча з каменю освітлення може бути м'яким і розсіяним, створюючи священну і містичну атмосферу.

3. Колірна палітра – вибір колірної гами допомагає встановити тон сцени. Холодні тони можуть створювати відчуття тривоги або містики, тоді як теплі тони – комфорт і безпеку.

Сюди можна додати аудіо-елементи (фонову музику і звукові ефекти), вони так само можуть підкреслити сцені настрій, залежно від того, як ми хочемо передати. Для наповненості сцени «Меча в камені» підійде спокійна музика, яка створить ефект містики.

2.2.5 Пошук референсів

Пошук референсів є важливим етапом у створенні високоякісних 3D моделей та віртуальних сцен. Цей процес починається з визначення вимог до референсів на основі ключових аспектів проекту, таких як архітектура, текстури, кольорова палітра і освітлення. Після цього проводиться систематичний збір візуальних матеріалів із різних джерел. Основними джерелами можуть бути спеціалізовані фотостоки, архітектурні та дизайнерські журнали, а також професійні портфоліо. Інтернет-ресурси, такі як *Pinterest*, *Google Images*, *Behance* та *ArtStation*, також широко використовуються для пошуку відповідних референсів.

Зібрані матеріали класифікуються та організовуються за категоріями, що забезпечує зручність їх подальшого використання. Для цього можуть застосовуватись інструменти для управління референсами, такі як *PureRef*, *Eagle* або *Adobe Bridge*.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 32 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |



Рис. 2.4 – 3D-модель меча як референс

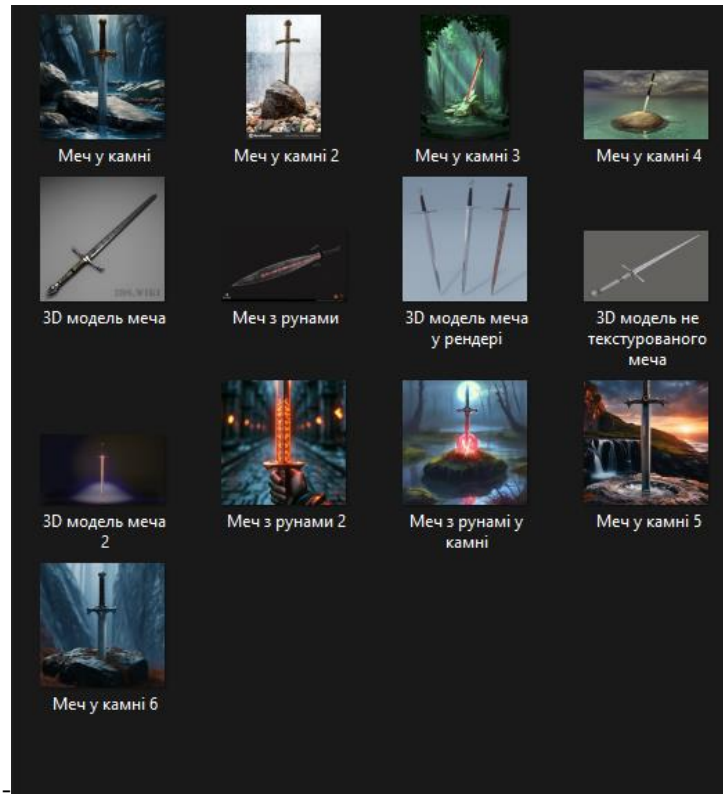


Рис. 2.5 – Референси

2.3 Огляд технології створення об'єктів

2.3.1 Технологія моделювання

3D-моделювання – це процес створення тривимірних об'єктів за допомогою спеціалізованого програмного забезпечення. Ці моделі можуть бути представлені у вигляді віртуальних об'єктів, які можна використовувати для візуалізації, аналізу, анімації, симуляції та багатьох інших цілей.

Для створення тривимірної графіки важливо мати розуміння основних принципів 3D-моделювання та володіння спеціалізованим програмним забезпеченням. Слід враховувати:

| | | | | | | |
|------|------|----------|--------|-----|-----------------------------|------|
| | | | | | <i>КРБ.КІ.1.442-03.1.10</i> | Арк. |
| | | | | | | 33 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

1. Декартову систему координат – позначаються як X, Y, Z , причому Z вісь перпендикулярна площині XY . У різних програмах орієнтація Z осі може бути різною.

2. Анатомія об'єкта або сцени – розуміння структури і форми об'єкта або сцени, яку ви хочете створити, є основоположним. Це може включати знання анатомії людського тіла, архітектурних принципів, механіки об'єкта тощо.

3. Розуміння топології та полігонального моделювання – важливо розуміти, як правильно структурувати та об'єднувати полігони (трикутники або чотирикутники), щоб забезпечити оптимальну роботу моделі надалі. Один із важливих аспектів – це розуміння топології, тобто розташування та з'єднання полігонів.

Формула для обчислення кількості полігонів:

$$P = F = \frac{2E-4}{2} \quad (2.1)$$

де: P – кількість полігонів;

E – кількість ребер;

F – кількість граней.

4. Освітлення та рендеринг – розуміння принципів освітлення та налаштування параметрів рендерингу дає змогу створювати якісні зображення та анімації. Важливо враховувати параметри освітлення, такі як інтенсивність, кут падіння світла та тіні, які суттєво впливають на кінцевий результат рендерингу.

Для обчислення інтенсивності світла на поверхні часто використовується рівняння освітленості [10]:

$$I = (L \cdot N) / r^2 \quad (2.2)$$

де: I – інтенсивність світла на поверхні;

L – вектор, що характеризує напрямок світла;

N – нормаль до поверхні в точці, де розраховується освітленість;

R – відстань між джерелом світла та точкою на поверхні.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 34 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

Для створення тривимірних об'єктів найпоширенішим способом є використання полігонального моделювання. Полігональне моделювання базується на побудові об'єктів із безлічі полігонів, кожен із яких характеризується вершинами, ребрами та гранями. Ці полігони утворюють полігональну сітку, яка може бути представлена у вигляді трикутників або прямокутників.

Воно надає гнучкість і ефективність при створенні різноманітних форм і поверхонь. В основі полігонального моделювання лежить ідея розбиття об'єкта на простіші елементи – полігони. Кожен полігон визначається своїми вершинами (точками в тривимірному просторі), ребрами (відрезками, що з'єднують вершини) і гранями (поверхнею, утвореною полігоном). Полігональна сітка складається з безлічі таких полігонів, утворюючи поверхню об'єкта [8].

Найчастіше в полігональному моделюванні використовують трикутники або чотирикутники (прямокутники), тому що вони легко обробляються комп'ютером і забезпечують хорошу апроксимацію поверхні. Після створення полігональної сітки об'єкта, її можна додатково модифікувати, додаючи деталі, змінюючи форму, текстуруючи тощо.

Крім полігонального моделювання, існує кілька інших методів створення тривимірних об'єктів:

1. Субдивізіонне моделювання – цей метод починається з базової геометрії (найчастіше простих форм, таких як куби або сфери) і потім застосовує операції субдивізії, щоб збільшити роздільну здатність і деталізацію моделі. Це дає змогу створювати плавні та органічні форми.

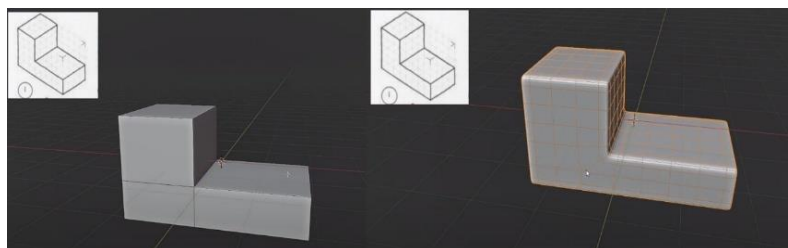


Рис. 2.6 – Субдивізіонне моделювання

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 35 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

можуть мати реалістичні деталі, такі як текстура шкіри, дерева або каменю, що надає їм характеру та індивідуальності.

Процес текстуровання складний і може бути визнаний мистецтвом. Він включає кілька основних етапів. Спочатку створюється *UV*-розгортка, яка передбачає розгортання *3D*-моделі на плоску поверхню для накладення текстури, що включає розбиття моделі на ділянки та їх розгортку на двовимірну площину (*UV*-координати).

Далі створюється сама текстура, яка може бути розроблена в графічних редакторах, таких як *Adobe Photoshop* або *GIMP*, або ж генеруватися процедурно. Ці текстури можуть включати різні карти, такі як дифузна (колірна), нормальна (рельєф), спекулярна (блиск) та інші.

На наступному етапі текстури накладаються на *3D*-модель з використанням *UV*-координат, при цьому важливо забезпечити, щоб текстура правильно відповідала поверхні моделі, не створюючи розтягувань або швів. Для створення вражаючих ефектів і деталей використовуються техніки, такі як відображення нерівностей (*bump mapping*), нормалізація (*normal mapping*) та відображення (*reflection mapping*). Наприклад, текстура старовинної цегляної стіни виглядає зовсім інакше, ніж текстура сучасної металевої поверхні. Розуміння матеріалу та його властивостей допомагає створювати автентичні текстури, що відображають реальність.

Однією з основних складнощів текстуровання є баланс між деталізацією та продуктивністю. Більше деталей означає більше текстур і ресурсів, що ускладнює процес створення та рендерингу. Важливо забезпечити оптимальну продуктивність без шкоди якості за рахунок грамотного використання технік оптимізації текстур.

Узгодженість стилів і текстур протягом усього проекту також є важливим аспектом, оскільки текстури мають доповнювати інші візуальні елементи, такі як освітлення, композиція і колірна палітра, щоб створити єдину атмосферу або настрій.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 37 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

Основні технології текстурування:

1. Фотографічне текстурування – один із найпоширеніших методів, коли текстури запозичуються з фотографій реальних поверхонь і застосовуються до об'єктів. Цей процес може бути ручним або автоматизованим за допомогою програмного забезпечення для обробки зображень.

2. Процедурне текстурування – текстури генеруються алгоритмічно, а не з реальних фотографій. Це дозволяє створювати текстури з високим ступенем деталізації та контролю над параметрами, що особливо корисно в комп'ютерній графіці.

3. Текстурування з використанням штучного інтелекту – нові підходи до текстурування включають використання машинного навчання і штучного інтелекту для автоматичного створення і редагування текстур.

Типи текстурних карт:

1. Дифузна карта (*Diffuse map, Base Color*) – відповідає за основний колір і малюнок поверхні об'єкта.

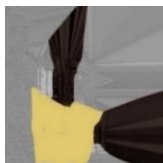


Рис. 2.9 – *Diffuse map*

2. Спекулярна карта (*Specular map*) – визначає, які ділянки поверхні будуть блищати і відбивати світло.



Рис. 2.10 – *Specular map*

3. Карта нормалей (*Normal map*) – використовується для створення ілюзії дрібних деталей на поверхні, як опуклості та заглиблення, без збільшення кількості полігонів.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 38 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |



Рис. 2.11 – *Normal map*

4. Карта висот (*Height map*) – схожа на карту нормалей, але замість нормалей використовує висоти для створення рельєфу поверхні.

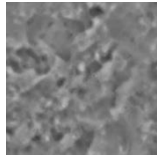


Рис. 2.12 – *Height map*

5. Карта відображень (*Reflection map*) – визначає, як поверхня об'єкта відображатиме навколишнє середовище.

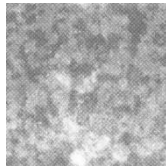


Рис. 2.13 – *Reflection map*

6. Карта шорсткості (*Roughness map*) – визначає ступінь розкиду відбитого світла, впливаючи на те, наскільки матовою або глянсовою буде поверхня.

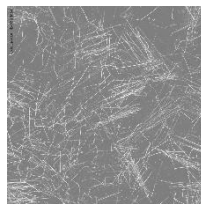


Рис. 2.14 – *Roughness map*

2.4 Технологія створення віртуальних світів у *Unreal Engine*

2.4.1 Можливості *Unreal Engine*

Unreal Engine пропонує безмежні можливості для створення віртуальних світів. Ця потужна платформа дозволяє реалізувати найсміливіші ідеї,

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 39 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

перетворюючи їх на захопливий віртуальний досвід. За допомогою *Unreal Engine* ми можемо створювати світи, які переносять користувачів у будь-який час і місце, забезпечуючи унікальну взаємодію з оточенням.

Процес створення віртуальних світів в *Unreal Engine* – це складний, багатоступеневий процес, що включає використання різноманітних інструментів та технологій.

Він має власне інтегроване середовище розроблення (*Unreal Editor*), що дає змогу розробникам створювати, редагувати та налагоджувати віртуальні світи. *Unreal Editor* надає інтуїтивно зрозумілий інтерфейс та безліч інструментів для роботи зі сценами, об'єктами, матеріалами, анімацією та багатьом іншим.

Прикладом, є *Landscape Tool* – інструмент ландшафту дозволяє розробникам створювати та редагувати терейни.

Кнопки та функції цього інструмента включають:

1. *Create Landscape* (Створити ландшафт) – дозволяє створити новий терейн та вибрати його розмір та роздільну здатність.

2. *Edit Landscape* (Редагувати ландшафт) – надає доступ до інструментів редагування терейну, таких як підняття, опускання, вирівнювання та інші.

3. *Paint Landscape* (Фарбувати ландшафт) – дозволяє розробникам застосовувати текстури та матеріали на терейн, створюючи різноманітність у вигляді.

4. *Sculpt* (Ліплення) – забезпечує інструменти для моделювання форми терейну, такі як підняття, опускання, різання та різноманітні пензлики для створення різних типів рельєфу.

5. *Smooth* (Згладжування) – дозволяє згладжувати нерівності та переходи між областями терейну, роблячи їх більш плавними та природними.

Також, є *Foliage Tool* (Інструмент рослинності) – цей інструмент використовується для додавання рослин та рослинності на ландшафт.

Кнопки та функції включають:

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 40 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

1. *Paint* (Фарбувати) – дозволяє вибирати рослини з бібліотеки та розміщувати їх на терейні.

2. *Density* (Щільність) – контролює кількість рослин, які розміщуються на певній площі терейну.

3. *Variety* (Різноманіття) – дозволяє налаштувати різноманіття рослин, які розміщуються, та їх розміри.

4. *Landscape Material* (Матеріал ландшафту) – цей інструмент дозволяє налаштовувати текстури та матеріали, які застосовуються до терейну. Кнопки та функції включають.

5. *Edit Layers* (Редагувати шари) – дозволяє додавати, видаляти та редагувати шари текстур та матеріалів на терейні.

6. *Paint* (Фарбувати) – дозволяє застосовувати текстури та матеріали на терейн, використовуючи різні режими редагування, такі як пензлі, градієнти та шаблони.

Для водної середі є *Water Body* (Водне тіло):

1. *Create Water Body* (Створити водне тіло) – ця опція дозволяє розробникам створювати озера, річки або інші водні форми на терейні.

2. *Edit Water Body* (Редагувати водне тіло) – після створення водного тіла розробники можуть використовувати цю опцію для редагування його форми, розмірів та властивостей.

Освітлення у *Unreal Engine* – це також ключовий аспект для створення реалістичних та захоплюючих віртуальних світів. Рушії надає розробникам різноманітні інструменти для створення та налаштування освітлення, які допомагають досягти потрібного настрою та атмосфери в їхніх проектах.

Приклад освітлення:

1. *Directional Light* (Напряме світло) – пряме світло відтворює ефект сонячного світла, яке поширюється в одному напрямку. Воно використовується для моделювання денного світла та встановлення загальної освітленості сцени.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 41 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

2. *Point Light* (Точкове світло) – точкове світло випромінює світло у всі напрямки від певної точки у просторі. Воно використовується для створення джерел світла, таких як ліхтарі, ліхтарі, світильники тощо.

3. *Spotlight* (Прожектор) – випромінює світло в конкретному напрямку з певним кутом розсіювання. Це ідеальний інструмент для створення прямих пучків світла, таких як світло з ліхтарів або прожекторів.

4. *Sky Light* (Небесне світло) – небесне світло моделює розсіяне світло від небесного простору, що допомагає створити реалістичне освітлення сцени, особливо під час сутінків та ранкового чи вечірнього світла.

5. *Exponential Height Fog* (Експоненційна висотна туманність) – цей інструмент створює ефект туману в залежності від висоти, що дозволяє створювати віддаленість та атмосферність в сцені.

6. *Post-Processing Effects* (Пост-процесінгові ефекти) – також надає широкий вибір пост-процесінгових ефектів, таких як блюр, кольорові корекції, *HDR (High Dynamic Range)* і багато інших, які дозволяють досягти потрібного візуального ефекту.

Також, інноваційна система «*Lumenosity*» система глобального освітлення в *Unreal Engine*, яка дозволяє автоматично розраховувати та відтворювати глобальне освітлення в реальному часі. Вона враховує різноманітні ефекти, такі як пряме світло, відбиття, розсіювання та поглинання, щоб створити реалістичні та привабливі освітлення в сцені.

Варто згадати і про «*Godray*» створює вражаючі "промені сонця", які проникають через об'єкти та створюють візуально захоплюючі ефекти. Ці промені найчастіше спостерігаються утворюються за рахунок розсіювання світла через атмосферу або прозорі об'єкти, що створюють враження м'якого та містичного світла.

Також є *Unreal Engine Marketplace* – внутрішній магазин асетів *Unreal Engine*, що надає розробникам доступ до широкого асортименту ресурсів,

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 42 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

інструментів та плагінів для прискорення розробки ігор та інших інтерактивних додатків.

Основні категорії асетів:

1. 3D-моделі та текстури – *marketplace* пропонує різноманітні 3D-моделі, включаючи персонажів, оточення, транспортні засоби, зброю та архітектурні елементи. Також доступні високоякісні текстури та матеріали для створення реалістичних поверхонь та візуальних ефектів.

2. Анімації – готові анімації для персонажів, тварин та об'єктів, включаючи ходьбу, біг, стрибки, бойові дії та інші рухи, що суттєво скорочують час розробки.

