

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ПОЛІСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій, обліку та фінансів  
Кафедра комп'ютерних технологій  
і моделювання систем

Кваліфікаційна робота  
на правах рукопису

Ємець Даніель Вадимович

УДК 004.68.21:68.13.1

## **КВАЛІФІКАЦІЙНА РОБОТА**

**Розробка системи скриптів для анімації 3D-моделей у відеоогрі**  
122 «Комп'ютерні науки»

Подається на здобуття освітнього ступеня бакалавр

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

---

(підпис, ініціали та прізвище здобувача вищої освіти)

Керівник роботи  
Тимонін Юрій Олександрович,  
Кандидат технічних наук, доцент кафедри КТіМС

**Висновок кафедри**

---

за результатами попереднього захисту:

---

Протокол засідання кафедри

---

№ \_\_\_\_\_ від «\_\_\_\_\_» \_\_\_\_\_ 20\_\_\_\_ р.

Завідувач кафедри

---

\_\_\_\_\_ (науковий ступінь, вчене звання) \_\_\_\_\_ (підпис) \_\_\_\_\_ (прізвище, ім'я, по батькові)

«\_\_\_\_\_» \_\_\_\_\_ 20\_\_\_\_ р.

**Результати захисту кваліфікаційної роботи**

Здобувач вищої освіти \_\_\_\_\_ захистив (ла)  
(прізвище, ім'я, по батькові)

кваліфікаційну роботу з оцінкою:

сума балів за 100-бальною шкалою \_\_\_\_\_

за шкалою ECTS \_\_\_\_\_

за національною шкалою \_\_\_\_\_

Секретар ЕК

---

\_\_\_\_\_ (науковий ступінь, вчене звання) \_\_\_\_\_ (підпис) \_\_\_\_\_ (прізвище, ім'я, по батькові)

## АНОТАЦІЯ

**Ємець Д.В. Розробка системи скриптів для анімації 3D-моделей у відеоігрі – кваліфікаційна робота на правах рукопису.**

*Кваліфікаційна робота на здобуття освітнього ступеня бакалавра за спеціальністю 122 – Комп'ютерні науки. – Поліський національний університет, Житомир, 2024.*

Ця кваліфікаційна робота присвячена розробці системи скриптів для анімації 3D-моделей у відеоігрі. Мета кваліфікаційної роботи присвячена розробці системи скриптів для анімації 3D-моделей у відеоігрі, яка дозволить аніматорам та геймдизайнерам легко створювати реалістичні та динамічні анімації для різних ігрових персонажів та об'єктів. В рамках роботи були виконані наступні завдання: проаналізовано предметну область дослідження та визначено вимоги до розробки системи скриптів; розроблено концептуальну модель системи скриптів; спроектовано прототип системи скриптів; протестовано розроблену систему скриптів.

Результатом роботи стала система скриптів, яка дозволяє: створювати реалістичні та динамічні анімації 3D-моделей; легко керувати анімаціями в реальному часі; використовувати різні методи анімації; інтегрувати анімації з геймплеєм.

Розроблена система скриптів може бути використана для підвищення ефективності розробки анімацій 3D-моделей, зменшення часу та зусиль, необхідних для анімування 3D-моделей, а також для підвищення продуктивності комп'ютерних ігор, які використовують 3D-моделі.

**Ключові слова:** анімація 3D-моделей, система скриптів, відеоігри, реалістичні анімації, динамічні анімації, геймплей.

## SUMMARY

**Yemets D.V. Development of a system of scripts for the animation of 3D models in a video game - qualifying work on the rights of the manuscript.**

*Qualification work for the degree a Bachelor in specialty 122 - Computer science. – Poliss National University, Zhytomyr, 2024.*

This qualification work is dedicated to the development of a scripting system for animating 3D models in a video game. The aim of the qualification work is to develop a scripting system for animating 3D models in a video game, which will allow animators and game designers to easily create realistic and dynamic animations for various game characters and objects. The following tasks were accomplished as part of the work: the research domain was analyzed, and the requirements for the development of the scripting system were defined; a conceptual model of the scripting system was developed; a prototype of the scripting system was designed; the developed scripting system was tested.

The result of the work is a scripting system that allows: creating realistic and dynamic animations of 3D models; easily controlling animations in real-time; using various animation methods; integrating animations with gameplay.

The developed scripting system can be used to increase the efficiency of 3D model animation development, reduce the time and effort required for animating 3D models, and enhance the performance of computer games that use 3D models.

**Keywords:** 3D model animation, scripting system, video games, realistic animations, dynamic animations, gameplay.

## ЗМІСТ

<b>ВСТУП.....</b>	<b>6</b>
<b>РОЗДІЛ 1. ТЕОРЕТИЧНИЙ АНАЛІЗ ОСОБЛИВОСТЕЙ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ .....</b>	<b>8</b>
<b>1.1 Аналіз інформаційних потреб і визначення предметної області дослідження .....</b>	<b>8</b>
<b>1.2. Аналіз вимог до системи скриптів для анімації 3D-моделей .....</b>	<b>10</b>
<b>1.3 Вибір технічного забезпечення для розробки та оптимізації 3D-моделей для комп'ютерних ігор .....</b>	<b>11</b>
<b>Висновки до першого розділу .....</b>	<b>13</b>
<b>РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ СКРИПТІВ ДЛЯ АНІМАЦІЇ 3D-МОДЕЛЕЙ У ВІДЕОГРІ.....</b>	<b>14</b>
<b>2.1 Моделювання бізнес-процесів предметної області.....</b>	<b>14</b>
<b>2.2. Моделювання інформаційної системи розробки та оптимізації 3D-моделей для комп'ютерних ігор з використанням сучасних технологій .....</b>	<b>15</b>
<b>2.3 Процес реалізації системи скриптів для анімації 3D-моделей у відеогрі.....</b>	<b>18</b>
<b>Висновок до другого розділу .....</b>	<b>21</b>
<b>РОЗДІЛ 3. ТЕСТУВАННЯ ГРІ ТА АНАЛІЗ РЕЗУЛЬТАТІВ .....</b>	<b>22</b>
<b>3.1 Інструкція користувачу інформаційної системи розробка та оптимізація 3D-моделей для комп'ютерних ігор з використанням сучасних технологій.....</b>	<b>22</b>
<b>3.2 Тестування системи скриптів для анімації 3D-моделей та аналіз результатів.....</b>	<b>23</b>
<b>Висновок до третього розділу .....</b>	<b>24</b>
<b>ЗАГАЛЬНІ ВИСНОВОКИ .....</b>	<b>25</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>27</b>
<b>ДОДАТКИ.....</b>	<b>30</b>

## ВСТУП

**Актуальність.** У сучасних відеоіграх 3D-моделі відіграють ключову роль у створенні реалістичного та захоплюючого візуального досвіду. Їх анімація є невід'ємною частиною геймплею, емоційної атмосфери та загального враження від гри. Розробка ефективних систем анімації 3D-моделей стає все більш важливою для розробників ігор, адже вона значно впливає на якість та успішність продукту.

**Мета кваліфікаційно роботи** присвячена розробці системи скриптів для анімації 3D-моделей у відеоіграх, яка дозволить аніматорам та геймдизайнерам легко створювати реалістичні та динамічні анімації для різних ігрових персонажів та об'єктів.

Виходячи, з вище означеної мети, перед нами постає ряд **завдань**:

- 1) проаналізувати предметну область дослідження;
- 2) визначити вимоги до розробок такого типу;
- 3) розробити концептуальну модель системи скриптів;
- 4) спроектувати прототип системи скриптів;
- 5) протестувати розроблену систему скриптів.

**Предмет дослідження** є процес анімації 3D-моделей у відеоіграх.

**Об'єктом дослідження** є система скриптів для анімації 3D-моделей, розроблена в рамках цієї кваліфікаційної роботи.

У процесі дослідження були використані наступні **методи дослідження**: *аналізу* (для дослідження предметної області та визначення вимог до системи скриптів); *моделювання* (для розробки концептуальної моделі системи скриптів та створення прототипу системи скриптів); *тестування* (для оцінки працездатності та ефективності системи скриптів).

**Практичне значення досліджуваної теми** полягає в тому, що результати дослідження можуть бути використані для підвищення ефективності розробки системи скриптів для анімації 3D-моделей у відеоіграх; зменшення часу та зусиль, необхідних для анімування 3D-моделей;

підвищення продуктивності комп'ютерних ігор, які використовують 3D-моделі.