3. Звукові ефекти та музика – широкий асортимент звукових ефектів та музичних треків, що додають атмосферу та аудіальну глибину проектам. Це можуть бути звуки оточення, ефекти зброї, кроків та фонові музичні треки різних жанрів.

4. Інструменти та плагіни – інструменти та плагіни, які розширюють функціональні можливості *Unreal Engine*. Включають засоби для створення ландшафтів, системи керування персонажами, *AI*, мережеві плагіни тощо. Ці інструменти допомагають розробникам швидше та ефективніше реалізовувати складні ігрові механіки.

5. Блюпринти та кодові проекти – готові блюпринти та кодові проекти, що можуть бути адаптовані під конкретні потреби. Включають шаблони ігор, механіки персонажів, системи інвентарю та інші рішення, які можуть бути використані як основа для власних проектів.

2.4.2 Використання системи частинок

Система *Niagara* в *Unreal Engine* є результатом багаторічної розробки та вдосконалення, створена для забезпечення гнучкості та потужності в створенні складних візуальних ефектів у реальному часі. Її історія почалася з *Unreal Engine 3*, де початкові спроби створення ефектів за допомогою частинок були

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 43 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

обмеженими. З виходом *Unreal Engine 4* у 2014 році команда розробників активно працювала над новою системою частинок, яка відповідала б сучасним потребам індустрії відеоігор та візуальних ефектів у кіноіндустрії. Після випуску в *Unreal Engine 4.20*, *Niagara* отримала значну увагу завдяки новим функціям та покращенням.

Одним з ключових аспектів розвитку *Niagara* була підтримка реалістичних фізичних ефектів, включаючи розсіяння світла, взаємодію частинок з середовищем та симуляцію флюїдів. Це дозволило створювати більш реалістичні та захопливі візуальні ефекти.

Крім того, було впроваджено значні покращення продуктивності та оптимізації, що сприяло підвищенню ефективності розробки.

Система *Niagara* включає в себе:

1. Емітери (*Emitters*) – є базовими компонентами системи *Niagara*. Вони відповідають за вироблення та управління потоком частинок. Кожен емітер визначає початкові параметри частинок, такі як швидкість, кольори, розмір, напрямок та життєвий цикл. Розробники можуть створювати один або кілька емітерів в рамках одного *Niagara* ефекту.

2. Модулі (*Modules*) – впливають на різні аспекти частинок, дозволяючи контролювати їхні властивості та поведінку. Вони використовуються для зміни руху, зміни вигляду, інтеракції з оточуючим середовищем, анімації та багато іншого. Наприклад, модуль "*Drag*" може впливати на швидкість руху частинок у повітрі, а модуль "*Color Over Life*" може змінювати колір частинок протягом їхнього життєвого циклу.

3. Вираження (*Expressions*) – це скриптові конструкції, які дозволяють розробникам виконувати розрахунки та логіку для налаштування поведінки частинок. Вони можуть використовувати математичні операції, функції та змінні для керування параметрами частинок. Можуть бути використані для створення різноманітних ефектів, таких як керування рухом, зміна параметрів в залежності від часу або взаємодія з іншими об'єктами в сцені.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 44 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

4. Модулі ініціалізації (*Initialization Modules*) – ці модулі визначають початкові параметри для частинок при їхньому створенні. Вони можуть встановлювати початкові значення для швидкості, позиції, кутів руху та інших параметрів.

5. Модулі оновлення (*Update Modules*) – відповідають за оновлення стану частинок кожного кадру. Вони можуть змінювати позицію, швидкість, кольори та інші параметри частинок на кожному кадрі в залежності від заданої логіки.

6. Модулі завершення (*Completion Modules*) – ці модулі визначають дії, які виконуються при завершенні життєвого циклу частинки. Наприклад, вони можуть викликати вибух або зміну кольору частинки при її знищенні.

7. Конструктори частинок (*Particle Spawn Scripts*) – ці скрипти визначають спосіб створення нових частинок в системі. Вони можуть використовувати різні критерії, такі як час, положення, швидкість та інші параметри, для генерації нових частинок.

2.4.3 Використання блюпринтів

Історія *Blueprints* в *Unreal Engine* бере свій початок ще з випуску *Unreal Engine 4* у 2014 році. *Blueprints* – це система візуального програмування, яка дозволяє розробникам створювати ігрові логіку та функціонал без необхідності вмінь програмування на мові C++.

У своїй суті, *Blueprints* представляють собою графічний інтерфейс, що складається з блоків, які можна з'єднувати між собою для створення складних алгоритмів та логіки гри. Ця система спрощує розробку, роблячи процес більш доступним для творчих людей, які можуть не мати досвіду програмування.

Робота з *Blueprints* полягає в розміщенні різних блоків, які відповідають за різні операції, на панелях та з'єднанні їх між собою. Блоки представляють собою різні елементи гри, такі як персонажі, об'єкти, події, функції та умови. Розміщення та з'єднання цих блоків визначають поведінку гри.

Однією з ключових переваг *Blueprints* є їхня візуальна природа, що дозволяє розробникам легко візуалізувати та розуміти логіку програми. Це

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 45 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

особливо корисно для команд, які мають графічних дизайнерів та аніматорів, які можуть бути менш знайомі з програмуванням.

Що стосується технологій *Blueprints*, то вони базуються на системі подій та дій. Кожен блок містить певну дію, яка відбувається відповідно до подій у грі. Наприклад, коли персонаж зіштовхується зі стіною, спрацьовує подія "зіткнення", і виконується певна дія, яка може бути задана у вигляді блоків.

Назва блоків *Blueprints*:

1. *Event Graph* (Граф подій) – основна частина *Blueprints*, де ви розміщуєте блоки для обробки подій, таких як натискання кнопок, зіткнення об'єктів чи активація тригерів.

2. *Construction Script* (Скрипт конструктора) – використовується для налаштування вигляду та властивостей об'єктів при їхньому створенні або розміщенні на сцені.

3. *Function Library* (Бібліотека функцій) – функціональна бібліотека дозволяє вам створювати власні функції, які можна використовувати багаторазово в різних частинах вашого проекту.

4. *Macros* (Макроси) – дозволяють вам створювати короткі блоки коду, які можна використовувати для швидкого виконання певних операцій.

5. *Variables* (Змінні) – змінні дозволяють зберігати та керувати даними в коді, такими як числа, рядки, об'єкти тощо.

6. *Flow Control* (Керування потоком) – це блоки, які дозволяють вам керувати послідовністю виконання коду, такі як умовні оператори (*if, else*) та цикли (*for, while*).

7. *Debugging Tools* (Інструментів налагодження) – інструменти налагодження допомагають відстежувати та виправляти помилки в коді, дозволяючи вам докладно аналізувати виконання програми.

8. *Class Defaults* (Класові налаштування) – ці налаштування дозволяють вам визначити значення за замовчуванням для властивостей класу.

9. *Blueprint Interfaces* (Інтерфейси *Blueprints*) – це спосіб визначити набір функцій, які мають бути реалізовані в класах, що використовують цей інтерфейс.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 46 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

10. *Custom Events* (Користувацькі події) – користувацькі події дозволяють вам визначати власні події, які можна викликати з будь-якого місця вашого *Blueprint*.

Ці інструменти утворюють основу для створення логіки та функціональності у вашому проєкті з використанням системи візуального програмування *Blueprints*. Це основний перелік інструментів системи *Blueprints*. Важливо пам'ятати, що так чи інакше воно полегшує роботу. Але, блюпринти все одно потрібно вчити. Щоб відповісти на запитання, як правильно їх розставляти і з'єднувати між собою.

Висновки до другого розділу

1. Розроблено пайплайн и описано порядок та склад робіт.
2. Розглянуті технології створення об'єктів.
3. Розглянуті технології створення віртуальних світів у *Unreal Engine*.
4. Була спроектована система для створення віртуального світу.

| | | | | | | |
|------|------|----------|--------|-----|-----------------------------|------|
| | | | | | <i>КРБ.КІ.1.442-03.1.10</i> | Арк. |
| | | | | | | 47 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОЕКТУ

3.1 Вибір програмного забезпечення

3.1.1 Інструмент моделювання

У сучасному цифровому світі 3D-моделювання здійснює революцію в таких галузях, як розроблення ігор, відео, архітектурне проектування та медичне моделювання. Візуалізуючи об'єкти в тривимірному просторі, 3D-моделювання стало необхідним інструментом для багатьох професій, оскільки дозволяє досліджувати та взаємодіяти з об'єктами. Вибір правильного програмного забезпечення відіграє вирішальну роль у цьому процесі.

Існує безліч програм для 3D-моделювання, таких як *Blender*, *Autodesk Maya*, *3ds Max*, *ZBrush* та інші. Вибір залежить від уподобань, цілей і рівня досвіду. Для створення моделей і асетів для *Unreal Engine* я надаю перевагу *Blender* з кількох причин. *Blender* має потужний набір інструментів для моделювання, анімації та рендерингу, безкоштовний, і має активну спільноту користувачів, що забезпечує постійний розвиток і підтримку. Його гнучкість і можливість інтеграції з *Unreal Engine* роблять його ідеальним вибором для моїх потреб.

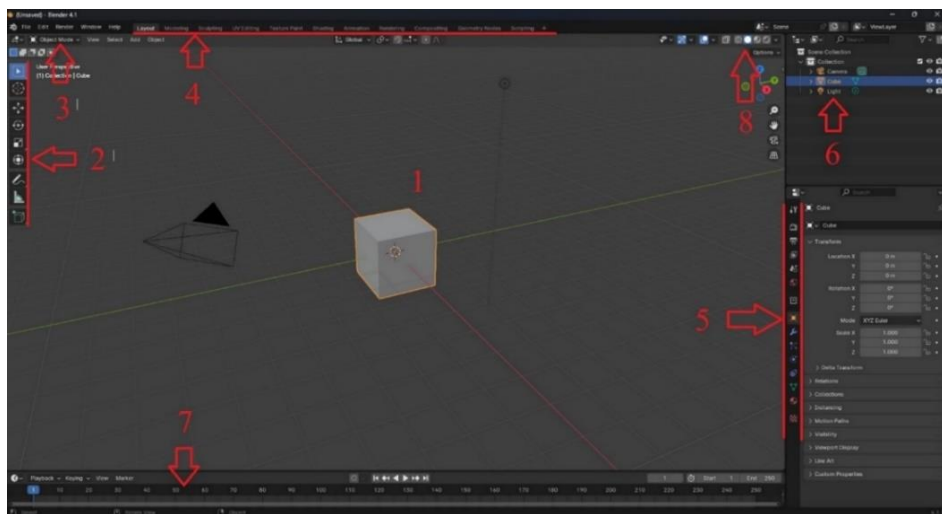


Рис. 3.1 – Інтерфейс програми *Blender* версії 4.1.1

| | | | | |
|------|------|----------|--------|-----|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дат |

КРБ.КІ.1.442-03.1.10

Арк.

48

1. Робоче вікно програми – сцена.
2. Панель інструментів переміщення.
3. Перемикання режимів вибором об'єкта і його редагуванням.
4. Перемикання редакторів від скульптинга до UV-розгортки, рендеринга і шейдинга.
5. Панель різних інструментів від тонкого редагування за допомогою чисел до модифікаторів, що дають змогу спростити роботу.
6. Позначення об'єктів, які розташовані на сцені.
7. Таймлайн – Там ставляться «ключі» для анімації.
8. Перемикання між видами, що дає змогу побачити грані та рендером.

Blender – це програмне забезпечення для тривимірної комп'ютерної графіки, створене нідерландським програмістом Тонном Розендалом у середині 90-х років. Перша версія була представлена у 1994 році під назвою «*Blender Creator*». У 2000 році компанія *NaN (Not a Number)* придбала права на програму та розпочала її розповсюдження під відкритою ліцензією. Після фінансових труднощів *NaN*, спільнота користувачів зібрала кошти для викупу програми, що призвело до створення некомерційної організації «*Blender Foundation*» та переходу програми під відкриту ліцензію *GPL*.

Blender став потужним інструментом для створення анімацій та тривимірних моделей, відзначаючись багатою функціональністю та активною спільнотою. Серед технологій, які він пропонує, виділяються розширені можливості для моделювання, включаючи полігональне, криві та скульптуру. Для текстурування і матеріалів використовуються сучасні методи *PBR (Physically Based Rendering)*, що забезпечують реалістичність зображень.

Анімаційні можливості включають складні системи кісток, інверсну кінематику та інструменти для симуляцій, таких як динаміка рідин, газів, твердих тіл та тканин. Візуалізація підтримує трасування променів в реальному часі завдяки *Cycles* і *Eevee*, забезпечуючи високоякісні результати як для статичних зображень, так і для анімацій.

| | | | | | | |
|------|------|----------|--------|-----|-----------------------------|------|
| | | | | | <i>КРБ.КІ.1.442-03.1.10</i> | Арк. |
| | | | | | | 49 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

Blender також інтегрує інструменти для відеомонтажу, композитингу та трекінгу руху, що робить його універсальним рішенням для постпродукції. Завдяки його безкоштовній природі, він знаходить застосування в різних галузях, включаючи відеоігри, архітектуру, медичну візуалізацію та кінематографію.

Особливості програмного забезпечення: *3D*-моделювання; *3D*-анімація; *3D*-рендеринг і візуалізація.

Підтримка мов: англійська, німецька, французька, італійська та інші.

Розробник: «*Blender Foundation*».

Рік випуску першої версії: 1994.

Таблиця 3.1

Системні вимоги *Blender 4.1.1*

| Компонет | Мінімальні вимоги | Рекомендовані вимоги |
|--------------------|---|--|
| Оперативна пам'ять | Від 8 ГБ <i>RAM</i> | Рекомендується: 16 ГБ <i>RAM</i> або більше |
| Відеокарта | Мінімум: Сумісна з <i>OpenGL 3.3</i> | Рекомендується: <i>NVIDIA GeForce GTX 1660 Ti</i> або <i>AMD Radeon RX 5000 Series</i> і вище |
| Сховище | Мінімум: 10 ГБ вільного простору на жорсткому диску | Рекомендується: <i>SSD</i> з високою швидкістю для прискорення завантаження файлів та проектів |
| Операційна система | <i>Windows 10, macOS 10.13, Linux</i> | <i>Windows 10, macOS 10.15, Linux</i> (останні версії <i>Ubuntu</i> і далі) |

3.1.2 Інструмент текстурування

У сучасному цифровому світі текстурування здійснює революцію в таких галузях, як розроблення ігор, відео, архітектурне проектування та медичне

моделювання. Створюючи високоякісні текстури для тривимірних моделей, текстурування стало необхідним інструментом для багатьох професій, оскільки дозволяє надавати об'єктам реалістичний вигляд та деталі. Вибір правильного програмного забезпечення відіграє вирішальну роль у цьому процесі.

Існує безліч програм для текстурування, таких як *Substance Painter*, *Mari*, *3D Coat*, *ArmorPaint* та інші. Вибір залежить від уподобань, цілей і рівня досвіду. Для створення високоякісних текстур я надаю перевагу *Substance Painter* з кількох причин. *Substance Painter* має потужний набір інструментів для текстурування, підтримку фізично коректного рендерингу (*PBR*) та інтеграцію з іншими програмами, такими як *Unreal Engine*. Його функціональність включає роботу з шарами, процедурні матеріали та текстури, а також підтримку роботи з високою роздільною здатністю, що забезпечує професійний результат.

Безкоштовні матеріали, доступні для користувачів, та активна спільнота забезпечують постійний розвиток і підтримку. Гнучкість *Substance Painter* та можливість інтеграції з іншими *3D*-пакетами роблять його ідеальним вибором для моїх потреб.

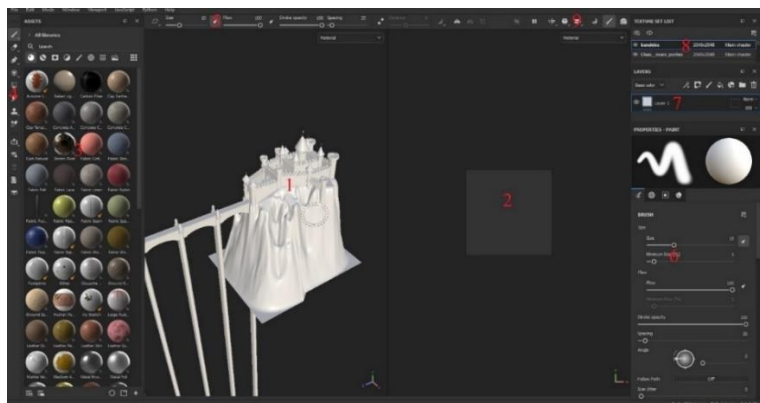


Рис. 3.2 – Інтерфейс програми *Substance Painter*

Substance Painter – потужний інструмент для текстурування та редагування *3D*-моделей, розроблений *Allegorithmic* і тепер входить до складу *Adobe Substance Suite*. Це програмне забезпечення дозволяє художникам створювати високоякісні текстури з неймовірним рівнем деталізації та реалізму. Користувачі можуть швидко додавати матеріали, знос, подряпини та інші деталі до своїх моделей.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 51 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

Особливості:

1. Інтуїтивний інтерфейс – простий у використанні інтерфейс полегшує навігацію та робить процес текстуровання ефективнішим.
2. Підтримка *PBR-workflow* – текстури реагують на світло і тіні, як реальні матеріали, забезпечуючи високий ступінь реалізму.
3. Потужні інструменти малювання – широкий вибір пензлів, масок і фільтрів для створення унікальних текстур.
4. *Smart Materials* і *Smart Masks* – автоматизоване застосування текстур та ефектів, що прискорює робочий процес.
5. Інтеграція з іншими додатками – *Live Link* з *Blender*, *Maya*, *3ds Max* та іншими забезпечує плавний робочий процес.

Порівняно з іншими програмами для текстуровання, *Substance Painter* виділяється своєю користувацькою дружністю, багатим набором інструментів і підтримкою індустріальних стандартів. Це програмне забезпечення вимагає потужних комп'ютерних ресурсів для оптимальної роботи: На менш потужних системах можливі обмеження в продуктивності та функціональності.