**Теоретичне значення досліджуваної теми** полягає в тому, що результати дослідження можуть бути використані для розвитку теорії розробки системи скриптів для анімації 3D-моделей у відеогрі, розробки нових методів і алгоритмів, уточнення існуючих теорій. Розширення області застосування теорії розробки системи скриптів для анімації 3D-моделей у відеогрі. **Унікальністю** кваліфікаційної роботи є розроблена система скриптів для анімації 3D-моделей.

**Результати роботи апробовано** на двох науково-практичних конференціях та опубліковано тези «Аналіз можливостей Blender як інструменту для створення, моделювання та візуалізації тривимірних об'єктів» та «Різноманітність та вплив застосування 3d-моделювання в різних галузях».

**Структура та обсяг роботи.** Кваліфікаційна робота складається з вступу, трьох розділів, висновку та списку використаних джерел. Загальний обсяг роботи становить 45 сторінок.

## РОЗДІЛ 1. ТЕОРЕТИЧНИЙ АНАЛІЗ ОСОБЛИВОСТЕЙ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ

### 1.1 Аналіз інформаційних потреб і визначення предметної області дослідження

3D-моделювання – це процес розробки математичного представлення будь-якої тривимірної поверхні за допомогою програмного забезпечення. Цей метод використовується в різних галузях, таких як комп'ютерні ігри, кіноіндустрія, архітектура, медицина, наука та багато інших. Предметна область даного проекту полягає в розробці та оптимізації 3d-моделей для комп'ютерних ігор, створенні текстур та простих анімацій. Розробка моделей буде зосереджена на двох іграх в жанрі фентезі але в різних часових проміжках тому моделі будуть різноманітними.

На меті зробити моделі високої якості с готовими для них текстурами та анімаціями для подальшого їх імпортування для моделювання та візуалізації 3D-сцени в ігровому рушії.

Більшість розробок по моделям та анімаціям буде проводитись у програмі Blender.

Blender – «це вільний та відкритий програмний продукт для 3D-моделювання, анімації, композитингу, рендерингу, обробки звуку та відео. Він надає широкий спектр можливостей для творчості у галузі комп'ютерної графіки та візуальних ефектів» [1][2]

Потенційні аналоги для Blender див табл. 1.1. у додатку А

Autodesk Maya – програма для 3D-моделювання, яка широкого використовується в галузі кіно та відеоігор, оскільки має потужний набір функцій для створення складних анімацій, ригінгу, симуляції та рендерингу [3][4].

Cinema 4D – програмний пакет для створення тривимірної графіки та анімації, розроблюваний Maxon. Cinema 4D є універсальною комплексною



програмою для створення і редагування тривимірних ефектів і об'єктів. Має інтуїтивно зрозумілий інтерфейс, що зробило його серед користувачів, що працюють в галузі реклами, телебачення, оскільки володіє потужними інструментами для створення анімації та спеціальних ефектів [5][6].

AutoDesk 3Ds Max – професійне програмне забезпечення для 3D-моделювання, анімації та візуалізації при створенні ігор і проектуванні. В даний час розробляється і видається компанією. AutoDesk добре підходить для архітектурного моделювання та візуалізації. Він довгий час був флагманом у галузі тривимірного моделювання, особливо у візуальних ефектах, архітектурній візуалізації та геймдеві [7][8].

З таблиці 1.1, конкуренти мають обмежений функціонал та велику ціну на їх продукт, що дає нам підстави використання Blender для розробки 3D-моделей, котрий буде функціональний та буде виконувати більш ширший спектр поставлених задач, з кращим результатом.

Розробка текстур для моделі буде проводитись в Substance Painter. Substance Painter – це програмне забезпечення для текстурного мапування та редагування текстур, яке широко використовується у галузі створення 3D-моделей та візуалізації [9][10].

Потенційні аналоги для Substance Painter див табл. 1.2. у додатку А

Marmoset – це багатофункціональний інструмент для малювання на основі сітки. Потужний режим форми забезпечує повний контроль над багатограними підмережами [11][12].

MARI – це креативний інструмент для малювання текстур, який може працювати з екстремальними проектами. MARI був розроблений у Weta Digital для виконання надзвичайно складної, дуже детальної роботи по створенню зовнішнього вигляду [13].

З таблиці 1.2, Substance Painter є кращим вибором серед програм, тому що має більший спектр можливостей для створення та редагування текстур, можливість модифікування програми для кращих умов використання.

## 1.2. Аналіз вимог до системи скриптів для анімації 3D-моделей

Система скриптів для анімації 3D-моделей повинна відповідати ряду вимог, щоб бути корисною та ефективною. Ось деякі з ключових вимог:

### *Функціональність:*

- Підтримка основних операцій анімації: переміщення, обертання, масштабування, деформація, анімація кісток.
- Підтримка ключових кадрів: можливість встановлювати ключові кадри для анімації параметрів об'єктів з часом.
- Підтримка кривих анімації: можливість використовувати криві анімації для плавного переходу між ключовими кадрами.
- Підтримка ієрархічних структур: можливість анімувати групи об'єктів та скелети.
- Підтримка фізичних симуляцій: можливість використовувати фізичні симуляції для реалістичного руху об'єктів.
- Підтримка ефектів: можливість додавати до анімації візуальні ефекти, такі як освітлення, тіні та частинки.

### *Інтерфейс:*

- Простота використання: система повинна бути інтуїтивно зрозумілою та простою у використанні, навіть для початківців.
- Гнучкість: система повинна пропонувати достатню гнучкість для створення складних анімацій.
- Налаштовуваність: система повинна дозволяти користувачам налаштовувати інтерфейс та робочі процеси.
- Підтримка сценаріїв: система повинна дозволяти користувачам писати сценарії для автоматизації завдань та розширення можливостей системи.

### *Інтеграція:*

- Сумісність з 3D-пакетами: система повинна бути сумісна з популярними 3D-пакетами, такими як Blender, Maya, 3ds Max.

- Підтримка плагінів: система повинна підтримувати плагіни для розширення її можливостей.
- Інтеграція з іншими інструментами: система повинна інтегруватися з іншими інструментами для створення контенту, такими як інструменти для редагування звуку та відео.

*Продуктивність:*

- Швидкість роботи: система повинна бути швидкою та чутливою, щоб користувачі могли працювати з анімацією без затримок.
- Ефективність: система повинна ефективно використовувати ресурси комп'ютера, щоб мінімізувати час рендерингу.
- Підтримка багатоядерних процесорів: система повинна підтримувати багатоядерні процесори для покращення продуктивності.

Виходячи із розглянутих вимог, наша система має бути простою у використанні, щоб користувачі з різним рівнем досвіду могли писати скрипти для анімації. Це може бути досягнуто за допомогою: розробки інтуїтивно зрозумілого інтерфейсу, який полегшує користувачам навігацію та використання системи; надання чіткої та лаконічної документації, яка пояснює функції системи та те, як їх використовувати; надання прикладів, які демонструють, як використовувати систему для створення різних типів анімації.

### **1.3 Вибір технічного забезпечення для розробки та оптимізації 3D-моделей для комп'ютерних ігор**

В залежності від платформи для якої будуть розроблюватись ігри потрібно розробити моделі відповідної якості. Якщо замовника цікавить створення ігор з передовою графікою та захопливим геймплеєм, то обрання ПК може бути відмінним рішенням. Ця платформа надає безліч можливостей для розробників 3D-моделей.

У випадку, якщо важлива простота розробки 3D-моделей, консоль може стати вибором для 3D-розробників. Тут потрібно створювати більш спрощені

та оптимізовані моделі в залежності від консолі. Але з іншого боку, якщо важливо мати можливість створювати та тестувати дуже прості 3D-моделі, то розробка для мобільних пристроїв може бути оптимальним варіантом. Для мобільних застосунків можна виключити деякі процеси розробки 3D-моделей. Для більшої наглядності розглянемо конкретні особливості розробки 3D-моделей для кожної з доступних платформ.

ПК – надає безмежні можливості для творчості. Гнучкість і розширюваність апаратної частини дозволить створювати моделі будь-якої деталізації. Розробники мають доступ до більш різноманітних інструментів та багатих можливостей створення 3D-моделей для комп'ютерних ігор.

Xbox – має найбільш потужну апаратну частину серед консолей. Тому для даної консолі є можливість розробляти моделі високої якості. Microsoft. Також Xbox Game Studios інтегрує 3D-розробників у спільноту, що покращує обмін ідеями та ресурсами [18]

PlayStation – надає можливість для створення унікальних ігор с 3D-моделями [19]. Велика кількість унікальних ексклюзивних ігор дозволяють привертати увагу гравців і підсилюють імідж гри. Партнерство з Sony відкриває можливості для отримання фінансування високобюджетних проектів.

Nintendo Switch – ця платформа може вимагає розробляти та адаптувати 3D-моделі через особливості апаратної частини, але це заважає створювати моделі для унікальних ігор під можливості консолі, такі як портативність та інтерактивність сенсорного екрана [20]. Через свою портативність навіть прості 3D-моделі у грі мають велику популярність.

Мобільні платформи – дозволяє створювати швидко ігри с простими моделями. Але через однотипність проектів та велику кількість донатного контенту, конкуренція в цьому сегменті велика, і розробникам важливо створити унікальні але прості 3D-моделі.

Для проекту було обрано саме ПК, оскільки ця платформа має різноманіття інструментів та можливостей для творчості. Гнучкість і

розширюваність апаратної частини дозволяють максимально реалізувати потенціал, а доступ до ігрових екосистем, зокрема через онлайн-платформи, надає зручний канал для розуміння саме які моделі подобаються гравцям. Вибір ПК став стратегічним рішенням, яке підтримає якість та різноманітність нашого проекту в ігровій індустрії. Гра для якої будуть розробляться 3D-моделі будуть високої якості Ігри на ПК мають певні технічні характеристики для їх запуск.

### **Висновки до першого розділу**

У першому розділі проаналізовано інформаційні потреби і визначено предметну область дослідження, яка полягає в розробці та оптимізації 3D-моделей для комп'ютерних ігор. Також було проаналізовано предметну область для системи скриптів для динамічної анімації

Охарактеризовано потенційних конкурентів інформаційної системи прийняття рішень в розробці та оптимізації 3D-моделей для комп'ютерних ігор. Також було охарактеризовано технічного забезпечення для розробки 3D-моделей.

## РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ СКРИПТІВ ДЛЯ АНІМАЦІЇ 3D-МОДЕЛЕЙ У ВІДЕОГРІ

### 2.1 Моделювання бізнес-процесів предметної області

Загальний вигляд роботи інформаційної системи розробки та оптимізації 3D-моделей для комп'ютерних ігор з використанням сучасних технологій наведено на рис. 2.1. у додатку Б

Система приймає дані у вигляді вимог технічного завдання та сюжету для розробки, і на виході отримуємо готову 3D-модель. Декомпозицію IDEF0 моделі наведено на рис 2.2. у додатку Б

На декомпозиції розписано, як саме система буде працювати. Спочатку аналіз предметної області, далі моделювання сцени. Моделювання сцени – це процес розробки всієї геометрії, яка буде використана в сцені. Після моделювання сцени йде її візуалізація. Візуалізація сцени – це процес налаштування матеріалів, ландшафту, анімацій, скриптів, які в підсумку утворюють готову сцену для подальшого створення додатка(гри).

Задля кращого уявлення про моделювання сцени була зроблена ще одна декомпозиція IDEF0 модель наведено рис. 2.3. у додатку Б

На декомпозиції детально показано процес моделювання сцени. Спочатку йде створення low-poly моделі, після того, як low-poly модель готова, йде її оптимізація (вирівнювання полігональної сітки, видалення непотрібних полігонів). Коли завершили з оптимізацією, ми робимо шейдинг (згладжування) моделі з подальшою розгорткою моделі на 2D-площину. Закінчивши с розгорткою на основі low-poly моделі ми створюємо high-poly модель для подальшого запікання карт нормалей. Після цих дій ми робимо рігінг моделі (сторення її скелету та інверсивної кінематики). На виході отримуємо готову модель(геометрію) для текстуровання.

Для кращого розуміння процесу візуалізації сцени було сформовано IDEF3 модель наведено на рис. 2.4. у додатку Б

На рис. 2.4 наведено загальний вигляд IDEF3 моделі інформаційної системи прийняття рішень для візуалізації сцени

Першим кроком є імпортування готової геометрії (3D-моделі) в ігровий рушій, коли модель була імпортована йде її налаштування матеріалу та анімації (для деяких моделей анімації не потрібні). Матеріал – це властивості, що визначають візуальну поведінку поверхні об'єкта. Матеріал визначає, як світло взаємодіє з поверхнею, які кольори вона відбиває, поглинає чи пропускає, наскільки вона блискуча чи матова.

Також потрібно налаштувати шейдер (програма, яка використовується у графічних обчисленнях для визначення вигляду та властивостей матеріалів та об'єктів на екрані), та освітлення на сцені. Ці параметри є обов'язковими. Ще одним кроком є налаштування ландшафту. Коли ландшафт налаштований, йде наступний етап, який включає два кроки: налаштування колізії( Визначення того, чи зіштовхується об'єкт (наприклад, персонаж гравця) з навколишнім середовищем, таким як стіни, підлога, меблі тощо) та створення навігаційного мешу для штучного інтелекту.

Коли всі попередні кроки були зроблені залишилось налаштування скриптів для сцени. Завдяки ним можна реалізовувати різноманітні аспекти сцени, такі як управління об'єктами, обробка подій, позиціонування об'єктів, налаштування ефектів освітлення та багато іншого.

## **2.2. Моделювання інформаційної системи розробки та оптимізації 3D-моделей для комп'ютерних ігор з використанням сучасних технологій**

Для роботи інформаційної системи для розробки і оптимізації 3D-моделей для комп'ютерних ігор було створено UML-діаграму станів (див. додаток В, рис. 2.5).

Геометрія моделі – на початковому етапі розробки є ключовим етапом, оскільки вона визначає основну структуру та форму об'єкта, який ви плануєте створити. Цей процес передбачає створення базового мешу, який є сіткою

з'єднаних полігонів або трикутників, що визначають поверхневу форму об'єкта. У цілому, правильно розроблена геометрична модель надає міцну основу для подальших етапів роботи над проектом та сприяє в досягненні бажаного результату.

Розгортка моделі – розділення тривимірної моделі на плоскі частини, які можуть бути розміщені на двовимірній площині. Основна мета розгортки полягає в тому, щоб підготувати поверхню об'єкта для подальшого текстурування, тобто накладання текстур або зображень на його поверхню так, щоб вони виглядали природно та правдоподібно [14],[15].

Високополігональна модель – створення 3D-моделі, яка складається з великої кількості полігонів, або вершин, що утворюють її поверхню. Це означає, що модель містить значно більше деталей та точок, ніж її низькополігональний еквівалент. Високополігональні моделі використовуються в графічному дизайні, ігровій індустрії та візуалізації для створення більш деталізованих та реалістичних зображень об'єктів чи персонажів. В нашому варіанті вона потрібна для запікання карт нормалей.

Карта нормалей – створення текстурного зображення, яке використовується в комп'ютерній графіці для візуалізації деталей на поверхні 3D-моделі. Дане зображення містить інформацію про напрямки поверхневих нормалей в кожній точці моделі. Нормалі вказують на те, як світло взаємодіє з поверхнею, визначаючи освітленість та тіньові ефекти. Карти нормалей потрібні для підвищення реалістичності графіки, додаючи деталі та рельєф до моделей, які, можливо, не представлені в їх основній геометрії [16].

Анімація моделі – створення руху та динаміки у тривимірному просторі для об'єктів чи персонажів за допомогою комп'ютерної графіки.

Текстура – створення двовимірного зображення або набору зображень, які накладаються на поверхню тривимірного об'єкта. Ці зображення містять інформацію про колір, рельєф, розмір, віддзеркалення світла, деталі та інші характеристики, які роблять об'єкт візуально цікавим та реалістичним [17].



Оптимізація – налаштування та вдосконалення різних етапів роботи над проектом для забезпечення ефективності та оптимальної продуктивності. Модель в ігровому рушії – налаштування матеріалів(текстур) та анімації моделі в ігровому рушії.

Також у системі представлено порядок дії по створенню моделі для імпортування в ігровий рушій (див. додаток В, рис. 2.6)

На рис. 2.6 можна побачити UML-діаграму активності ІС розробки та оптимізації 3D-моделей для комп'ютерних ігор. На ній більш детально розписано, як працює сама система, також показано розгалуження, при умові необхідності створення анімації для моделі. Якщо анімація потрібна то ми створюємо скелет та рігінг та потім імпортуємо готову анімацію до ігрового рушія.