Таблиця 3.2

Системні вимоги *Substance Painter*

| Компонет | Мінімальні вимоги | Рекомендовані вимоги |
|--------------------|--|---|
| Оперативна пам'ять | Від 8 ГБ <i>RAM</i> | Рекомендується: 16 ГБ <i>RAM</i> або більше |
| Відеокарта | Мінімум: 2 ГБ <i>VRAM</i> , <i>DirectX</i> 11 сумісна | Рекомендується: 6 ГБ <i>VRAM</i> , <i>DirectX</i> 12 сумісна |

| Компонет | Мінімальні вимоги | Рекомендовані вимоги |
|--------------------|---|--|
| Сховище | Мінімум: 2 ГБ вільного місця на <i>SSD</i> | Рекомендовано: <i>SSD</i> з високою швидкістю для прискорення завантаження і роботи з текстурами |
| Операційна система | <i>Windows 7 / 8 / 10</i> (64-біт), <i>macOS 10.13</i> и вище, <i>Linux</i> | <i>Windows 10</i> (64-біт), <i>macOS 10.15</i> i вище, <i>Linux</i> (підтримувані дистрибутиви) |

Існує безліч програмних пакетів для текстурзування, які пропонують широкий спектр можливостей та інструментів для художників і дизайнерів. Ось деякі з найпопулярніших програм для текстурзування:

1. *Mari* – це інший популярний інструмент для текстурзування, часто використовуваний в кіноіндустрії та візуальних ефектах. Він спеціалізується на роботі з великими текстурами та складними проектами, такими як створення текстур для персонажів та оточення відразу.

2. *Quixel Mixer* – це безкоштовна програма для створення текстур, яка пропонує широкий вибір матеріалів та текстур. Вона має простий інтерфейс та інтуїтивно зрозумілі інструменти для швидкого та ефективного текстурзування. *Quixel Mixer* також інтегрується з іншими продуктами *Quixel*, що дозволяє легко імпортувати текстури та матеріали для подальшої обробки.

3.2 Моделювання меча

Створення 3D-моделі меча передбачає використання різних технік та інструментів, як примітиви, модифікатори та ручне редагування, щоб досягти високого рівня деталізації та реалізму.

| | | | | | | |
|------|------|----------|--------|-----|-----------------------------|------|
| | | | | | <i>КРБ.КІ.1.442-03.1.10</i> | Арк. |
| | | | | | | 53 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

Ми можемо виокремити кілька ключових етапів, з яких складається меч, як клинок, гарда, руків'я та навершя. Кожен із цих компонентів потребує особливої уваги і застосування різних методик моделювання, щоб досягти найкращого результату.

Ми розглянемо такі аспекти:

1. Клинок – процес створення клинка, включно з використанням примітивів, екструдуванням і масштабуванням, а також застосуванням модифікаторів для чіткості граней і поліпшення геометрії.

2. Гарда – техніки створення складних симетричних форм, використання модифікаторів «*Mirror*» і «*Boolean*» для створення складних переходів і вигинів.

3. Рукоть – підходи до моделювання рукоті з потовщеннями і вигинами, застосування модифікатора «*Subdivision Surface*» для згладжування і надання органічності формам.

4. Навершя – деталізація навершя з використанням ручного редагування.

5. Загальна геометрія і сітка – оптимізація моделі для використання в іграх і анімації, симетрія моделі із застосуванням модифікатора «*Mirror*», а також використання модифікаторів «*Subdivision Surface*» для поліпшення зовнішнього вигляду моделі.

Під час більш докладного опису, ми розберемо кожен деталь створення, а також будуть надані рекомендації щодо інструментів, які були використані для створення меча.

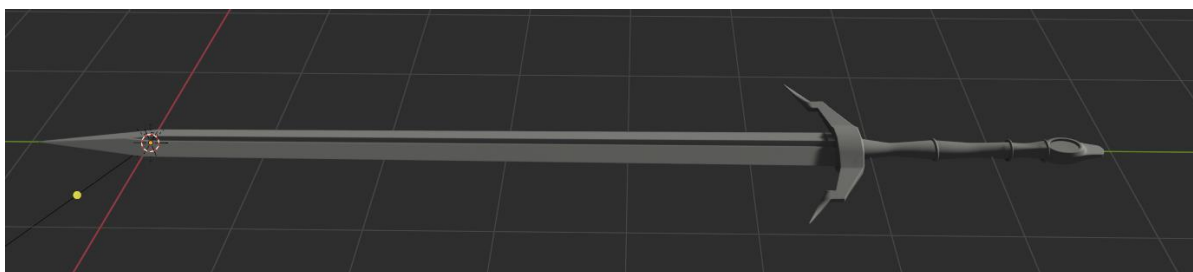


Рис. 3.3 – Модель меча у «Blender»

Для початку почнемо з референсу, підійде будь-який малюнок готового меча, начерки і т.д.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 54 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

У моєму випадку, я вирішив взяти креслення. У *Blender* у вкладці «*Add*» вибираємо *Image* і закріплюємо як референс (*Reference*) Починаємо з куба, для цього ми будемо його масштабувати вручну за допомогою панелі *Object Properties*.

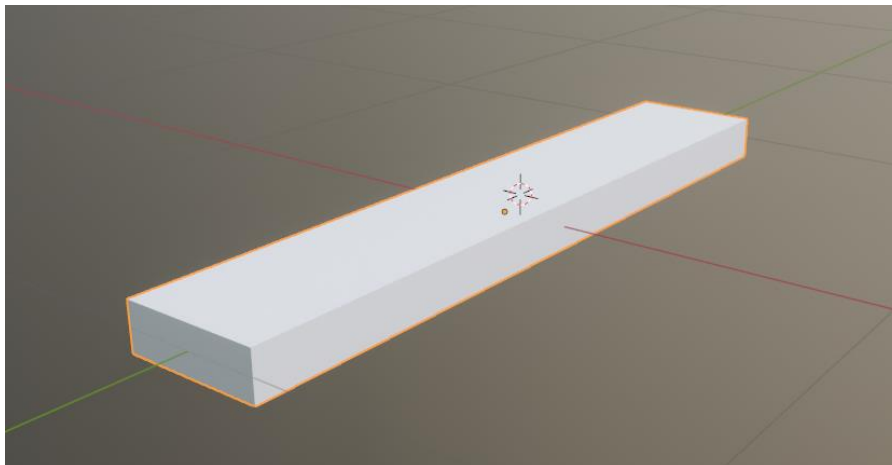


Рис. 3.4 – Модель після «Object Properties»

Після переходимо в режим редагування (*Edit mode*) через гарячу клавішу *TAB*. Вибираємо всі грані через клавішу *A* і через праву кнопку миші накладаємо *Subdivide*. Цей інструмент використовується для додавання нових геометричних деталей до сітки шляхом ділення наявних граней, ребер і вершин.

Так само, може збільшувати щільність сітки, водночас створюючи додаткові вершини та грані, що дає змогу моделювати більш деталізовані та складні форми.

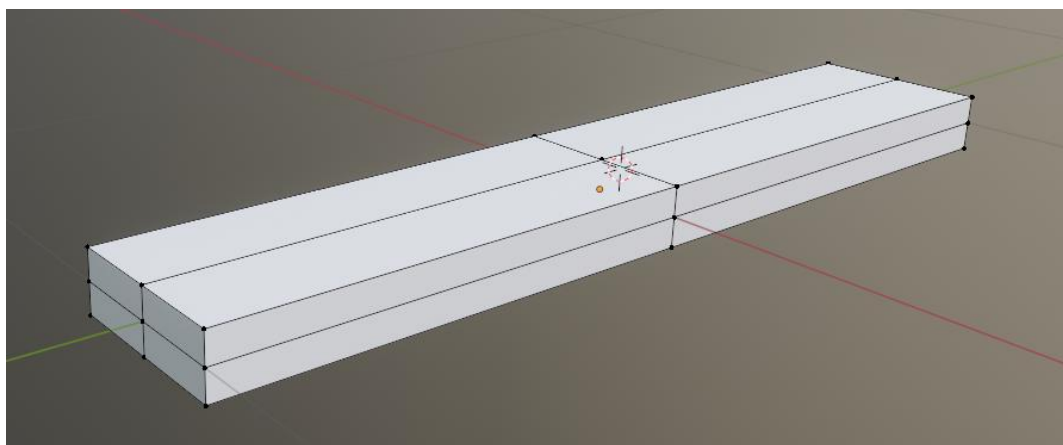


Рис. 3.5 – Модель після накладення «Subdivide»

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 55 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

Тепер, ми повинні з'єднати вершини (*Vertex*) у центрі. У режимі *Edit Mode* (перемиканням *TAB*) виділяємо вершини грані з будь-якого боку і переходимо правою кнопкою миші в контекстне меню, де вибираємо інструмент *Merge Vertices*, він дає змогу об'єднувати дві або більше вершин в одну. Так само робимо і з бічними вершинами, з'єднуючи їх у центрі.

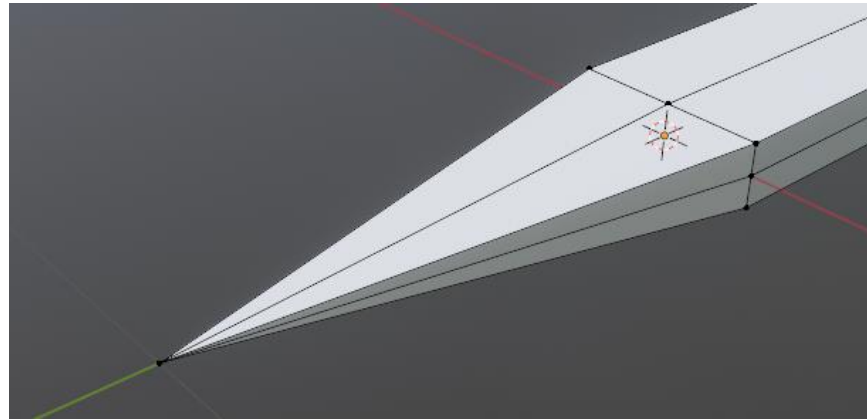


Рис. 3.6 – Після «*Merge Vertices*»

Тепер, нам потрібно зробити дол меча. Робиться це за допомогою двох інструментів *Bevel* – він використовується для створення скошених або закруглених країв на моделях. Він додає додаткові геометричні елементи (ребра і грані) до об'єкта, що робить краї менш різкими і більш реалістичними.

А, *Loop Cut* – використовується для додавання нових ребер навколо моделі, слідуючи контуру обраних граней. Це дає змогу розділити модель на більш дрібні частини, полегшуючи подальше моделювання та деталізацію.

Далі, де зробили скіс, за допомогою *Loop Cut* робимо «розріз», а потім вибираємо грань і масштабуємо її по *Z* вниз. Результат показано на малюнку 3.12.

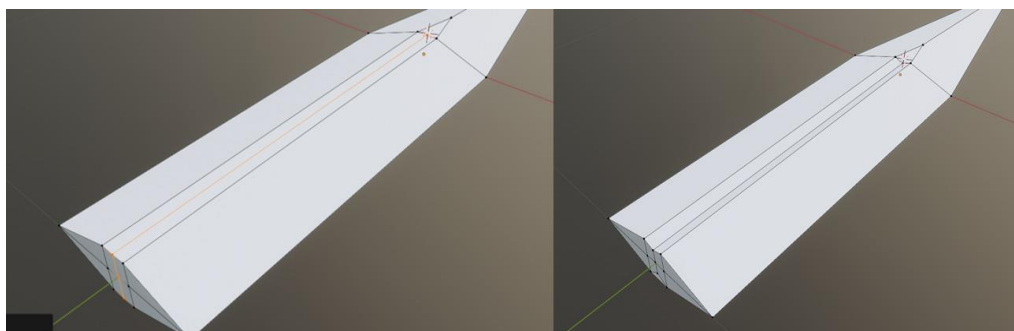


Рис. 3.12 – Результат роботи

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 56 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

Потім, нам потрібен модифікатор *Subdivision Surface* – він використовується для згладжування і додавання деталей до 3D-моделей. Він працює шляхом ітеративного поділу, кожної грані сітки на більш дрібні частини, що дає змогу отримувати плавні та деталізовані поверхні з відносно простої геометрії.

Нам потрібен, для згладжування країв леза, налаштування опцій – накладення 2 рази дасть, нам змогу зробити леза більш загостреними з боків.

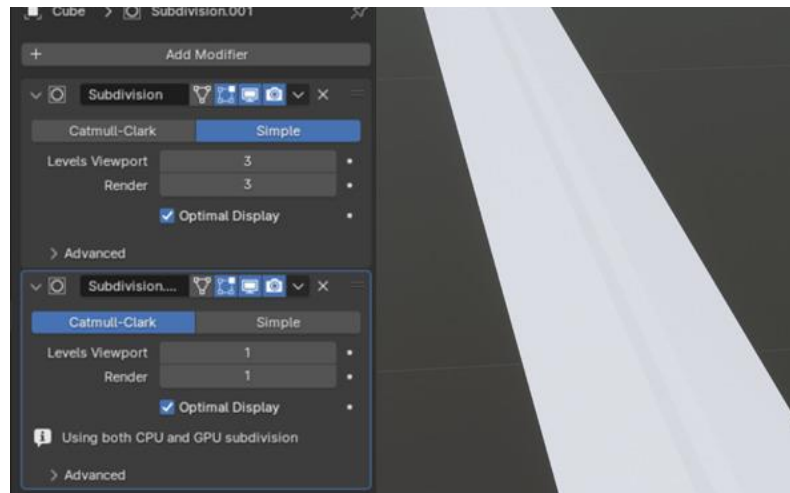


Рис 3.13 – Лезо при «*Subdivision Surface*»

Лезо готове, тепер переходимо до гарди. Для цього ми створюємо циліндр у вкладці «*Add*» і задаємо йому такі параметри: вертикалі 10, у *Cap Fill Type* вказуємо *Nothing* Після створення і налаштування на початку, вказуємо параметри, обертання по X на 90 градусів і масштабуємо (*Scale*) в *Object Properties*.

Потім, переміщаємо до леза на незагострену його частину.

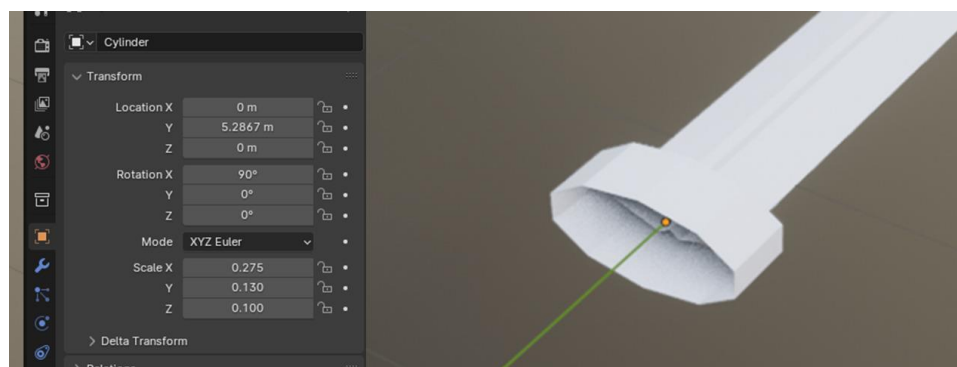


Рис. 3.14 – меч при «*Subdivision Surface*»

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 57 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

Переходимо до референсу, щоб зробити гарду. Вибираємо грань і починаємо його екструдити по самому референсу за допомогою інструменту *Extrude*.

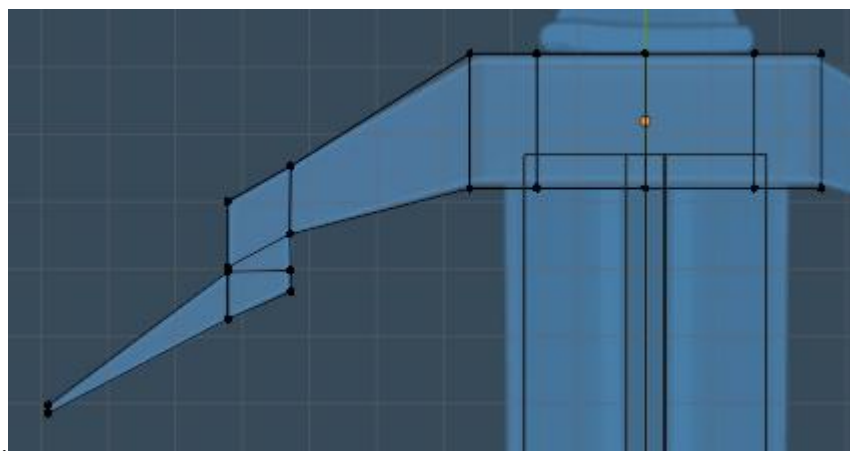


Рис. 3.15 – Обведення за референсом

Далі як обвели, заповнюємо грані й накладаємо модифікатор *Mirror*, він дає нам змогу відобразити об'єкт за потрібною віссю, приміром, наразі нам потрібно за X.

На початку, коли ми створили циліндр, ми вказали, вказали *Nothing*, що дало нам ці порожнечі і нам потрібно заповнити їх. Робиться за допомогою інструменту *Fill* (гаряча клавіша *F*), вибираємо грані в *Edit Mode* і заповнюємо.

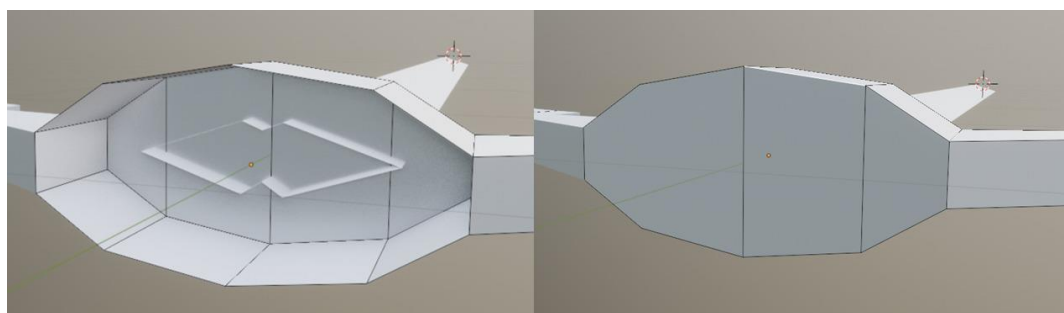


Рис. 3.16 – Гарда меча

Тепер, переходимо до рукояті, ми вже створили заготовки раніше, використовуючи *Loop Cut* ділимо на 3 частини та скейлимо, а потім накладаємо модифікатор *Subdivision Surface* Додаємо об'єкт *Torus*, це будуть наші кільця при створенні вказуємо параметри і переміщаємо ці кільця по рукояті за допомогою інструменту *Grabe* (гаряча клавіша *G*).