Також представлено представлено UML-діаграму класів ІС прийняття рішень в для розробки і оптимізації 3D-моделей для комп'ютерних ігор (див. додаток В, рис. 2.7).

Клас `AActor` є базовим класом для всіх об'єктів в грі. Кожен об'єкт, який може з'явитися або взаємодіяти з ігровим світом, є похідним від цього класу. Він має методи для управління розташуванням, життєвим циклом та іншими аспектами гри.

`UObject` є базовим класом для усіх об'єктів у Unreal Engine. Він надає основні функції для керування пам'яттю, взаємодії об'єктів та відстеження їхнього життєвого циклу.

`UActorComponent` є компонентом, який можна приєднати до `AActor`, щоб надати йому додаткову функціональність. Він включає в себе методи для активації та деактивації, а також інші подібні функції.

`UStaticMeshComponent` використовується для відображення статичних 3D об'єктів у грі, таких як будинки чи ландшафт. Він містить методи для зміни статичного мешу та його розташування у просторі.

`USceneComponent` є базовим класом для компонентів, які можна приєднати до `AActor`. Він включає в себе методи для роботи з розташуванням

та просторовими трансформаціями. Використовується як основа для різноманітних компонентів, таких як світло, камера, звук та інші.

Для загального розуміння взаємодії замовник та розробника моделей було створено UML-діаграму послідовності (див. додаток В, рис. 2.8).

На даній UML-діаграмі виокремлено 6 окремих об'єктів ІС, а саме: розробник гри(замовник); графічний дизайнер; Blender; оптимізатор моделей;

Розробник гри виконує функцію замовлення моделі у графічного дизайнера та редагування її разом з графічним дизайнером. Графічний дизайнер приймає замовлення, створює концепцію моделі, а потім та розробляє першу її версію. Blender є програмою, в якій створює модель графічний дизайнер, та вже готову модель оптимізує оптимізатор моделей. Оптимізатор моделей виконує функцію аналізу на можливість оптимізації моделі, а також її оптимізації.

Для наглядності, чим займається замовник а чим розробник – створено діаграму прецедентів (див. додаток В, рис. 2.9)

У цій ІС є два ключових актора: замовник та розробник. Замовник формує замовлення для розробник ,якщо розробник його приймає розробник надає ТЗ. Коли розробник отримав ТЗ, він створює модель яку потім відправляє розробнику. Якщо модель не підходить під потреби ТЗ замовника а то він разом с розробником модернізують модель.

### **2.3 Процес реалізації системи скриптів для анімації 3D-моделей у відеогрі**

Система анімації в Unreal Engine включає різні інструменти та редактори, які поєднують скелетну деформацію сіток з деформацією вершин на основі морфінгу, щоб забезпечити складну анімацію. Ця система дозволяє створювати реалістичні базові рухи гравців, відтворюючи та змішуючи стандартні анімаційні послідовності, а також створюючи спеціальні рухи, такі як масштабування виступів і стін за допомогою Anim Montages. Також можна

застосовувати ефекти пошкоджень або вирази обличчя через Morph Targets, керувати перетворенням кісток за допомогою Skeletal Controls і створювати логіку State Machines, яка визначає, яку анімацію використовувати персонажу в певних ситуаціях.

Створення анімованих персонажів в Unreal Engine включає використання різних інструментів анімації, кожен з яких зосереджується на певному аспекті анімації.

Наприклад, редактор скелетів використовується для керування кісткою (ієрархією суглобів), яка контролює скелетну сітку та анімацію. Редактор скелетної сітки дозволяє змінювати скелетну сітку, яка пов'язана зі скелетом і є зовнішнім виглядом персонажа. Редактор анімації дозволяє створювати та змінювати анімаційні ресурси, а також налаштовувати їх. Редактор анімаційного плану використовується для створення логіки, яка визначає, які анімації використовує персонаж і коли, а також як вони поєднуються. Нарешті, редактор активів фізики використовується для створення та редагування фізичних тіл, які забезпечують зіткнення скелетних сіток (див. додаток Г, рис.2.10).

Скелет в UE — це ієрархія кісток, яка використовується для деформації скелетної сітки. В UE скелети абстрагуються від скелетних сіток у власному активі, що дозволяє анімації застосовуватися до скелета, а не до сітки. Завдяки цьому кілька скелетних сіток можуть використовувати одну анімацію. Приклад скелету в рушії наведено на Рис. 2.11 (див. додаток Г).

Animation Sequences — це ресурс, який можна відтворювати на Skeletal Mesh. Він містить ключові кадри, які визначають положення, обертання та масштаб кістки в певний момент часу. Відтворення цих ключових кадрів у послідовності дозволяє плавно анімувати кістки скелетної сітки.

Сповіщення анімації (AnimNotifies або просто Notifies) дозволяють програмістам анімації налаштовувати події, які мають відбуватися в певні моменти під час анімації. Сповіщення використовуються для додавання ефектів, таких як звуки кроків або системи частинок, і можуть бути розширені

для задоволення потреб будь-якої гри; Рис.2.12 (див. додаток 1) показує наглядне їх використання.

Animation Blueprints — це спеціалізований графік, який керує анімацією скелетної сітки. Графіки редагуються в редакторі анімаційного плану, де можна змішувати анімації, керувати кістками скелета або налаштовувати логіку для визначення остаточної анімаційної пози для кожного кадру. Він скаладається з «AnimGraph» (див. додаток Г, рис.2.13), де прив'язуються анімації та їх стани між собою, та «Event Graph», де за допомогою коду виконується логіка їх використання (див. додаток Г, рис.2.14).

Blend Spaces — це спеціальні ресурси для змішування анімацій на основі двох вхідних даних. Вони дозволяють виконувати більш складне змішування між кількома анімаціями на основі кількох значень, зменшуючи потребу у створенні жорстко закодованих вузлів для конкретних властивостей або умов. Blend Spaces дозволяють аніматору чи програмісту вказувати вхідні дані, анімації та спосіб змішування між ними. Кожна виділена точка, на Рис. 2.15(див. додаток Г), зображає окрему анімацію, що використовується при змішуванні.

Animation Montage (анімаційні монтажі) дозволяють використовувати широкий спектр анімаційних ефектів, зокрема інтелектуальні цикли анімації, перемикання анімацій на основі логіки тощо. Їх можна застосовувати через напряму викликаючи у Blueprint Visual Scripting або код. Вони являють собою звичайні пресети анімації, тому і робота над ними не є складною Рис. 2.16(див. додаток Г).

State Machines надають графічний спосіб розбити анімацію Skeletal Mesh на ряд станів, які регулюються правилами переходу, що контролюють, як переходити з одного стану в інший (див. додаток Г, рис.2.17). Це значно спрощує процес проектування анімацій Skeletal Mesh, дозволяючи легко контролювати, як персонажі переходять між типами анімацій без необхідності створювати складну мережу Blueprint.

Правильність роботи всіх анімацій можна побачити лише під час ігрово процесу. Тому необхідно запрограмувати персонажа. Вибираючи між Blueprint та C++, для даної задачі, було вибрано саме другий варіант. Код, що відповідає за рух, під час якого і можна протестувати всі анімації.

### **Висновок до другого розділу**

У другому розділі змодельовано роботу інформаційної системи, що лежить в основі розробки 3D-моделей, за допомогою різних типів UML-діаграм. Діаграма станів ілюструє, як система переходить з одного стану в інший; діаграма активності показує, як система виконує певні завдання; діаграма класів демонструє які класи присутні в системі та як вони взаємодіють між собою; діаграма прецедентів відображає, які функції система надає користувачам; діаграма послідовності показує, як система взаємодіє з користувачами.

Описано процес створення системи скриптів для динамічних анімацій для гри, що розробленій на рушії Unreal Engine. Розглянуто етапи від постановки задачі та проектування, що гарантує його ідеальну функціональність та зручність у користуванні.