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 58 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

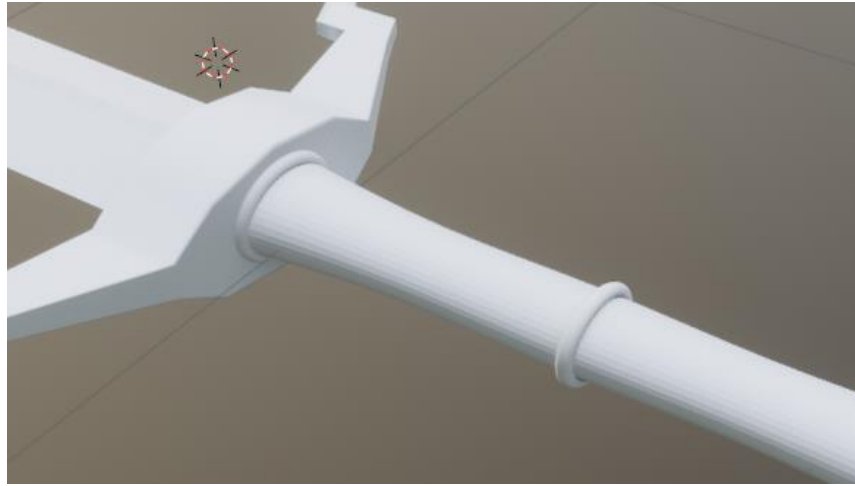


Рис. 3.17 – Рукоять меча з кільцями

Продовжуємо робити рукоять, переходимо до наступної заготовки. Накладаємо модифікатор *Subdivision Surface*, вибираємо вершини кінця циліндра і сколюємо, а потім за допомогою *Loop Cut* знаходимо центр і трохи зменшуємо.



Рис. 3.18 – Рукоять меча форма

Далі, робимо кінець. За допомогою *Loop Cut* знаходимо центр циліндра, потім заповнюємо його вгору і починаємо масштабувати (*Scale*). Потім вгору робимо з інструментом *Inset* – використовується для створення нового контуру всередині обраної грані або групи граней, відступаючи їхні краї всередину. Це дає змогу швидко створювати рамки, панелі або заглиблення на поверхні моделі. За допомогою інструменту *Extrude*, зсуваємо вниз по *Z* і за допомогою *Inset* і потім з'єднуємо саме коло в центрі за допомогою *Merge*.

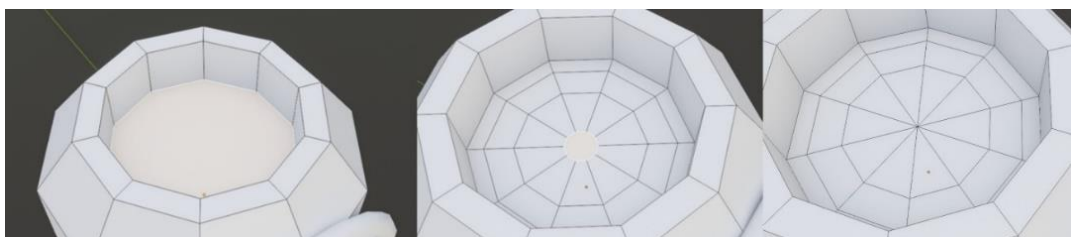


Рис 3.19 – Наверх меча надання йому форми

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дат | | 59 |

Тепер від кінця екструдуюємо і масштабуємо по *X* і потім відпускаємо за допомогою інструменту *Move* вниз. Потім, видаляємо низ, вибираємо об'єкт і використовуємо *Mirror* і масштабуємо по *X* і по *Y* і накладаємо *Subdivision Surface*, якщо треба трохи редагуємо.

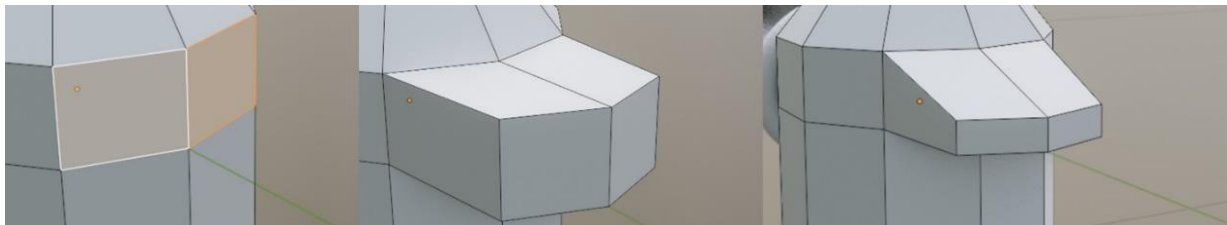


Рис. 3.20 – Наверх меч надання йому форми



Рис. 3.21 – Готове навершя меча

У нас виходить готовий меч, який був на малюнку 3.3.

3.3 Текстурування меча

До текстурування, потрібно зробити *UV*-розгортку в *Blender*. З того, що у нас у більшості випадків прості геометричні форми. Можна скористатися *Smart UV Project*, воно саме нам зробить якісні карти. Потім, трохи відокремимо і все готово. Після, експортуємо в *FBX*, при цьому в параметрах експорту вказуємо *Apply Transform*, а також прибираємо все і залишаємо лише *Mesh*.

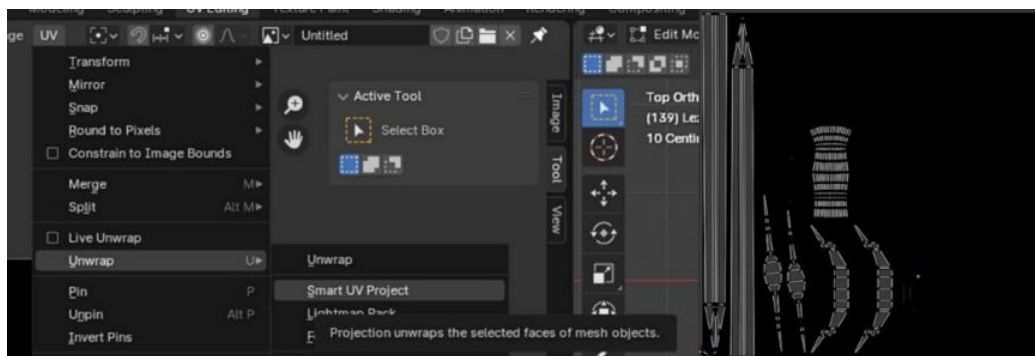


Рис. 3.22 – Автоматична розгортка у «Blender»

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дат | | 60 |

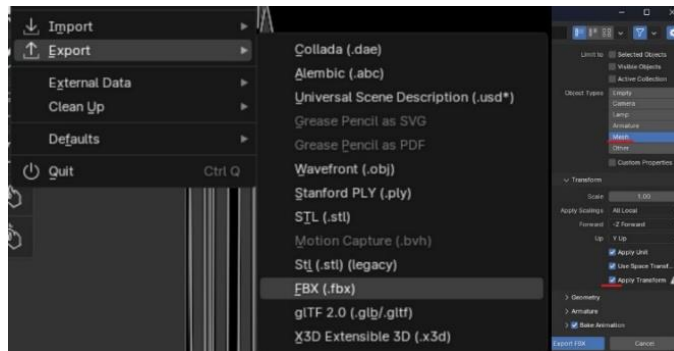


Рис. 3.23 – Експорт файлу FBX у «Blender»

У програмі для текстуригу *Substance Painter* імпортуємо файл *FBX*. У *template* вибираємо пресет *Unreal Engine 4*. Це дасть змогу уникнути нам менше конфлікту в майбутньому. Так само в *Document Resolution* (Роздільна здатність документа) – визначає роздільну здатність текстур, які використовуються в проєкті, вкажемо 2048x2048. А, *auto-unwrap* не обираємо, оскільки ми зробили вже розгортку в *Blender*.

Import Baked Maps (Імпорт запечених карт) – опція для імпорту вже запечених карт (наприклад, карт нормалей, *ambient occlusion* тощо).

Це налаштування відкривається під час натискання на стрілку ліворуч від тексту. *Color Management* (Керування кольором) – налаштування керування кольором у проєкті, що важливо для коректного відображення текстур і матеріалів. Це налаштування також відкривається після натискання на стрілку ліворуч від тексту.

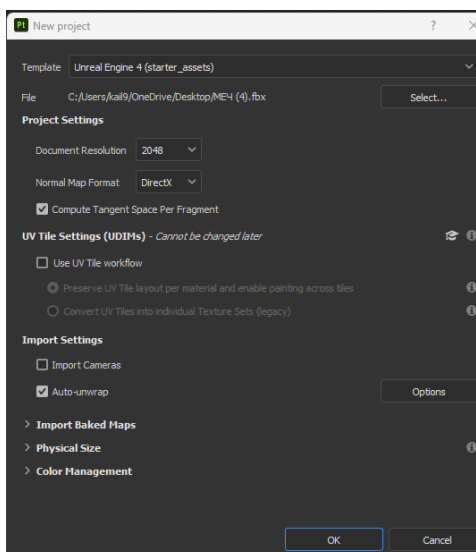


Рис. 3.24 – Первинні налаштування під час імпорту в *Substance Painter*

Після створення, переходимо в *Texture Set List* вибираємо потрібну нам *UV* карту. Для прикладу візьмемо рукояць, у шарах створюємо *Fill Layer* і до нього додаємо чорну маску. У масці вказуємо потрібну нам текстуру, оскільки в нас ручка візьмемо дерево.

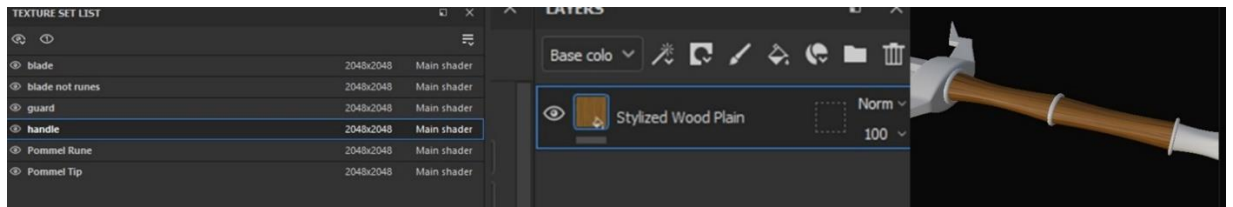


Рис. 3.25 – Текстуриг рукояці меча

Після, підбираючи матеріали, накладаємо потрібні, при цьому регулюємо текстуру.

У цьому меню *Projection* (Проекція) – вибір методу проєкції текстури на модель. Тут обрано *UV-projection* (*UV*-проєкція), це означає, що текстура буде проєктуватися на модель на основі її *UV*-розгортки.

Filtering (Фільтрація) – вибір методу фільтрації текстури для підвищення її якості. У цьому разі використовується *Bilinear* (Білінійна фільтрація, висока якість), що забезпечує згладжування текстури.

1. *UV Wrap* (*UV* обгортання) – опція повторення текстури за *UV*-координатами. *Repeat* (Повтор) означає, що текстура повторюватиметься по краях *UV*-розгортки. Регулюючи, ми можемо домогтися гарного результату віддзеркалення.

2. Металеве світло ми так само досягаємо за допомогою параметра металіка в *Channels*.

3. *Metal Type* (Тип Металу) – меню, що випадає для вибору типу металу. У цьому разі обрано «*Custom Metal*» (Користувацький метал).

4. *Metal Color* (Колір Металу) – поле для вибору кольору металу.

5. *Metal Color Variation* (Варіація кольору металу) – повзунок для налаштування ступеня варіації кольору металу.

6. *Metal Roughness* (Шорсткість Металу) – повзунок для налаштування шорсткості металу.

| | | | | | | |
|------|------|----------|--------|-----|-----------------------------|------|
| | | | | | <i>КРБ.КІ.1.442-03.1.10</i> | Арк. |
| | | | | | | 62 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

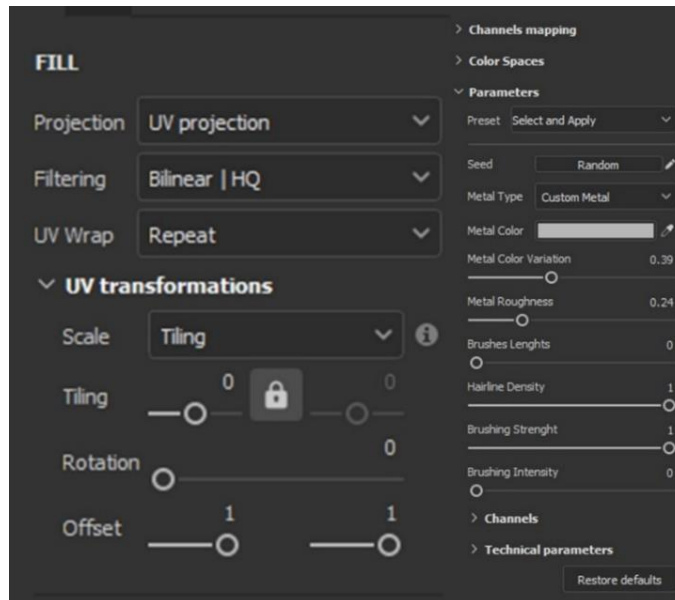


Рис. 3.26 – Параметри текстури

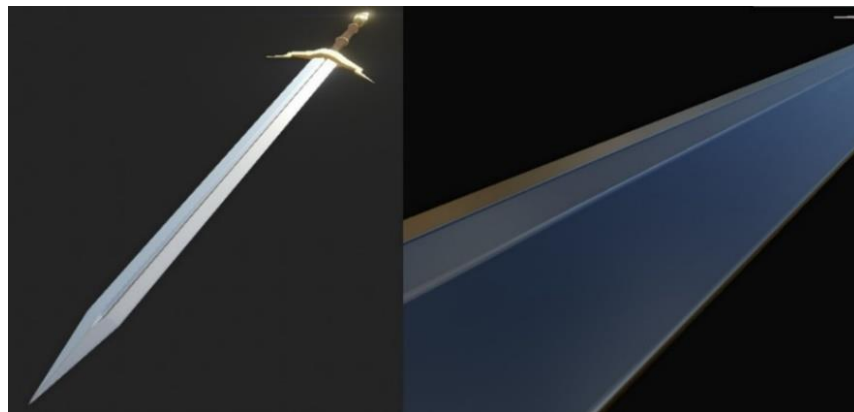


Рис. 3.27 – Готовий меч із текстурами без рун

Залишилося додати якісь ієрогліфи, щоб надати міфічності. Спочатку шукаємо матеріал.

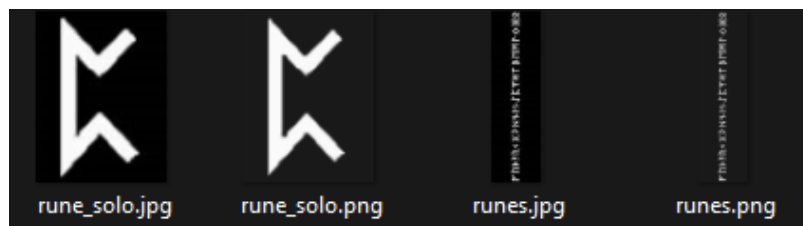


Рис. 3.28 – Матеріал

Тепер експортуємо всі руни в *Substance Painter*, простим перетягуванням. Тут вибираємо як текстуру, потім зберігаємо, вибираючи в цій сесії Тепер переносимо руни на клинок, трохи редагуємо на *UV*-розгортки і на каналі вмикаємо *Emission*, так само робимо і з руків'ям. Тепер дивимося на результат.

| | | | | | | |
|------|------|----------|--------|-----|-----------------------------|------|
| | | | | | <i>КРБ.КІ.1.442-03.1.10</i> | Арк. |
| | | | | | | 63 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

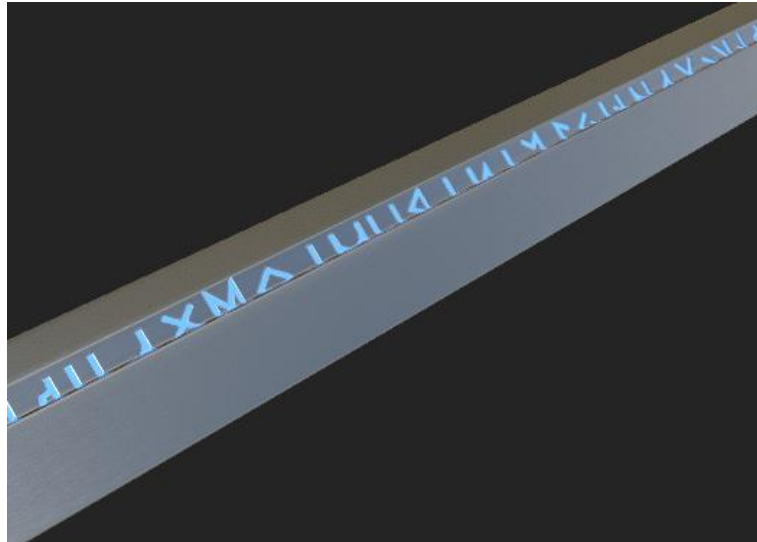


Рис. 3.29 – Текстура меча з рунами

3.4 Побудови сцени у *Unreal Engine*

Створення сцени в *Unreal Engine* може включати в себе кілька кроків, від проектування ландшафту до налаштування освітлення і розміщення об'єктів.

Для початку запусимо *Unreal Engine*, нас зустрічає хаб на кшталт *Unity*, так ми робимо «демонстраційну» стіну. Вибираємо у вкладці «*Film/Video&LiveEvents*» обираємо сцену *Blank*.

Це стандартна сцена без зайвих деталей, де є платформа, камера і стандартне освітлення.

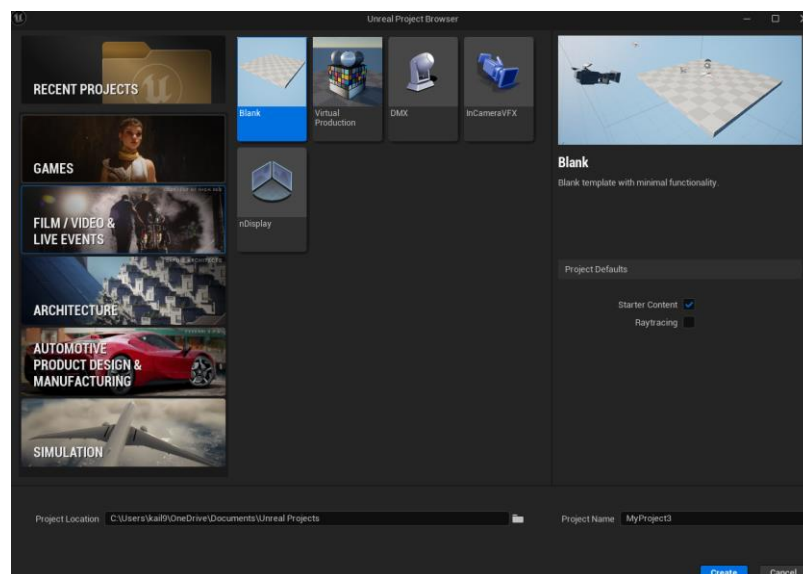


Рис. 3.30 – Хаб *Unreal Engine 5*

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 64 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

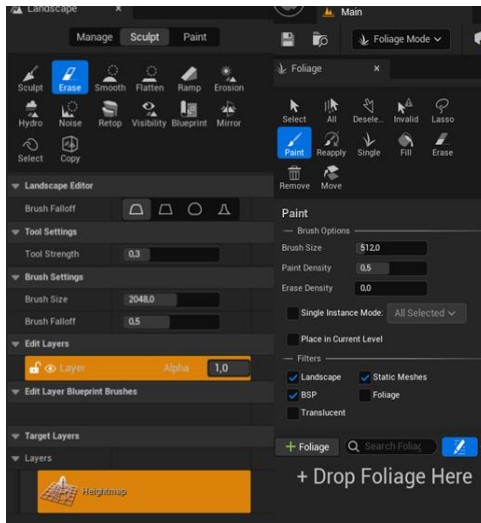


Рис. 3.33 – Інструменти для скульптингу ландшафту та параметри *Foliage Mode*



Рис. 3.34 – Сцена зі зміненим ландшафтом і розставленими деревами

3.4.1 Імпорт моделей

Коли модель готова або знайшли потрібну, ми експортуємо її з програми для 3D-моделювання у форматі, підтримуваному *Unreal Engine*, як *FBX* або *OBJ*.

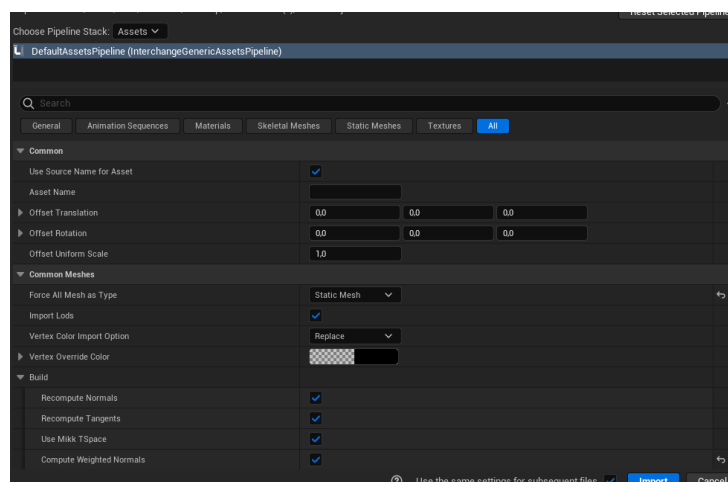


Рис. 3.35 – Інтерфейс *Unreal Engine* під час перенесення асету

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дат | | 66 |

Потім, ми переходимо в Niagara System, перекидаємо наш матеріал. А пізніше додаємо емітер частинок, налаштовуємо його параметри, як швидкість, розмір і напрямок частинок, щоб вони падали вниз, імітуючи дощ. Також використовується спеціальний матеріал для частинок, щоб надати їм прозорості та візуальних характеристик крапель дощу.

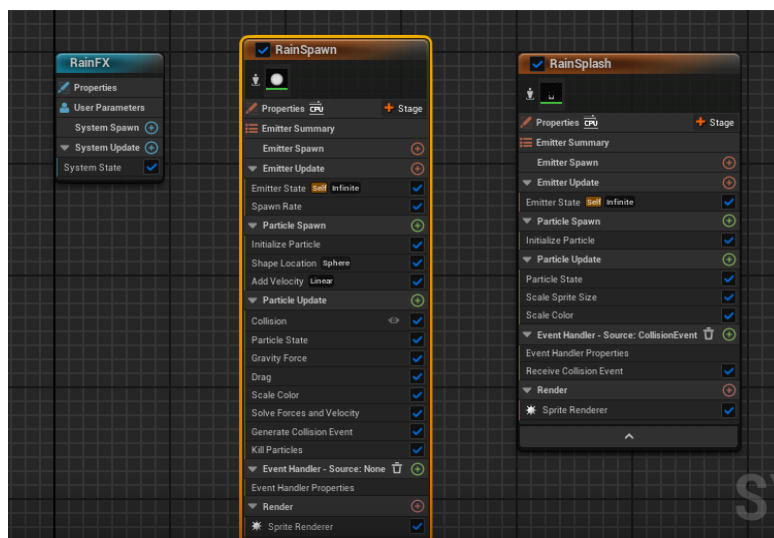


Рис. 3.37 – Інтерфейс Niagara для створення дощу

3.4.3 Використання блюпринтів

Створюємо новий матеріал, і за допомогою *Blueprint* робимо туман, використовуючи такі параметри:

1. *Camera Position* і *Actor Position* – визначають позиції камери й актора.
2. *Subtract* і *Distance* – обчислюють відстань від камери до об'єкта.
3. *Divide* – нормалізує значення відстані.
4. *Clamp* – обмежує значення в допустимих межах.
5. *Multiply* – масштабує значення, такі як інтенсивність туману.
6. *Texture Sample* – використовує текстуру для деталізації туману.
7. *Fog Color* і *Fog Density* – керують кольором і щільністю туману.
8. *Add* – підсумовує вкладки у фінальний ефект.
9. *Material Attributes* – об'єднує всі параметри у фінальний матеріал.

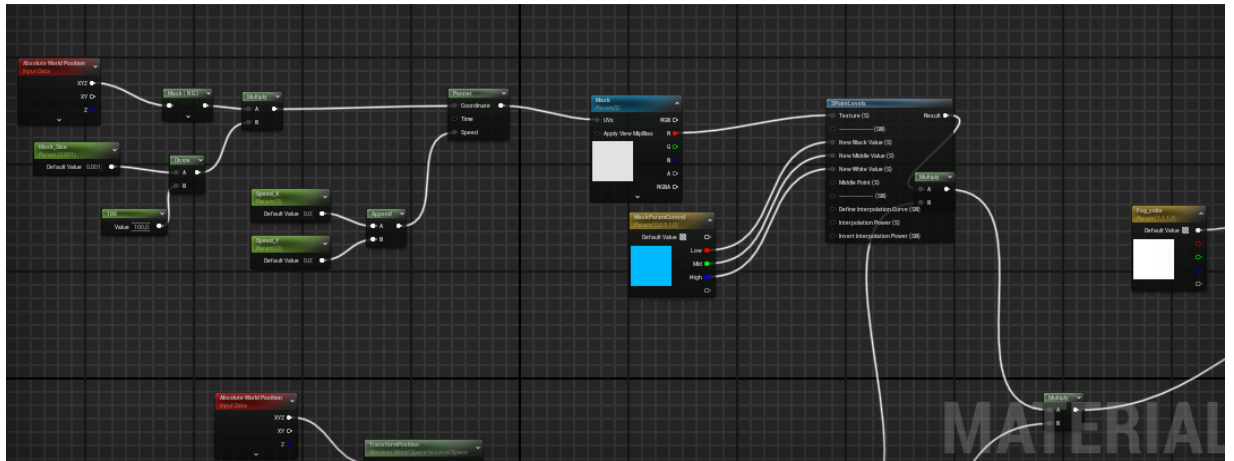


Рис. 3.38 – Створення туману за допомогою Blueprint

Тепер, створюємо куб і накладаємо на нього матеріал, після чого отримуємо туман, який ми можемо регулювати в налаштуваннях.



Рис. 3.39 – Тустування тумана у сцені

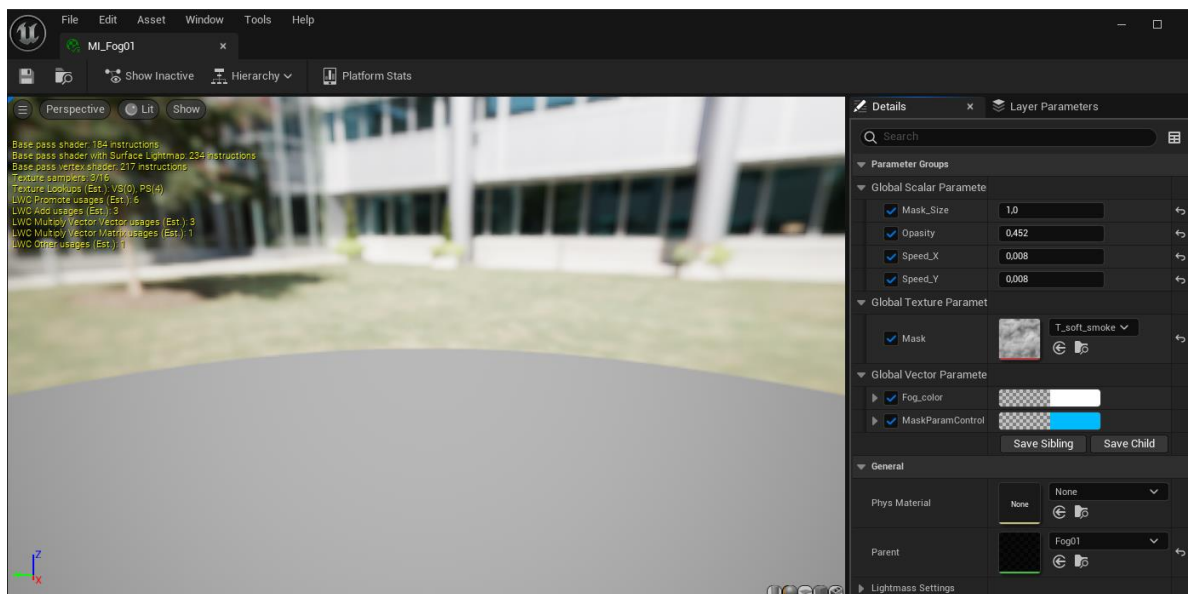


Рис. 3.40 – Налаштування туману його щільності, кольору і т.д

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дат | | 69 |

3.4.4 Використання освітлення

Unreal Engine має просунуту систему освітлення, яка забезпечує створення реалістичних і динамічних сцен. Незалежно від того, чи створюєте ви гру, архітектурну візуалізацію, кінематографічний проєкту або будь-який інший вид інтерактивного досвіду, правильне освітлення здатне значно посилити візуальне сприйняття й атмосферу вашої роботи.

Приклад «*Lumen*» для глобального освітлення та віддзеркалень у реальному часі або «*Nanite*» для високодеталізованої геометрії.

Існує 4 типи джерел освітлення:

1. *Direct Light* (Директне світло) – основне джерело світла, часто використовуване для симуляції сонячного світла.

2. *Point Light* (Точкове світло) – випромінює світло в усіх напрямках з однієї точки.

3. *Spot Light* (Прожектор) – випромінює світло в конусоподібній формі.

4. *Rect Light* (Прямокутне світло) – випромінює світло з плоскої поверхні, що підходить для освітлення великих областей.

Для свого проєкту я використовував *Direct Light* для місячного освітлення і *Point Light* для самого замку, щоб надати замку більш житлового вигляду. А так само для підсвічування меча було використано *Spot Light*, щоб сконцентрувати ще більшу увагу на мечі та його таємничість.

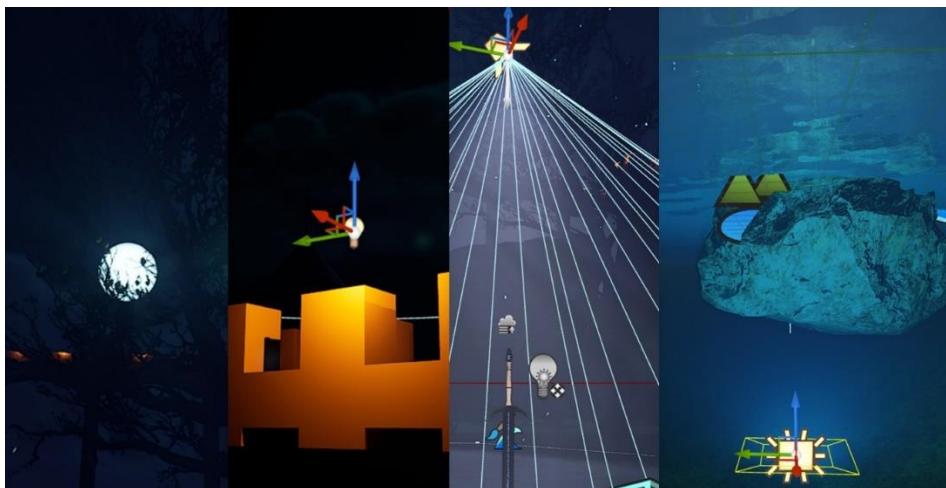


Рис. 3.41 – Різні джерела освітлення у UE5

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 70 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

3.5 Створення відео

3.5.1 Додавання камери і її анімація

Анімація камери в *Unreal Engine 5 (UE5)* здійснюється за допомогою інструменту *Sequencer*, який надає можливість створення кінематографічних сцен з високою точністю та гнучкістю. Основний принцип роботи з *Sequencer* полягає у використанні тимчасової шкали для керування анімацією, включаючи налаштування ключових кадрів, кривих анімації та інших параметрів.

Для початку роботи створюється новий рівень *Sequencer* у проекті, що дозволяє організувати анімаційні дані. У вікні *World Outliner* обирається опція створення *Level Sequence*, після чого можна додати камеру до цього рівня. Камеру, створену через *Cine Camera Actor*, розташовують у потрібному місці сцени, використовуючи видошукач для точного налаштування кута та положення.

Анімація положення камери починається з додавання ключових кадрів для її початкового положення та обертання на початку тимчасової шкали. Переміщуючи часовий індикатор на різні моменти часу, змінюють параметри позиції та обертання камери, додаючи відповідні ключові кадри для кожної зміни. Це створює плавний рух камери протягом визначеного проміжку часу.

Інші параметри камери, такі як фокусна відстань та глибина різкості, також можуть бути анімовані шляхом додавання ключових кадрів для відповідних параметрів. Це дозволяє детально контролювати всі аспекти візуалізації та створювати складні кінематографічні ефекти.

Тимчасова шкала *Sequencer* складається з кількох ключових елементів, таких як часовий індикатор, доріжки, ключові кадри та криві анімації. Часовий індикатор показує поточний час анімації та дозволяє переглядати і редагувати різні моменти. Доріжки представляють об'єкти або параметри, що анімуються, такі як положення або обертання камери. Ключові кадри визначають конкретні значення параметрів у певні моменти часу, і можуть бути додані, видалені або переміщені для створення бажаної анімації. Криві анімації показують зміни

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 71 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

параметрів між ключовими кадрами та можуть бути відредаговані для досягнення плавних переходів і контролю інтерполяції параметрів.

Робочий процес з *Sequencer* включає створення логічних сцен, анімацію необхідних об'єктів з додаванням ключових кадрів, регулярний перегляд та редагування анімації для досягнення бажаного результату. Коли анімація готова, її можна експортувати як відео або використовувати безпосередньо в проєкті.

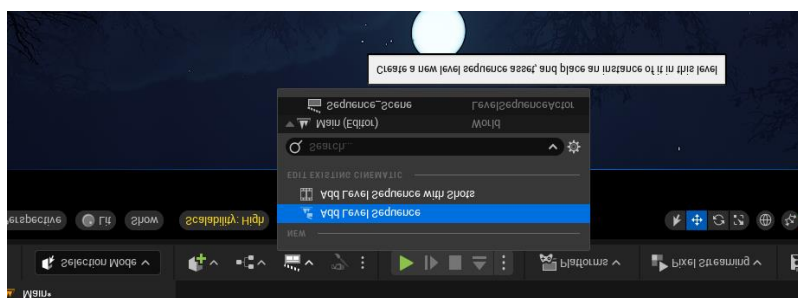


Рис. 3.42 – Створення сцени

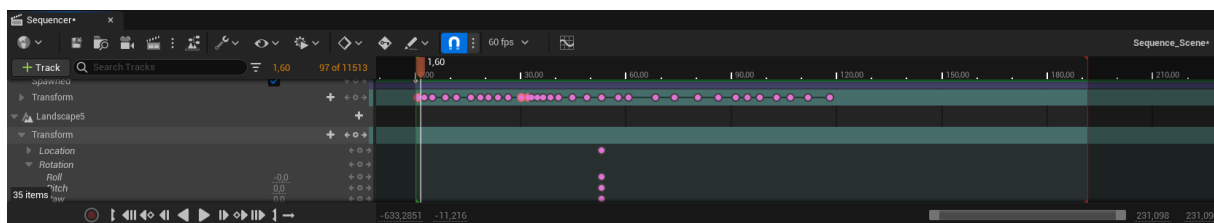


Рис. 3.43 – Таймлінія у *Sequence*

Так само ми можемо налаштувати камеру, використовуючи такі налаштування:

1. Фокусна відстань (*Focal Length*) – визначає кут огляду камери. Менше значення створить більш широкий кут (ширококутний об'єктив), більше значення – вузький кут.