## РОЗДІЛ 3. ТЕСТУВАННЯ ГРИ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

### 3.1 Інструкція користувачу інформаційної системи розробка та оптимізація 3D-моделей для комп'ютерних ігор з використанням сучасних технологій

Інструкція для розробника 3D-моделей. Програми для розробки та текстурування 3D-моделей вимагають конкретних технічних характеристик для запуску. Рекомендовані вимоги для Blender включають процесор від 4 ядер до 8 ядер, відеокарту з обсягом пам'яті 2GB або 8GB, і обсяг оперативної пам'яті від 8 ГБ. Для початку використання розробнику просто потрібно запустити файл з розширенням ".exe", і відбудеться завантаження головного меню. Коли головне меню завантажилось та після завершення процесу розробки, готова 3D-модель експортується з Blender в Substance Painter.

Рекомендовані вимоги для Substance Painter включають процесор Intel Core i3 або AMD Ryzen 3, відеокарту Nvidia Geforce GTX 1060 та 8 ГБ оперативної пам'яті. Для початку використання розробнику потрібно запустити файл з розширенням ".exe", і відбудеться завантаження головного меню в якому розробник імпортує модель для подальшої розробки текстур. Після завершення процесу текстурування, готову модель та текстури відправляють розробнику гри.

Інструкція для розробника гри. Для того щоб імпортувати готову 3D-модель в Unreal Engine перейдіть в редактор контенту. Клацніть правою кнопкою миші та виберіть "Import" для імпорту 3D-моделі. В діалоговому вікні виберіть формат файлу (зазвичай FBX) та вкажіть шлях до вашої 3D-моделі. Також виберіть параметри імпорту, такі як масштаб, повороти, тощо. Якщо ваша модель містить анімації, оберіть відповідні опції для імпорту анімацій. Використовуйте систему анімації Unreal Engine для створення аніматорів для 3D-моделі. Для текстур в Unreal Engine створіть матеріали. Прикріпіть текстури до ваших матеріалів та налаштуйте параметри, такі як блиск, прозорість і т.д. Перетягніть вашу модель у сцену Unreal Engine та визначте її

позицію, масштаб та орієнтацію. Завершивши ці кроки, можна приступити до тестування та оптимізації вашого проекту в реальному часі.

### **3.2 Тестування системи скриптів для анімації 3D-моделей та аналіз результатів**

Тестування скриптів для динамічної анімації — це процес перевірки та оцінки скриптів, які використовують різні методики і інструменти для створення анімацій, що змінюються в реальному часі.

Це важливий етап у розробці динамічних анімацій, що включає різні методи перевірки якості перед її реалізацією. Основна мета тестування – виявлення й усунення помилок, а також оцінка загальної якості динамічної анімації.

Стрес-тестування використовувалася для перевірки стабільності роботи системи скриптів під високими навантаженнями. Для Unreal Engine це може означати створення сценаріїв, що імітують велику кількість одночасних анімацій. Навантаження перевірялося за рахунок великої кількості зміни анімацій у полі зору рендеру. Різницю FPS можна побачити на рисунках 3.1 та 3.2 (див. додаток Г). Використання інструменту Unreal Insights дозволило нам моніторити продуктивність гри під час стрес-тестів та визначати, які аспекти потребують оптимізації.

Ручне тестування – це процес перевірки програмного забезпечення, який здійснюється вручну тестувальниками без використання автоматизованих інструментів. Основною метою ручного тестування є виявлення дефектів, помилок та невідповідностей у програмному забезпеченні до того, як воно буде випущене у виробництво. За допомогою ручного тестування проводилось реагування при натисканні клавіш керування. Оскільки система скриптів має підтримку лише ПК, система тестувалась на різних конфігураціях заліза на цій операційній системі.

Реагування на натискання клавіш керування можна побачити на рисунку 3.3 (див. додаток Г) . Також було налаштовано систему поєднання анімацій та її тестування, можна побачити на рисунку 3.4 (див. додаток Г).

Останніми етапами було «альфа» та «бета» тестування. Альфа тестування – це ранній етап тестування програмного забезпечення, який проводиться на початкових версіях продукту. Основні характеристики альфа тестування включають тестування розробниками та окремим колом користувачів. Бета тестування – це етап тестування програмного забезпечення, який проводиться після завершення альфа тестування і перед реалізацією продукту. Основні характеристики бета тестування включають зовнішнє тестування, яке проводиться реальними користувачами для виявлення багів та проблем, які можуть виникнути при реальних умовах експлуатації продукту.

Проаналізувавши різні етапи розробки, можна сказати, що кожен них не мав проблеми та баги. Але за допомогою тестування різними методами та численним виправленням, система скриптів пройшла всі необхідні тести для того, щоб бути готовою для реалізації в комп'ютерній грі.

### **Висновок до третього розділу**

У третьому розділі розроблено інструкцію для розробника 3D-моделей та розробнику гри та їх взаємодій з системою роботи с моделями та текстурами.

Також описано процес тестування готового продукту, який вимагав використання різноманітних методик для забезпечення її якості. Зокрема, Стрес-тестування продемонструвало стабільність скриптів для динамічної анімації під високими навантаженнями, а тестування системи на реагування, дає нам розуміння як система реагує на дії користувача з клавішами керування. Альфа- та бета-тестування забезпечило додаткову перевірку перед реалізацією в комп'ютерній. Завдяки комплексному підходу до тестування, система успішно пройшла всі перевірки й готова до реалізації.



## ЗАГАЛЬНІ ВИСНОВОКИ

У дипломній кваліфікаційній під час аналізу інформаційних потреб було визначено, що 3D-моделювання є важливим аспектом розробки комп'ютерних ігор. 3D-моделі можуть використовуватися для створення реалістичних ігрових світів, персонажів та об'єктів. Предметною областю даного проекту є розробка та оптимізація 3D-моделей для комп'ютерних ігор. Здійснено аналіз вимог до системи скриптів для анімації 3D-моделей. Ці вимоги включають функціональність, інтерфейс, інтеграцію та продуктивність.

Здійснено моделювання інформаційної системи, що лежить в основі розробки 3D-моделей, за допомогою різних типів UML-діаграм: діаграма станів (ілюструє, як система переходить з одного стану в інший); діаграма активності (показує, як система виконує певні завдання); діаграма класів (демонструє які класи присутні в системі та як вони взаємодіють між собою); діаграма прецедентів (відображає, які функції система надає користувачам); діаграма послідовності (показує, як система взаємодіє з користувачами).

Описано процес створення системи скриптів для динамічних анімацій для гри, що розроблена на русії Unreal Engine. Описано етапи створення системи скриптів, що включають – постановку задачі, проектування, реалізацію та тестування. Розроблена система може бути використана для створення 3D-моделей та анімацій для комп'ютерних ігор. Система є гнучкою та може бути адаптована до різних проектів.

Розроблено інструкцію для розробника 3D-моделей та розробника гри. Ця інструкція описує процес розробки та оптимізації 3D-моделей для комп'ютерних ігор з використанням сучасних технологій та запропонованої системи скриптів для динамічних анімацій.

Описано процес тестування готового продукту. Цей процес включав в себе стресс-тестування, тестування системи на реагування, а також альфа- та бета-тестування. Завдяки комплексному підходу до тестування, система скриптів динамічної анімації успішно пройшла всі перевірки й готова до реалізації.

Розроблена система скриптів динамічної анімації є стійкою та ефективною. Вона успішно пройшла стресс-тестування та продемонструвала високу продуктивність під високими навантаженнями. Система скриптів динамічної анімації зручна у використанні є гнучкою та розширюваною. Розробники 3D-моделей та розробники ігор можуть легко використовувати цю систему для створення анімацій різної складності та для різних типів комп'ютерних ігор.

Важливо зазначити, що це лише перший крок у розробці системи скриптів для динамічної анімації. У майбутньому планується розширити функціональність системи, додати нові можливості та вдосконалити її продуктивність. А саме, дослідити можливість використання системи скриптів динамічної анімації для створення анімацій в віртуальній реальності; розробити додаткові інструменти та методи для покращення зручності використання та гнучкості системи скриптів динамічної анімації; провести дослідження для визначення впливу системи скриптів динамічної анімації на візуальну якість ігрового процесу та на досвід гравців.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Adobe Substance Painter: <https://www.adobe.com/ru/products/substance3d-painter.html> (Дата звернення: 13.03.2024)
2. Autodesk 3ds Max: <https://www.autodesk.com/products/3ds-max/overview?term=1-YEAR&tab=subscription> (Дата звернення: 13.03.2024)
3. Blender documentation: <https://docs.blender.org> (Дата звернення: 13.03.2024)
4. Blender: <https://www.blender.org> (Дата звернення: 13.03.2024)
5. Cinema 4D: <https://www.maxon.net/en/cinema-4d> (Дата звернення: 14.03.2024)
6. Erik Van Horn, 3D Character Development Workshop Rigging Fundamentals for Artists and Animators: посібник , Mercury Learning and Information , 2018, 234с.
7. Mari - High-resolution digital 3D texture painting: <https://www.foundry.com/products/mari> (Дата звернення: 17.03.2024)
8. Marmoset: <https://marmoset.co> (Дата звернення: 17.03.2024)
9. Maya documentation: <https://www.autodesk.com/support/technical/article/caas/tsarticles/ts/1C3jaffqnWFyQoLPEPm7n.html> (Дата звернення: 17.03.2024)
10. Maya: <https://www.autodesk.com/products/maya/overview?term=1-YEAR&tab=subscription> (Дата звернення: 17.03.2024)
11. Nintendo Switch: <https://www.nintendo.com/us/switch/> (Дата звернення: 17.03.2024)
12. Playstation: <https://www.playstation.com/en-us/> (Дата звернення: 18.03.2024)
13. Substance 3D Painter 2023: [https://store.steampowered.com/app/2199970/Substance\\_3D\\_Painter\\_2023/](https://store.steampowered.com/app/2199970/Substance_3D_Painter_2023/) (Дата звернення: 20.03.2024)

14. Tony Parisi, *Learning Virtual Reality: Developing Immersive Experiences and Applications for Desktop, Web, and Mobile 1st Edition*: посібник, O'Reilly , 2015 , 172с.
15. XBOX: <https://www.xbox.com/en-US?source=lp> (Дата звернення: 4.04.2024)
16. Алгоритм створення тривимірних ігрових моделей високої складності: <http://mdu.edu.ua/wp-content/uploads/gmit6-72.pdf> (Дата звернення: 15.04.2024)
17. Засоби редагування та візуалізації в реальному часі для 3D-художників: <https://idealsoft.com.ua/vendory/marmoset/> (Дата звернення: 16.04.2024)
18. Ковальчук, М. О., & Колесник, Н. Є. (2020). *Графічний дизайн та комп'ютерна графіка: монографія*. Житомир: ТОВ, 505, 440.
19. Лотошинська Н. Д., Ізонін І.В. *Технології 3D-моделювання в програмному середовищі 3ds Max з дисципліни "3D-Графіка"*: посібник, Львівська політехніка, 2020, 216 с.
20. Майстерня Dota 2 – UV-розгортка предмета: <https://help.steampowered.com/uk/faqs/view/33BE-0778-61E9-6270> (Дата звернення: 17.04.2024)
21. Огляд 3ds Max: <https://3ddevice.com.ua/ru/blog/3d-printer-obzory/obzor-3ds-max/> (Дата звернення: 18.04.2024)
22. Програмне забезпечення Maxon Cinema 4D: <https://dreamtech.ua/ua/programnoe-obespechenie-maxon-cinema-4d---annual-subscription-mx-u> (Дата звернення: 19.04.2024)
23. Процедурні текстури в 3D-графіці: суть та значення: <https://buki.com.ua/blogs/procedurni-teksturi-v-3d-grafici-sut-ta-znacennya/> (Дата звернення: 2.05.2024)
24. Синиця, М. (2013). Використання мультимедійних технологій в умовах сталого розвитку освіти. *Сталий розвиток: проблеми та перспективи, збірник наукових праць*. Житомир : ЖДУ ім. І. Франка, 369-384.

25. Чернявський, Б., Ковальчук, М., Яворський, О., & Могила, Є. (2022). Концепції проектування дизайн-продуктів. Актуальні проблеми сучасного дизайну. Київський національний університет технологій та дизайну

[https://er.knutd.edu.ua/bitstream/123456789/21028/1/APSD\\_2022\\_V1\\_P290-292.pdf](https://er.knutd.edu.ua/bitstream/123456789/21028/1/APSD_2022_V1_P290-292.pdf)

26. Що таке карти нормалей: <https://zkan.com.ua/sho/shho-take-karti-normalej.html> (Дата звернення: 05.05.2024)

## ДОДАТКИ

## Додаток А.

Таблиця 1.1 – Порівняльна характеристика аналогів Blender

Конкуренти	Необхідність в оплаті	Рігінг та анімація	Плагіни	Часті оновлення продукту	Рендери	Зручність Інтерфейсу для користувача
Autodesk Maya	Так	Так	Так	Ні	Так	Ні
Cinema 4D	Так	Так	Так	Ні	Так	Ні
Autodesk 3Ds Max	Так	Так	Так	Ні	Так	Так
Blender	Ні	Так	Так	Так	Так	Так

Таблиця 1.2 – Порівняльна характеристика аналогів Substance

## Painter

Конкуренти	Робота з матеріалами	Запікання текстур	Плагіни	Можливість рендеру моделі с текстурой
Marmoset	Так	Так	Ні	Так
MARI	Так	Так	Ні	Ні
Substance Painter	Так	Так	Так	Так