2. Діафрагма (*Aperture*) – контролює глибину різкості. Менше значення (широка діафрагма) створить малу глибину різкості (розмитий фон).

Це лише малі налаштування. Для того, щоб створити анімацію камери, ми пересуваємо камеру за ключовими кадрами в тимчасовій шкалі (*Sequencer*) і переміщаємо її по сцені. Вкладка в *Transform* допоможе зробити зміни більш точно, таке як *Roll*, *Yaw*, *Location*.

Усі ці налаштування допомагають у створенні сцени, наприклад:

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 72 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

1. Широкі пейзажні кадри – будемо ширококутний об'єктив з довгою глибиною різкості, щоб захопити більше деталей сцени.

2. Портретні кадри – ми застосуємо вузький кут огляду і малу глибину різкості, щоб виділити об'єкт на розмитому тлі.

3. Динамічні сцени – анімуємо камеру за допомогою *Sequencer* для створення ефектних рухів.

3.5.2 Візуалізація і постобробка

З точки зору візуалізації, було використано *Directional Light*, що імітує місячне світло. Це світло створює м'які, довгі тіні і надає сцені холодного синього відтінку, характерного для нічного освітлення. навколо меча видно невелике світіння, створене за допомогою точкового джерела світла *Point Light* для привернення уваги.

За допомогою постобробки було застосовано такі налаштування:

1. Глибина різкості (*Depth of Field*) – використано, щоб злегка розмити задній план і акцентувати увагу на центральному об'єкті — мечі в камені. Це дозволяє створити відчуття глибини та фокусує увагу глядача на головному елементі сцени.

2. Кольорокорекція та тональний маппінг (*Color Grading and Tone Mapping*) – застосовані для створення холодної нічної атмосфери. Кольорові коригування зміщені в бік синіх відтінків, що посилює нічний ефект. Використання тонального маппінгу дозволяє зберегти деталізацію у світлих та темних областях зображення, створюючи збалансований і природний вигляд сцени. Це також допомагає підкреслити ключові елементи композиції, зокрема меч, і забезпечити емоційний вплив на глядача.

3. Віньєтування (*Vignetting*) – легке віньєтування додано по краях зображення, щоб привернути увагу до центру сцени і створити ефект занурення. Це підкреслює головний об'єкт і додає сцені кінематографічного вигляду.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 73 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

Віньєтування зосереджує погляд глядача на центральному елементі, зменшуючи відволікаючі фактори по краях кадру. Це також додає сцені певну атмосферність і глибину, підсилюючи візуальний ефект нічного освітлення.

4. Ефекти частинок (*Particle Effects*) – видимі точки, що світяться, імітують магічну частинку, а частинки дощу надають зловісності. Ці ефекти додають сцені динамічності та підсилюють враження магії та загадковості.

5. Постобробка неба (*Skybox and HDRI*) – нічне небо створено за допомогою динамічної *Skybox* текстур. Через динамічність небо трохи пересувається, що додає сцені живого і реалістичного вигляду. Використання *HDRI (High Dynamic Range Imaging)* забезпечує детальний і реалістичний фон, що гармонійно поєднується з освітленням сцени. Динамічний *Skybox* додає реалізму і природності, відображаючи зміни в освітленні та атмосфері, що сприяє більш глибокому зануренню глядача в сцену. Це дозволяє створити переконливий і візуально привабливий фон, який доповнює основні елементи сцени.

Висновки до третього розділу

1. В ході виконання було досягнуто успішного створення високоякісної тривимірної моделі меча, що інтегрована в реалістичну сцену за допомогою сучасних інструментів і методів. Процес моделювання в *Blender* включав інструменти самої програми. Також, розглянуто використання *Smart UV-Project* для ефективною *UV*-розгортки, що забезпечило високу якість текстур. Подальше текстурування у *Substance Painter* дозволило створити детальні і реалістичні текстури, використовуючи пресет *Unreal Engine 4* і роздільну здатність 2048x2048 пікселів.

2. Імпорт моделі в *Unreal Engine* супроводжувався налаштуванням параметрів масштабу, типу мешу, імпорту матеріалів і текстур, що забезпечило коректне відображення моделі у сцені. Створення ефектів за допомогою системи

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 74 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

частинок *Niagara*, таких як дощ, а також застосування *Blueprint* для створення туману, підсилили реалістичність і атмосферність сцени.

3. Використання просунутих технік освітлення в *Unreal Engine*, таких як *Direct Light* для симуляції місячного світла і *Point Light* для підсвічування ключових елементів, додало сцені глибини і виразності. Крім того, налаштування постобробки, включаючи глибину різкості, кольорокорекцію і тональний мапінг, дозволили створити необхідний настрій і посилити візуальний ефект.

4. Процес створення відео, що включав анімацію камери і налаштування її параметрів, таких як фокусна відстань і діафрагма, забезпечив динамічність і привабливість фінального результату. Візуалізація сцени з використанням різних технік освітлення і постобробки продемонструвала можливості сучасних інструментів для досягнення високого рівня деталізації та реалістичності.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 75 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

РОЗДІЛ 4

ЕКОНОМІЧНА ЧАСТИНА

4.1 Організаційно-економічне та маркетингове обґрунтування проекту

4.1.1 Порівняльний аналіз пропонованого проекту

У даній кваліфікаційній роботі було проведено дослідження процесу проектування та розробки сцени "Меч у камені" у відомому ігровому движку *Unreal Engine*.

Головною метою проекту було створення прототипу сцени, що демонструє вражаючу візуалізацію миттєвостей легендарного образу.

У процесі розробки сцени були використані різноманітні технології та інструменти. Наприклад, для моделювання використовувалася програма *Blender*, яка надає широкі можливості у створенні складних тривимірних об'єктів. Для створення текстур та матеріалів був використаний *Substance Painter*, що дозволив долучити реалістичні деталі та текстури до сцени.

Перед початком розробки був проведений аналіз подібних сцен у ігровій індустрії та фільмографії, що допомогло зрозуміти ключові аспекти візуальної привабливості та функціональності. Були розглянуті різні стилізації та підходи до втілення міфічних сценаріїв у тривимірному середовищі.

Використання *Unreal Engine* як ігрового рушія дало змогу досягти високої якості графіки та продемонструвати можливості самого рушія.

Також використання інструментарію дозволило швидко реалізувати ідеї та втілити їх у життя. Завдяки інтеграції різних програмних пакетів та технологій, вдалося створити сцену, яка не лише візуально приваблива, але й технічно досконала.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 76 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

Загальна таблиця з перевагами та недоліками використання *Unity* та *Unreal Engine 5* для розробки сцени «Меч у камні»

| Особливості | <i>Unreal Engine 5</i> | <i>Unity</i> |
|------------------------|---|--|
| Переваги | | |
| Графічні можливості | Можливості для VR і AR, фотореалізм, якісні ефекти. | Більш обмежені візуальні ефекти, але при цьому все одно непогані графічні можливості. |
| Графічний інтерфейс | Громіздкий, але потужний і гнучкий. | Простий та інтуїтивно зрозумілий. |
| Мови програмування | <i>C++</i> , <i>Blueprints</i> | <i>C#</i> , <i>UnityScript</i> |
| Моделювання фізики | Моделювання фізики за допомогою потужного інструменту « <i>PhysX</i> ». | Моделювання фізики за допомогою <i>Unity Physics</i> . |
| Інструменти анімації | Потужні інструменти анімації. | Простий, інтуїтивний інтерфейс для базової анімації. |
| Платформи | Підтримка різних платформ(ПК, консолі, мобільні пристрої). | Підтримка різних платформ(ПК, консолі, мобільні пристрої). |
| Спільнота та підтримка | Велика спільнота розробників, активна підтримка <i>Epic Games</i> . | Також має велику спільноту, активний розвиток за допомогою магазину <i>Asset Store</i> . |

| Особливості | <i>Unreal Engine 5</i> | <i>Unity</i> |
|-------------|---|--|
| Недоліки | | |
| Оптимізація | Треба постійно слідкувати за оптимізацією. | Оптимізація легше. |
| Навчання | Складне через потужність і гнучкість. | Більш просте і швидке. |
| Ціна | Має безкоштовну версію, але з деякими обмеженнями та вимогами щодо сплати роялті після досягнення певного рівня доходу від вашої гри або додатка. <i>Unreal Engine</i> також пропонує підписку на доступ до додаткових ресурсів і послуг. | Має більш доступну безкоштовну версію, яка дає змогу розпочати роботу з розробкою ігор без витрат. Однак для більш серйозних проектів або для доступу до розширених функцій, таких як ви повинні будете придбати ліцензію. |

4.1.2 Основні положення

1. Клас – одиночний проект.
2. Тип проекту – змішаний.
3. Вид проекту – комбінований.
4. Термін проекту – короткостроковий.
5. Складність проекту – висока складність.
6. Масштаб проекту – корпоративний.

| | | | | | | |
|------|------|----------|--------|-----|-----------------------------|------|
| | | | | | <i>КРБ.КІ.1.442-03.1.10</i> | Арк. |
| | | | | | | 81 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

Етапи виконання розділів кваліфікаційної роботи:

1. Розробка робочого проекту – пошук референсів, пошук ідеї, складання схеми послідовного процесу і вивчення необхідних технологій програмного забезпечення – термін виконання 1 місяці.
2. Створення 3D моделі – термін виконання 4 дні.
3. Текстурування моделі – термін виконання 3 дні.
4. Підготовка сцени в рушії – термін виконання 7 днів.
5. Перенесення моделі в рушії – термін виконання 1 день.
6. Оптимізація сцени – термін виконання 7 днів.
7. Створення анімації – термін виконання 2 днів.
8. Візуалізація (створення камери, розміщення) – термін виконання 3 дні.
9. Постобробка (рендер) – термін виконання 3 дні.
10. Представлення готового проекту – термін виконання 7 днів.

Таблиця 4.3

Склад робіт по життєвому циклу проекту

| № код роботи | Назва роботи | T, Дні |
|--------------|---|--------|
| 0-1 | Вибір програмного забезпечення | 1 |
| 1-2 | Вивчення програмного забезпечення | 19 |
| 2-3 | Пошук референсів, пошук ідеї | 10 |
| 3-4 | Створення 3D моделі | 3 |
| 4-5 | Перенесення моделі в рушії | 1 |
| 5-6 | Оптимізація сцени. | 7 |
| 6-8 | Створення анімації | 2 |
| 5-7 | Візуалізація (створення камери, розміщення) | 3 |
| 7-8 | Постобробка (рендер) | 3 |
| 6-8 | Створення тестової версії | 2 |

| | | | | | | |
|------|------|----------|--------|-----|-----------------------------|------|
| | | | | | <i>КРБ.КІ.1.442-03.1.10</i> | Арк. |
| | | | | | | 82 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

| № код роботи | Назва роботи | T, Дні |
|--------------|-----------------------------------|--------|
| 8-9 | Тестування проекту | 1 |
| 9-10 | Представлення готового проекту | 7 |
| 11-12 | Оцінка результатів проекту | 1 |
| 12-13 | Підготовка підсумкових документів | 2 |

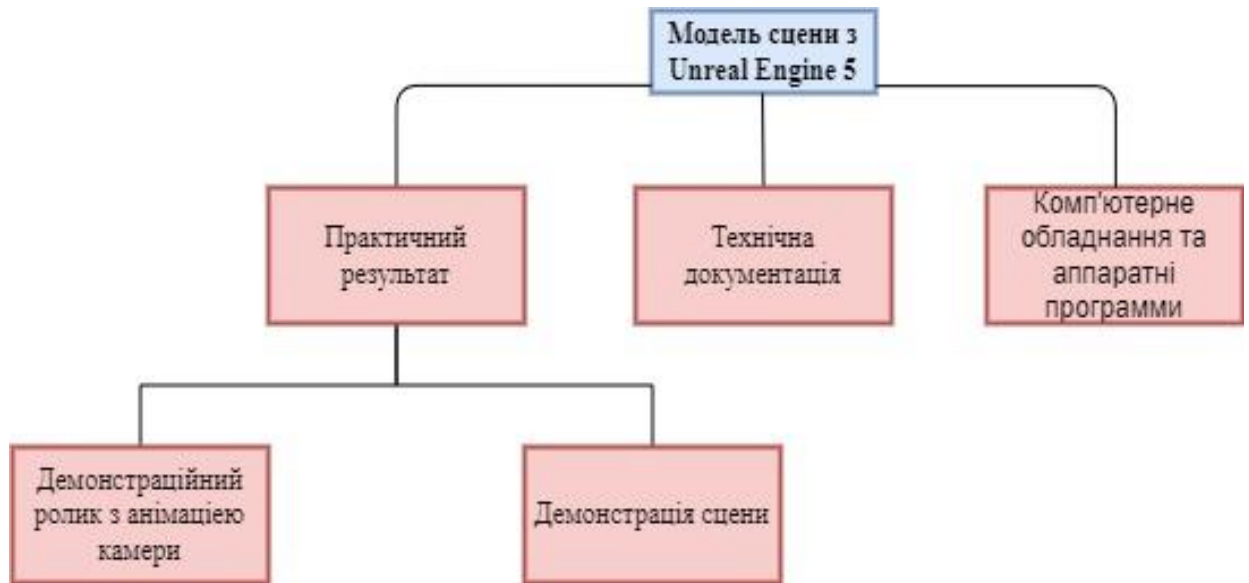


Рис. 4.1 – Структура (декомпозиція) проекту

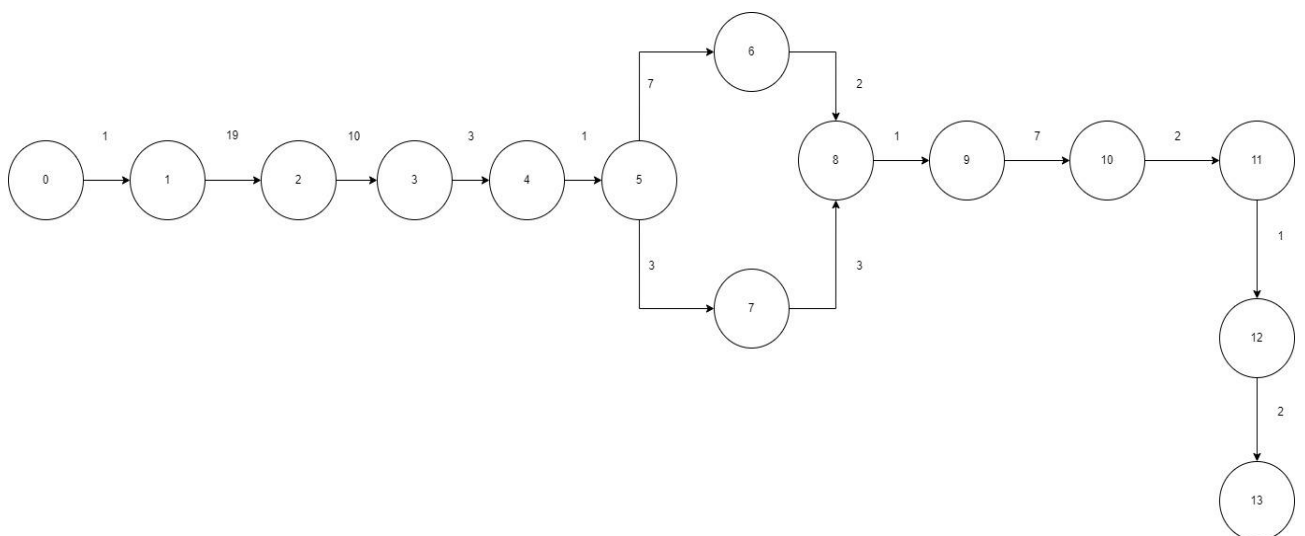


Рис. 4.2 – Мережний графік робіт

Розрахунок параметрів мережевого графіка:

T_{ij} – Тривалість робіт;

$T_{рн}$ – Ранній початок робіт;

$T_{ро}$ – Раннє закінчення робіт;

$T_{пн}$ – Пізній початок робіт;

$T_{по}$ – Пізнє закінчення робіт;

R_j – Резерв часу.

Таблиця 4.4

Розрахунок параметрів мережевого графіку

| Назва роботи | Попередня | T_{ij} | $T_{рн}$ | $T_{ро}$ | $T_{пн}$ | $T_{по}$ | R_j |
|--------------|-----------|----------|----------|----------|----------|----------|-------|
| 0-1 | - | 1 | 0 | 1 | 0 | 1 | 0 |
| 1-2 | 0-1 | 19 | 1 | 20 | 1 | 20 | 0 |
| 2-3 | 1-2 | 10 | 20 | 30 | 20 | 30 | 0 |
| 3-4 | 2-3 | 3 | 30 | 33 | 30 | 33 | 0 |
| 4-5 | 3-4 | 1 | 33 | 34 | 33 | 34 | 0 |
| 5-6 | 4-5 | 7 | 34 | 43 | 43 | 45 | 0 |
| 6-8 | 5-6 | 2 | 41 | 43 | 43 | 45 | 2 |
| 5-7 | 4-5 | 3 | 34 | 37 | 37 | 40 | 3 |
| 7-8 | 5-7 | 2 | 37 | 39 | 43 | 45 | 6 |
| 8-9 | 6-8; 7-8 | 1 | 43 | 44 | 45 | 46 | 2 |
| 9-10 | 8-9 | 2 | 44 | 46 | 46 | 48 | 2 |
| 10-11 | 9-10 | 2 | 46 | 48 | 48 | 50 | 2 |

4.1.3 Маркетингове обґрунтування проекту

Під час розробки сцени "Меч у камені" в *Unreal Engine 5* було проведено глибокі дослідження сучасних технологій створення віртуальних середовищ, аналіз існуючих інструментів для 3D-моделювання та текстурування, а також

| | | | | | | |
|------|------|----------|--------|-----|-----------------------------|------|
| | | | | | <i>КРБ.КІ.1.442-03.1.10</i> | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дат | | 84 |

використані методи генерації візуальних та анімаційних ефектів. Створений програмний продукт надає можливість користувачам зануритися у пророблений віртуальний світ.