## Моделювання бізнес-процесів предметної області

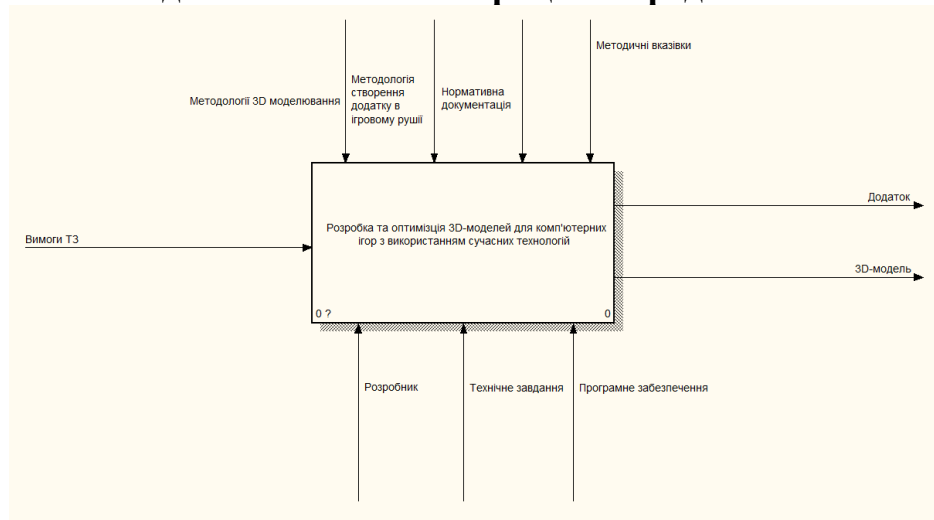


Рис. 2.1 – IDEF0 діаграма роботи ІС

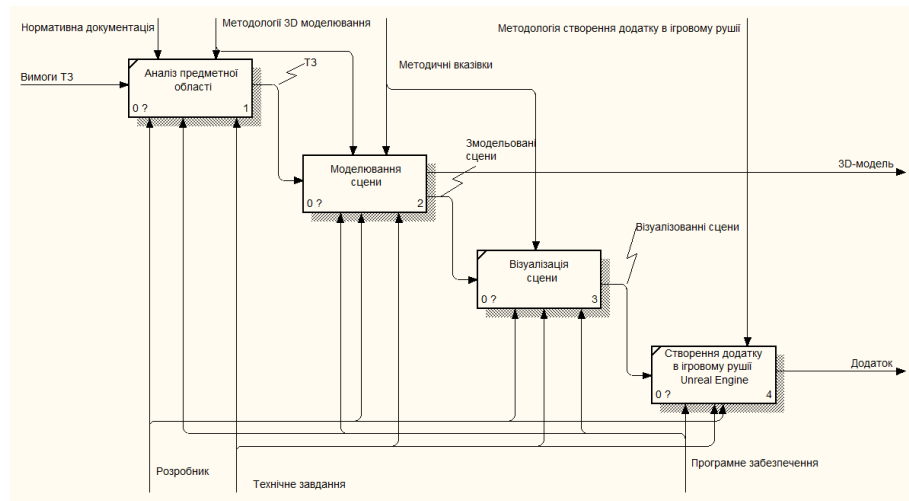


Рис. 2.2 – IDEF0 «Розробка та оптимізація 3D-моделей для комп'ютерних

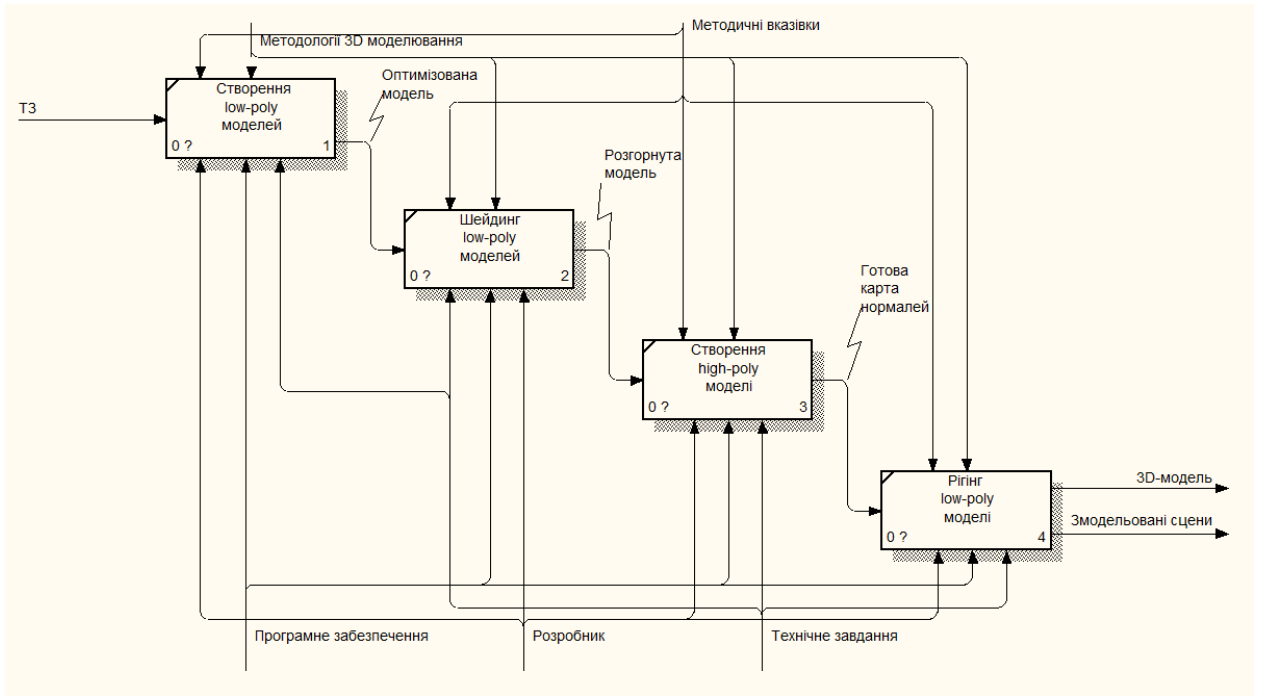


Рис. 2.3 – IDEF0 «Моделювання сцени»

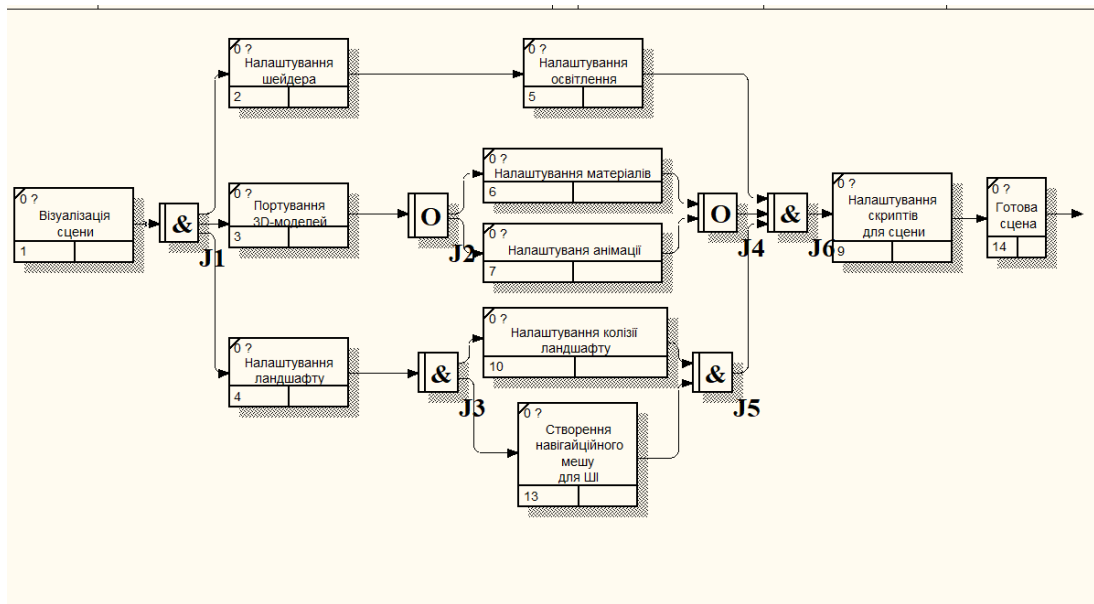


Рис. 2.4 – IDEF3 модель візуалізації сцени



**Моделювання інформаційної системи розробки та оптимізації 3D-моделей для комп'ютерних ігор з використанням сучасних технологій**