Сцена "Меч у камені", створена в *Unreal Engine 5*, поєднує унікальний процес створення 3D-моделей з візуальними ефектами, створюючи захопливий неімерсійні досвід для користувачів. Вона може зацікавити як любителів 3D-моделювання, так і шанувальників віртуальної реальності.

Цей проект зможе здобути популярність на ринку завдяки своїй унікальності та високій якості. Захоплююча візуалізація та анімаційні ефекти створюють неповторний досвід для користувачів. Також рекламні кампанії та співпраця з іншими компаніями допоможуть привернути увагу користувачів і забезпечити успіх на ринку.

Перед визначенням ціни продукту для різних країн важливо враховувати фактори ціноутворення. Фактори аналізу ціноутворення для програмного забезпечення в різних країнах можуть включати:

1. Економічні умови – рівень доходів населення, інфляція, валютні коливання, економічна стабільність і безпека впливають на платоспроможність користувачів і можливості розробників.
2. Конку rentне середовище – кількість та якість інших програм на ринку, а також стратегії ціноутворення конкурентів.
3. Культурні вподобання – різниці в культурі та ігрових тенденціях можуть впливати на те, як користувачі сприймають ціни на програмне забезпечення.
4. Піратство та правова сфера – рівень піратства та ефективність захисту авторських прав може впливати на ціни, які розробники можуть встановити.
5. Податкова політика – рівень податків на програмну індустрію, в тому числі податки на продажі, може визначати цінову політику.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 85 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

6. Технологічні інновації – впровадження нових технологій, які можуть підвищити якість програмного забезпечення або знизити виробничі витрати, може впливати на ціни.

7. Доступність платформ – розрізнення між платформами (консолі, ПК, мобільні пристрої) і їх розповсюдженість в різних країнах.

8. Стратегії маркетингу та продажу – ефективність маркетингових кампаній, знижки, акції, сезонні розпродажі та інші маркетингові стратегії також впливають на ціни та продажі програмного забезпечення.

Для дослідження було порівняно ціни на інструменти для 3D-моделювання та текстурування в Україні, ЄС та США. Такий аналіз дозволяє визначити оптимальну цінову політику для нових продуктів, враховуючи регіональні відмінності в платоспроможності та попиті.

Таблиця 4.5

Порівняння ціни на популярні інструменти
на офіційному сайті розробника

| Інструменти | Україна (UAH) | ЄС (EUR) | США (USD) | Японія (JPY) |
|--------------------------|------------------|----------|-----------|-----------------|
| <i>Substance Painter</i> | 4500 | 120 | 130 | 15000 |
| <i>Photoshop</i> | 4500 | 120 | 130 | 15000 |
| <i>3dsMax</i> | 12000 | 320 | 350 | 40000 |
| <i>Autodesk Maya</i> | 12000 | 320 | 350 | 40000 |
| <i>Marmoset</i> | 5000 | 130 | 150 | 17000 |
| <i>3D-Coat</i> | 4500 | 120 | 130 | 15000 |

Порівняння цін на інструменти для 3D-моделювання та текстурування в Україні, Європейському Союзі, США та Японії показує цікаві результати.

Вартість *Adobe Photoshop* та *Substance Painter* в Україні становить 4500 гривень, що приблизно відповідає цінам у ЄС (120 євро) та США (130 доларів), але трохи перевищує японську ціну в 15000 єн. Таким чином, ці програми в Японії коштують дешевше. Щодо *3dsMax* та *Autodesk Maya*, ціни в Україні (12000 гривень) співпадають з європейськими (320 євро), але нижчі, ніж у США (350 доларів).

Водночас, ціни на ці інструменти в Японії (40000 єн) є найнижчими серед всіх розглянутих регіонів. Ціна *Marmoset* в Україні складає 5000 гривень, що трохи вища за європейську ціну (130 євро) та японську (17000 єн), але нижча за американську (150 доларів). Цікаво, що саме в США *Marmoset* коштує найдорожче.

На *3D-Coat* ціни в Україні знову ж таки становлять 4500 гривень, що аналогічно європейським та дещо нижче американських (130 доларів). В Японії ця програма найдешевша і коштує 15000 єн. Загалом, можна відзначити, що українські користувачі мають змогу придбати програмне забезпечення для 3D-моделювання та текстурювання за конкурентоспроможними цінами, які часто близькі до європейських, але нижчі за американські.

При цьому, японський ринок демонструє найнижчі ціни на більшість інструментів з розглянутих. Це робить український ринок привабливим для користувачів, які шукають доступні рішення.

4.2 Розрахунки ціни програмного продукту

Як вихідні дані для визначення трудомісткості розробки ПП використовується типовий склад етапів і укрупнені норми часу на розробку програмних засобів (ПЗ). Використаємо аналогічний підхід з типовим складом етапів і укрупненими нормами часу на розробку програмних засобів (ПЗ).

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 87 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

Припустимо, що обсяг розроблювального ПП складає 3000 умовних машинних команд і трудомісткість $T_p = 100$ люд.-годин.

Трудомісткість розробки ПП включає розробку наступних етапів:

1. Технічного завдання – ТЗ.
2. Технічного проекту – ТП.
3. Робочого проекту – РП.
4. Впровадження – ВН.

$$T_{mz} = T_r \cdot L_1 \cdot K_n; \quad (4.1)$$

$$T_{tp} = T_r \cdot L_2 \cdot K_n; \quad (4.2)$$

$$T_{rp} = T_r \cdot L_3 \cdot K_n \cdot K_t; \quad (4.3)$$

$$T_{un} = T_r \cdot L_4 \cdot K_n. \quad (4.4)$$

де: T_r – укрупнена норма часу на розробку аналога ПП, люд.-годин, коректується поправочним коефіцієнтом K_k , який враховує умови розробки ПП;

$$K_k = 0.9.$$

$$T_r = 200 \times 0.9 = 90 \text{ люд.-год};$$

Цей проект має доступні аналоги, тому його можна віднести до ступеня новизни: В.

L_j – питома вага i -го етапу розробки в залежності від ступеня новизни:

$$L_1 = 0,10;$$

$$L_2 = 0,50;$$

$$L_3 = 0,25;$$

$$L_4 = 0,15.$$

K_n – поправочний коефіцієнт, який враховує ступінь новизни (0.8);

K_t – поправочний коефіцієнт, який враховує ступінь використання програм в розробці (0.7).

Тоді:

$$T_{tz} = 90 \cdot 0,10 \cdot 0,8 = 7,2(\text{дні})$$

$$T_{tz} = 90 \cdot 0,50 \cdot 0,8 = 36(\text{дні})$$

$$T_{tz} = 90 \cdot 0,25 \cdot 0,8 \cdot 0,7 = 12,6(\text{дні})$$

$$T_{un} = 90 \cdot 0,15 \cdot 0,8 = 10,8(\text{дні})$$

Сумарна тривалість розробки ПП визначається за формулою 4.5

$$T_{PP} = T_{tz} + T_{tp} + T_{rp} + T_{un} \quad (4.5)$$

$$T_{PP} = 7,2 + 36 + 12,6 + 10,8 = 66,6$$

За час розробки візьмемо середнє значення розрахованих показників тривалості проектування та розробки ПП:

$$T_{PP} = \frac{66,6}{2} = 33,3(\text{дні})$$

Таблиця 4.6

Розрахунок витрат на програмне забезпечення

| Найменування Програмного забезпечення | Кількість | Ціна за одиницю, грн. | Вартість, грн |
|---------------------------------------|-----------|-----------------------|---------------|
| <i>Unreal Engine 5</i> | 1 | Безкоштовно | 0 |
| <i>Blender</i> | 1 | Безкоштовно | 0 |
| <i>Adobe Substance Painter</i> | 1 | 733/міс | 8800/рік |
| <i>Adobe Photoshop</i> | 1 | 880/міс | 10560/рік |
| Усього | | 1613/міс | 19360/рік |

Розрахунок основної заробітної плати

| Найменування робіт | Трудомісткість робіт у днях | Місячний оклад | Денна заробітна плата | Заробітна плата |
|--------------------|--------------------------------|-------------------|--------------------------|--------------------|
| Розробка моделі | 25 | 15000 | 682 | 17050 |
| Усього | 25 | – | – | 17050 |

Додаткова заробітна плата враховує оплату чергових відпусток, премії, інші доплати. Приймається в розрахунках 10% від основної.

$$ЗП_{\text{дод}} = ЗП_{\text{осн}} \cdot 0,10 \quad (4.6)$$

$$ЗП_{\text{дод}} = 17050 \cdot 0,10 = 1705 \text{ грн.}$$

Єдиний соціальний внесок (Єсв) приймаються в розмірі 22% від суми основної і додаткової заробітної плати.

$$Єсв = (ЗП_{\text{осн}} + ЗП_{\text{дод}}) \cdot 0,22 \quad (4.7)$$

$$Єсв = (17050 + 1705) \cdot 0,22 = 4135,1 \text{ грн.}$$

Витрати, зв'язані з використанням обчислювальної техніки, визначаються:

$$C_{\text{еом}} = t^{\text{еом}} \cdot K_{\text{евм}}^{\text{еом}} \cdot Ц_{\text{свм}}^{\text{еом}} \cdot K_{\text{евм}}^{\text{е}}; \quad (4.8)$$

де: $T_{\text{евм}}$ – 160 годин;

$K_{\text{евм}}$ – поправочний коефіцієнт обліку часу використання ЕОМ 1,08;

$Ц_{\text{свм}}$ – ціна однієї години роботи на ЕОМ, грн 5 грн;

$K_{\text{евм}}$ – 1,0.

$$C_{\text{еом}} = 160 \cdot 1,08 \cdot 5 \cdot 1,0 = 864 \text{ грн}$$

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 90 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

Накладні витрати враховують адміністративні, загальнопромислові витрати, витрати на збут. Приймаються в розмірі 50% від основної заробітної плати.

$$H_B = 0,50 \cdot ЗП_{осн} \quad (4.9)$$

$$H_B = 0,50 \cdot 17050 = 8\,525 \text{ грн}$$

На підставі здійснених розрахунків складається калькуляція планової собівартості ПП.

Таблиця 4.8

Калькуляція собівартості

| Найменування статей витрат | Сума витрат (грн.) | Питома вага, % |
|---|--------------------|----------------|
| Програмне забезпечення | 19360 | 32,1 |
| Основна заробітна плата | 17050 | 28,3 |
| Додаткова заробітна плата | 1705 | 2,8 |
| Єдиний соціальний внесок | 4135,1 | 6,9 |
| Витрати, зв'язані з обчислювальною технікою | 864 | 1,4 |
| Накладні витрати | 8525 | 14,1 |
| Разом: | 50139,1 | 100 |

Ціна ПП визначається по формулі:

$$Ц = C + П_p, \quad (4.10)$$

де: C – витрати на розробку програмної продукції, грн;

$П_p$ – розмір прибутку.

Розрахований по формулі:

$$П_p = C \cdot \%P_H / 100 \quad (4.11)$$

де: P_H – плановий рівень рентабельності (25%).

Прибуток у ціні ПП становить:

| | | | | | | |
|------|------|----------|--------|-----|-----------------------------|------|
| | | | | | <i>КРБ.КІ.1.442-03.1.10</i> | Арк. |
| | | | | | | 91 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

$$\Pi = 50139,1 \cdot 0,25 = 12534,78 \text{ грн}$$

Ціна ПП становить: $\text{Ц} = 50139,1 \cdot 1,1 + 12534,78 = 62673,88 \text{ грн}$

4.2.1 Розрахунок капітальних витрат

Розрахунок капітальних витрат, зв'язаних із впровадженням програмного продукту, здійснюється за формулою:

$$K_2 = K_{\text{пп}} + K_{\text{п}} + K_{\text{ко}} + K_{\text{во}} \quad (4.12)$$

де: $K_{\text{пп}}$ – ціна програмного продукту;

$K_{\text{п}}$ – поперед виробничі витрати;

$K_{\text{ко}}$ – вартість комп'ютерного устаткування;

$K_{\text{во}}$ – вартість допоміжного устаткування, необхідного для надійної роботи грн.

Ціна програмного продукту становить – 62,673.88 грн.

Попередвиробничі витрати містять у собі усі витрати, зв'язані з налагодженням і впровадженням сцени і її моделі – постановка задач та їхня алгоритмізація, розробка, налагодження і впровадження програмного продукту (ПП).

Приймаються $K_{\text{п}}$ у розмірі 100% від вартості розробленого ПП.

$$K_{\text{п}} = 62\,433 \text{ грн.}$$

Вартість комп'ютера становить 24,000 грн.

Вартість допоміжного устаткування визначається укрупнено в розмірі 10% від вартості комп'ютера.

$$K_{\text{во}} = 24\,000 \cdot 0,10 = 2\,400 \text{ грн.}$$

Розрахунок загальних капітальних витрат.

$$K_2 = 62\,673,88 + 24\,000 + 2\,400 = 89\,073,88 \text{ грн.}$$

| | | | | | | |
|------|------|----------|--------|-----|-----------------------------|------|
| | | | | | <i>КРБ.КІ.1.442-03.1.10</i> | Арк. |
| | | | | | | 92 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

4.2.2 Розрахунок показників економічної ефективності проекту

Очікуваний економічний ефект визначається за формулою:

$$E_o = (C_1 - C_2) - E_n \cdot (K_2 - K_1) \quad (4.13)$$

де: C_1, C_2 – поточні витрати відповідно до і після впровадження проекту; $(C_1 - C_2)$ – річна економія на поточних витратах, грн;

K_2 – капітальні витрати на впровадження ПП, грн;

K_1 – капітальні витрати до впровадження ПП, грн;

E_n – нормативний коефіцієнт ефективності одноразових витрат (рекомендовано $E_n = 0,25$).

Припустимо, що капітальні витрати до впровадження ПП (K_1) дорівнюють нулю.

Припустимо, що:

1. Поточні витрати після впровадження проекту (C_2) дорівнюють 29,501 грн/рік.

2. Поточні витрати до впровадження проекту (C_1) були 61,701.88 грн/рік (щоб забезпечити відповідну економію).

$$K_2 = 89,073.88 \text{ грн}$$

Розрахунок економічного ефекту:

$$E_o = (61,701.88 - 29,501) - 0.25 \cdot 89,073.88 = 9,932.41 \text{ грн}$$

Потім розраховується коефіцієнт ефективності капітальних витрат по формулі:

$$E = (C_1 - C_2) / K_2 - K_1 \quad (4.14)$$

$$E = 32\,200.88 / 89,073.88 = 0.362$$

Так, як $E > E_n$, то проект є ефективним.

Розраховується строк окупності капітальних витрат на впровадження проекту.

$$T = 1 / E = 1 / 0,362 = 2,76 \text{ року} \quad (4.15)$$

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 93 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

Висновки до четвертого розділу

1. Цей проект має потенціал створення не тільки однієї сцени, а й розроблення серії подібних сцен на *Unreal Engine 5*. Це відкриває перспективи для створення і подальшого продажу різних сцен іншим розробникам, студіям або ігровим компаніям. Можливість створення та продажу додаткових тематичних пакетів, а також надання послуг з кастомізації сцен під індивідуальні потреби клієнтів може значно розширити можливості для додаткового доходу в майбутньому.

2. Розробили й описали бізнес-план для проекту створення унікальної сцени на *Unreal Engine 5*.

3. Очікуваний економічний ефект від реалізації проекту становить 9,932.41 грн на рік, що свідчить про його прибутковість. Коефіцієнт ефективності капітальних витрат становить 0.362, що перевищує нормативний коефіцієнт ефективності одноразових витрат (0.25), підтверджуючи економічну доцільність проекту. Строк окупності капітальних витрат на впровадження проекту становить 2.76 роки, що є прийнятним показником для інвестиційного проекту.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 96 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

РОЗДІЛ 5 ОХОРОНА ПРАЦІ

5.1 Основні положення охорони праці

Охорона праці – це комплекс заходів і систем, включаючи правові, організаційно-технічні, соціально-економічні, санітарно-гігієнічні та лікувально-профілактичні заходи. Ці заходи спрямовані на захист життя, здоров'я і працездатності працівників у процесі роботи.

Основна мета охорони праці та техніки безпеки – створити умови праці, які не тільки забезпечують безпеку і здоров'я співробітників, а й сприяють максимальній продуктивності та задоволеності роботою. Основна мета управління охороною праці - це діяльність різних відділів, служб і груп підприємства, спрямована на забезпечення безпечних і здорових умов праці.

Це включає розроблення та реалізацію заходів щодо запобігання нещасним випадкам на виробництві та професійним захворюванням, поліпшення умов праці та підвищення культури безпеки на робочому місці.

Основними елементами системи захисту праці є:

1. Правові заходи – включає розроблення та дотримання правил, стандартів і норм, що стосуються безпеки праці.

2. Організаційно-технічні заходи – включають проектування та експлуатацію безпечного обладнання, розроблення безпечних технологій, організацію робочих місць.

3. Соціально-економічні заходи – заохочення та компенсації за важкі та шкідливі умови праці, надання соціального страхування працівникам.

4. Санітарно-гігієнічні заходи – охоплюють моніторинг стану робочого місця та забезпечення санітарно-гігієнічних умов праці.

5. Лікувально-профілактичні заходи – включає медичне обстеження, організацію профілактичних заходів, спрямованих на зниження ризику захворювання.

| | | | | | | |
|------|------|----------|--------|-----|-----------------------------|------|
| | | | | | <i>КРБ.КІ.1.442-03.1.10</i> | Арк. |
| | | | | | | 97 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

У контексті розробки комп'ютерних ігор питання безпеки праці набули певних особливостей, пов'язаних з особливостями цієї галузі.

Розробка ігор зазвичай передбачає тривалу роботу за комп'ютером, що може призвести до проблем із зором, опорно-руховим апаратом і психічним здоров'ям через інтенсивну роботу над проектами з високим рівнем стресу та стислими термінами.

5.2 Недоліки та умови роботи за комп'ютером

Робота за комп'ютером має багато переваг, але також пов'язана з багатьма недоліками та умовами, які можуть негативно вплинути на здоров'я співробітників.

До основних недоліків та умов експлуатації комп'ютера належать:

1. Напруга очей – постійне перебування перед екраном призводить до стомлення очей, зниження зору, сухості очей і головного болю. Необхідність регулярних перерв для відпочинку очей і використання спеціальних програм і налаштувань для зниження зорового навантаження.

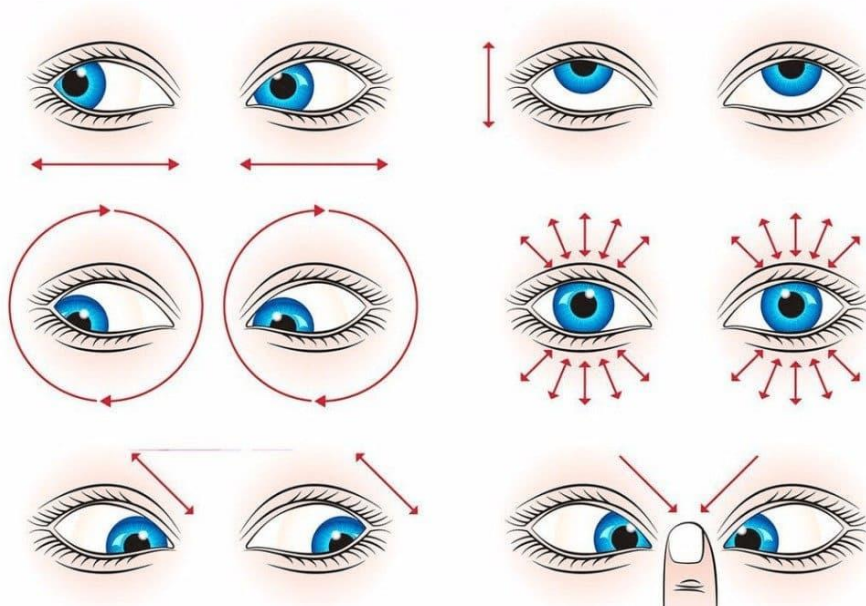


Рис 5.1 – Перелік вправ для очей

2. Проблеми з осанкою та опорно-руховим апаратом – тривале сидіння в незручному положенні може спричинити біль у спині, шиї, плечах і проблеми з хребтом. Треба забезпечити правильну ергономіку робочого місця, використання регульованих стільців і столів, регулярні вправи для спини та шиї.

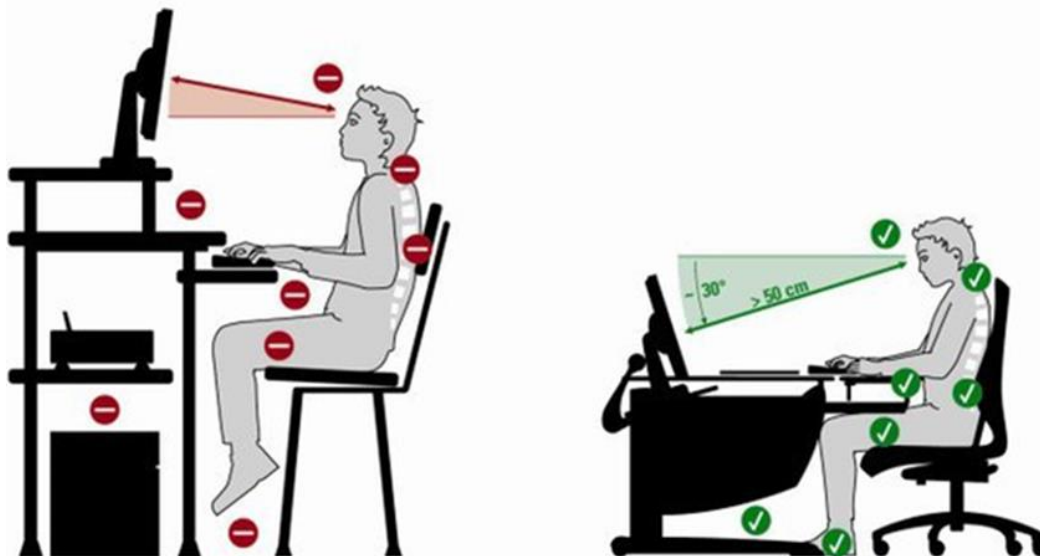


Рис 5.2 – Правильне робоче місце

3. Психічне напруження і стрес – інтенсивна робота над проектами зі стислими термінами може призвести до вигорання, стресу та інших психічних проблем. Необхідно збалансувати робочий та особистий час, організувати психологічну підтримку і провести заходи щодо зниження стресу.

4. Зниження фізичної активності малорухливий спосіб життя – збільшує ризик серцево-судинних захворювань, ожиріння та інших проблем зі здоров'ям. Необхідно ввести перерву на фізичну активність і створення умов для занять спортом.

5. Ризик розвитку синдрому зап'ястного каналу – часта робота з клавіатурою і мишею може призвести до болю й оніміння в руках і пальцях. Використання ергономічних клавіатур і мишей, виконання вправ для рук і кистей

5.3 Електробезпека

Електробезпека – це система організаційно-технічних заходів, спрямованих на захист людей від потенційно шкідливого та небезпечного впливу

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дат | | 99 |

електричних струмів, дуг, електромагнітних полів і статичної електрики. Забезпечення електробезпеки на підприємстві є дуже важливим аспектом охорони праці, важливо дотримуватися вимог, викладених у відповідних законах законодавства України.

Основні нормативні акти, що регулюють питання електробезпеки на підприємстві, включають:

1. Правила безпечної експлуатації електроустановок споживачів (наказ Держнаглядохоронпраці від 09.01.1998 № 4). Ці правила визначають вимоги до безпечної експлуатації електроустановок для всіх споживачів електроенергії. Вони містять положення щодо організації робіт, технічного обслуговування та профілактики аварій.

2. Правила безпечної експлуатації електроустановок (наказ Держнаглядохоронпраці України від 06.10.1997 № 257). Цей документ поширюється на працівників, які працюють з електроустановками Міністерства енергетики України, і включає вимоги щодо безпечної роботи та заходів з попередження нещасних випадків.

3. Правила технічної експлуатації електроустановок споживачів (наказ Мінпаливенерго України від 25.07.2006 № 258, зі змінами). Встановлюють технічні вимоги до експлуатації електроустановок, регламентуючи порядок їх обслуговування, ремонту та модернізації.

4. Правила експлуатації електрозахисних засобів (наказ Міністерства праці та соціальної політики України від 05.06.2001 № 253). Цей наказ регламентує вимоги до застосування засобів захисту від електричних ризиків, включаючи діелектричні рукавички, килимки та інші захисні пристосування.

5. Правила улаштування електроустановок (наказ Міністерства енергетики та вугільної промисловості України від 24.07.2017 № 476). Визначають вимоги до проектування, монтажу та введення в експлуатацію електроустановок, забезпечуючи їх безпечність та надійність.

6. ДСТУ 2843-94 "Електротехніка. Основні поняття. Терміни та визначення". Цей стандарт встановлює терміни та визначення основних понять

| | | | | | | |
|------|------|----------|--------|-----|-----------------------------|------|
| | | | | | <i>КРБ.КІ.1.442-03.1.10</i> | Арк. |
| | | | | | | 100 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

у галузі електробезпеки, що дозволяє уніфікувати підхід до забезпечення електробезпеки на підприємствах.

5.4 Пожежна безпека при роботі з комп'ютером

Пожежна безпека при роботі з комп'ютерним обладнанням є важливою складовою загальної системи охорони праці та безпеки на підприємстві. Використання електронних пристроїв, таких як комп'ютери, може стати причиною пожеж, якщо не дотримуватися відповідних правил та норм.

Основні нормативні акти, що регулюють пожежну безпеку при роботі з комп'ютерами, включають:

1. Правила пожежної безпеки в Україні (НАПБ А.01.001-2014). Затверджені наказом МВС України від 30.12.2014 № 1417. Ці правила регулюють порядок забезпечення пожежної безпеки на об'єктах, включаючи офіси та приміщення з комп'ютерним обладнанням.

2. Наказ Міністерства внутрішніх справ України від 15.06.2015 № 677 "Про затвердження Правил пожежної безпеки для підприємств, установ та організацій України". Цей наказ встановлює вимоги до забезпечення пожежної безпеки, включаючи організацію роботи з електронним обладнанням.

3. Правила експлуатації електроустановок споживачів (ПЕЕ). Містять вимоги до експлуатації електроустановок і електрообладнання, включаючи комп'ютери та периферійні пристрої.

Основні вимоги та заходи для забезпечення пожежної безпеки при роботі з комп'ютерами включають:

1. Правильна організація робочого місця – забезпечення достатньої вентиляції для комп'ютерного обладнання для запобігання його перегріву и використання сертифікованих подовжувачів та розеток для підключення комп'ютерної техніки.

2. Регулярні перевірки та обслуговування – проведення регулярних перевірок електропроводки та стану електрообладнання. Своєчасне

| | | | | | | |
|------|------|----------|--------|-----|-----------------------------|------|
| | | | | | <i>КРБ.КІ.1.442-03.1.10</i> | Арк. |
| | | | | | | 101 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

обслуговування комп'ютерного обладнання для запобігання несправностей, які можуть призвести до пожежі.

3. Використання сертифікованих засобів захисту – використання пристроїв захисту від перенапруги, таких як стабілізатори напруги та мережеві фільтри, для захисту комп'ютерної техніки.

4. Інструктажі та навчання працівників – регулярне проведення інструктажів та навчання працівників щодо правил пожежної безпеки та дій у разі виникнення пожежі.

5. Ознайомлення працівників з розташуванням первинних засобів пожежогасіння, таких як вогнегасники, і способами їх використання.

6. Заборона на залишення увімкненого комп'ютерного обладнання без нагляду на тривалий час.

7. Дотримання правил безпеки при використанні електрообладнання, включаючи уникнення перевантаження електромережі.

8. Розробка і впровадження планів евакуації та дій на випадок пожежі – створення планів евакуації та регулярне проведення тренувальних евакуацій для працівників.

9. Забезпечення приміщень з комп'ютерним обладнанням засобами зв'язку для швидкого виклику пожежної служби у разі необхідності.

10. Забезпечення приміщень вогнегасниками – приміщення з комп'ютерним обладнанням повинні бути оснащені вуглекислотними (CO₂) або порошкові вогнегасники. Вони ефективні для гасіння пожеж класу E (електрообладнання під напругою).

Висновки до п'ятого розділу

Було розглянуто питання, пов'язані з охороною праці на місці, де відбувається безпосереднє виконання роботи, включаючи розробку гри та умови праці. Необхідні норми пожежної безпеки при роботі з комп'ютером також були визначені.

| | | | | | | |
|------|------|----------|--------|-----|-----------------------------|------|
| | | | | | <i>КРБ.КІ.1.442-03.1.10</i> | Арк. |
| | | | | | | 102 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

ЗАГАЛЬНІ ВИСНОВКИ

1. У результаті аналізу було досліджені віртуальні світи. Зокрема поняття класифікації, інструменти, способи взаємодії та приклади використання.

2. На підставі дослідження була зроблена постановка завдань.

3. Розроблено пайплайн и описано порядок та склад робіт.

4. Розглянуті технології створення об'єктів.

5. Розглянуті технології створення віртуальних світів у *Unreal Engine*.

6. Була спроектована система віртуального світу.

7. В ході виконання було досягнуто успішного створення високоякісної тривимірної моделі меча, що інтегрована в реалістичну сцену за допомогою сучасних інструментів і методів. Процес моделювання в *Blender* включав інструменти самої програми. Також, розглянуто використання *Smart UV-Project* для ефективною *UV*-розгортки, що забезпечило високу якість текстур. Подальше текстурування у *Substance Painter* дозволило створити детальні і реалістичні текстури, використовуючи пресет *Unreal Engine 4* і роздільну здатність 2048x2048 пікселів.

8. Імпорт моделі в *Unreal Engine* супроводжувався налаштуванням параметрів масштабу, типу мешу, імпорту матеріалів і текстур, що забезпечило коректне відображення моделі у сцені. Створення ефектів за допомогою системи частинок *Niagara*, таких як дощ, а також застосування *Blueprint* для створення туману, підсилили реалістичність і атмосферність сцени.

9. Використання просунутих технік освітлення в *Unreal Engine*, таких як *Direct Light* для симуляції місячного світла і *Point Light* для підсвічування ключових елементів, додало сцені глибини і виразності. Крім того, налаштування постобробки, включаючи глибину різкості, кольорокорекцію і тональний маппінг, дозволили створити необхідний настрій і посилити візуальний ефект.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 103 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

10. Процес створення відео, що включав анімацію камери і налаштування її параметрів, таких як фокусна відстань і діафрагма, забезпечив динамічність і привабливість фінального результату. Візуалізація сцени з використанням різних технік освітлення і постобробки продемонструвала можливості сучасних інструментів для досягнення високого рівня деталізації та реалістичності.

11. Цей проект має потенціал створення не тільки однієї сцени, а й розроблення серії подібних сцен на *Unreal Engine 5*. Це відкриває перспективи для створення і подальшого продажу різних сцен іншим розробникам, студіям або ігровим компаніям. Можливість створення та продажу додаткових тематичних пакетів, а також надання послуг з кастомізації сцен під індивідуальні потреби клієнтів може значно розширити можливості для додаткового доходу в майбутньому.

12. Розробили й описали бізнес-план для проекту створення унікальної сцени на *Unreal Engine 5*.

13. Очікуваний економічний ефект від реалізації проекту становить 9,932.41 грн на рік, що свідчить про його прибутковість. Коефіцієнт ефективності капітальних витрат становить 0.362, що перевищує нормативний коефіцієнт ефективності одноразових витрат (0.25), підтверджуючи економічну доцільність проекту. Строк окупності капітальних витрат на впровадження проекту становить 2.76 роки, що є прийнятним показником для інвестиційного проекту.

14. Було розглянуто питання, пов'язані з охороною праці на місці, де відбувається безпосереднє виконання роботи, включаючи розробку гри та умови праці. Необхідні норми пожежної безпеки при роботі з комп'ютером також були визначені.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| | | | | | | 104 |
| Змн. | Арк. | № докум. | Підпис | Дат | | |

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДСТУ для оформлення дипломних (курскових) проектів та дисертацій в університеті. – URL: <https://library.ontu.edu.ua/standards> (дата звернення 21.03.2024).
2. Селіванова А. В. Методичні вказівки до виконання кваліфікаційних робіт для студентів СВО «Магістр» спеціальності 122 «Комп'ютерні науки» / А.В. Селіванова, О.В. Ольшевська. Одеса, ОНТУ, 2022. 43 с.
3. Epic Games. (2023). *Unreal Engine 5 Documentation*. – URL: <https://docs.unrealengine.com/5.0/en-US/> (дата звернення 24.03.2024).
4. Офіційна документація «Blender» – URL: <https://docs.blender.org/manual/en/latest/> (дата звернення 20.03.2024).
5. Офіційна документація «Substance 3D Painter» – URL: <https://substance3d.adobe.com/documentation/spdoc/substance-3d-painter-2021-4-documentation-198975775.html> (дата звернення 21.03.2024).
6. ArtStation – URL: <https://www.artstation.com/> (дата звернення 19.03.2024).
7. Вільна енциклопедія «Вікіпедія». – URL: <https://uk.wikipedia.org/wiki/>
8. Коваль, В. В. Основи 3D-моделювання. Київ, Кондор, 2019. 256 с.
9. Петренко, І. І. Тривимірне моделювання у середовищі *Blender*. Львів, Світ, 2018. 280 с.
10. Домаскіна, М. ІНФОРМАТИКА 10-11 кл. Тривимірне моделювання. Посібник. Ранок, 2020. 176 с.
11. Андрій Мельник, Текстурування в «Substance Painter»: Практичне керівництво – Літера ЛТД, 2021. 280 с.
12. Олексій Ковальчук, Технології 3D текстурування: від основ до професійного рівня – Видавничий дім "Академія", 2022. 320 с.
13. Богачук О. В. Розробка ігор на *Unreal Engine 5*. Київ, Видавничий дім "Освіта", 2022. 280 с.
14. Ковальчук М. І. Створення віртуальних світів з *Unreal Engine 5*. Харків

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дат | | 105 |

Видавництво "Фоліо", 2023. 320 с.

15. Доброскок, С. В. Сучасні технології тривимірного моделювання. Київ, Ліра К, 2023. 260 с.
16. Ray, D. (2019). *Learning Unreal Engine Game Development*. Packt Publishing. 358 p.
17. Сергієнко, П.Г. Основи моделювання в 3D середовищі. Харків, ХНУРЕ, 2020. 280 с.
18. Тараненко, Ю. І. 3D графіка та анімація в «Blender». Львів, Видавництво Львівської політехніки, 2022. 320 с.
19. Vasquez, O. (2021). *Creating Games with Unreal Engine 5*. Packt Publishing. 480 с.
20. Kaufman, A. E., & Mueller, K. (2019). *Principles of 3D Computer Graphics: Modeling, Rendering, and Animation*. Springer. 416 с.
21. «Sketchfab» – URL: <https://sketchfab.com/> (дата звернення 22.03.2024).
22. «Turbosquid» – URL: <https://www.turbosquid.com/> (дата звернення 27.03.2024).
23. «Blender Artists» – URL: <https://blenderartists.org/> (дата звернення 29.03.2024).
24. Нестеренко П.О. Віртуальні світи та доповнена реальність. Київ, Наукова думка, 2022. 245 с.
25. Мироненко В. Г. Тривимірна графіка та моделювання. Київ, Либідь, 2018. 275 с.

| | | | | | | |
|------|------|----------|--------|-----|----------------------|------|
| | | | | | КРБ.КІ.1.442-03.1.10 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дат | | 106 |