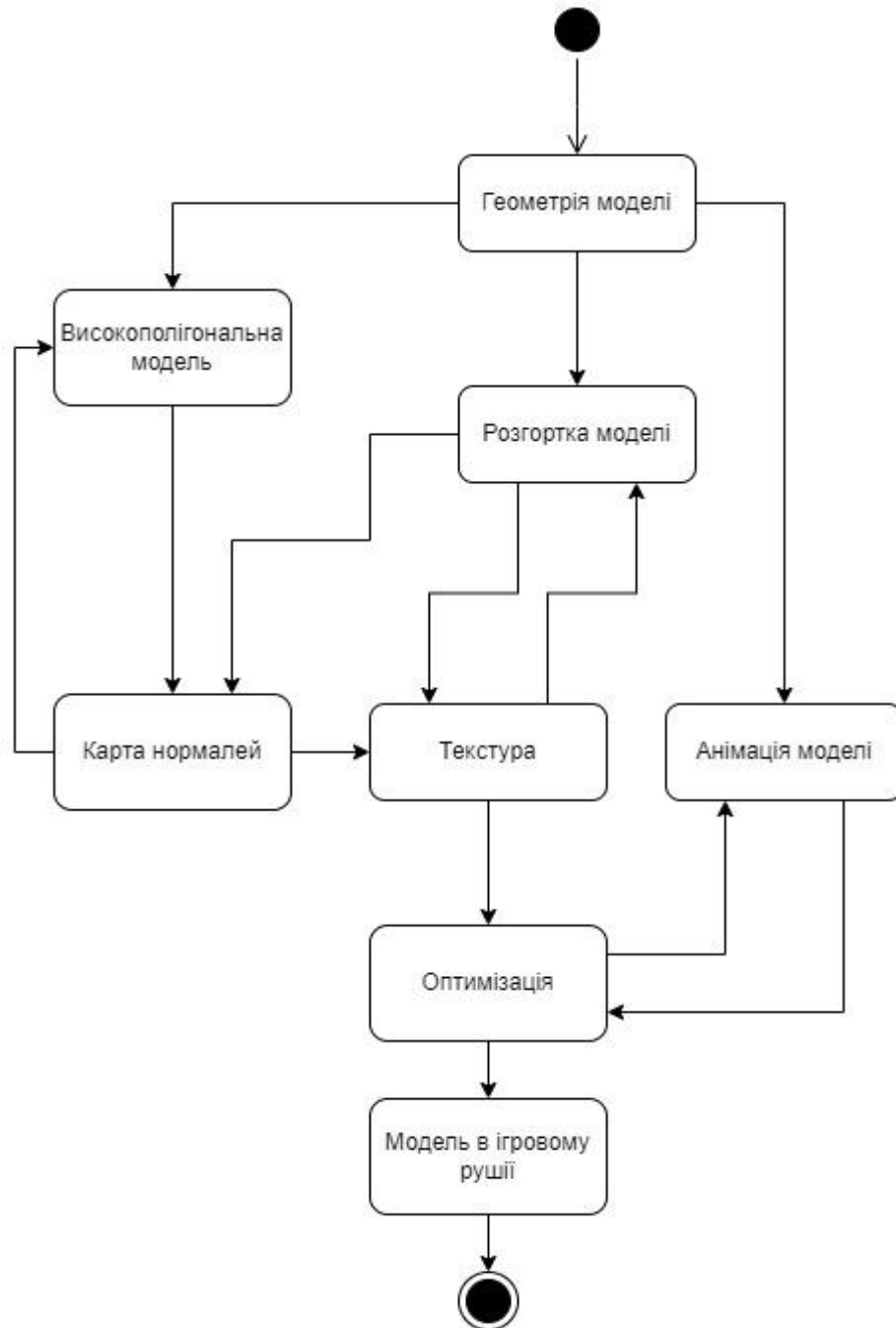


Рис. 2.5 – UML-діаграма станів

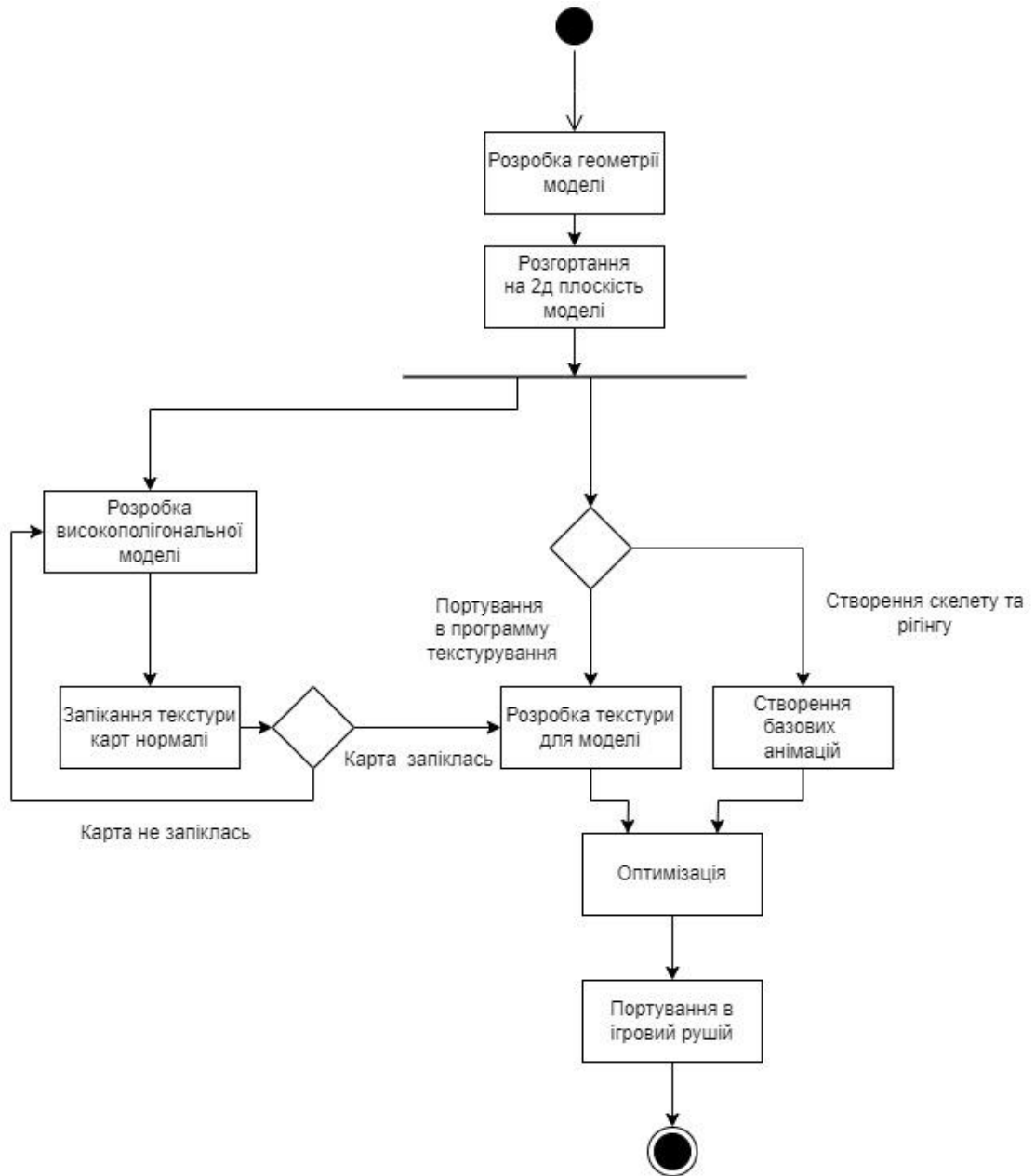


Рис. 2.6 – UML-діаграма активності

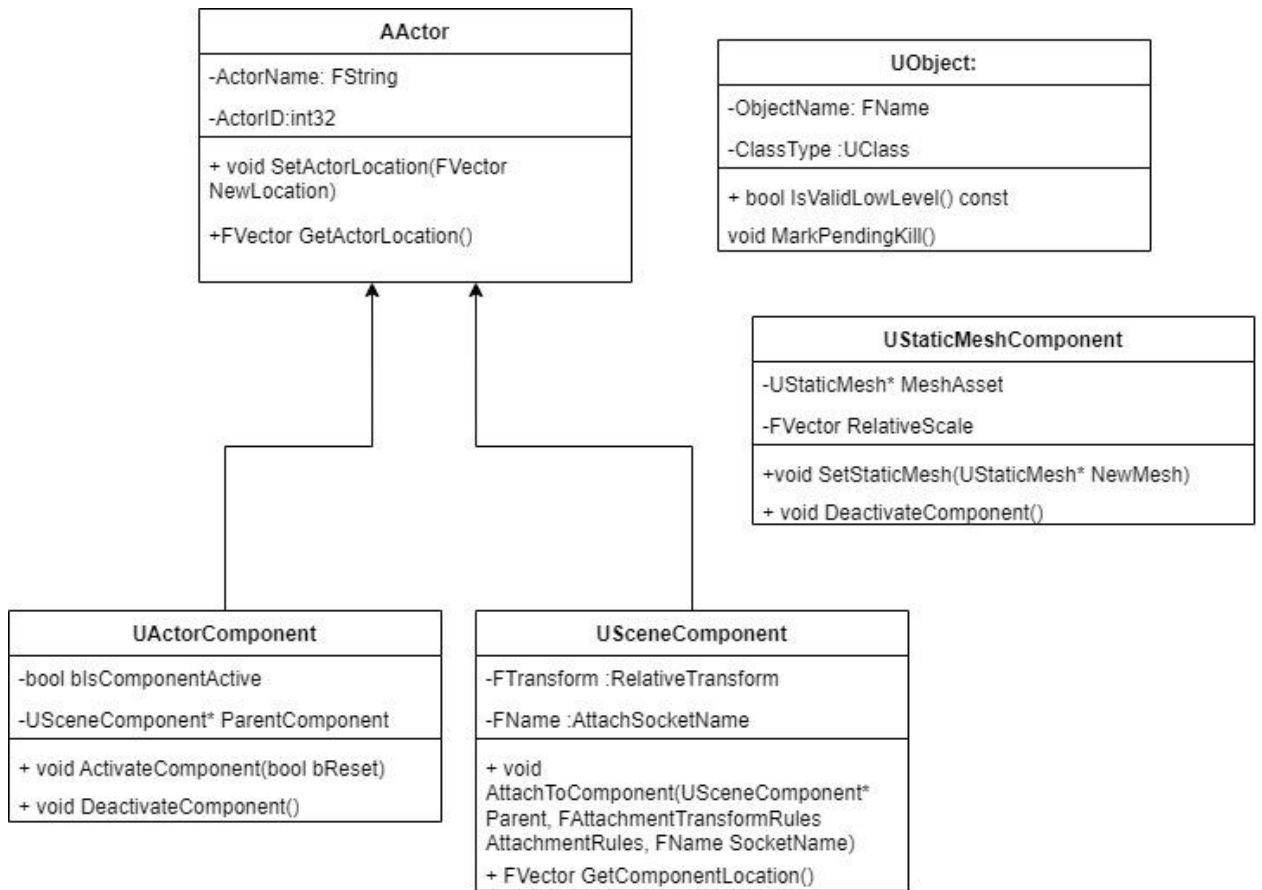


Рис. 2.7 – UML-діаграма класів

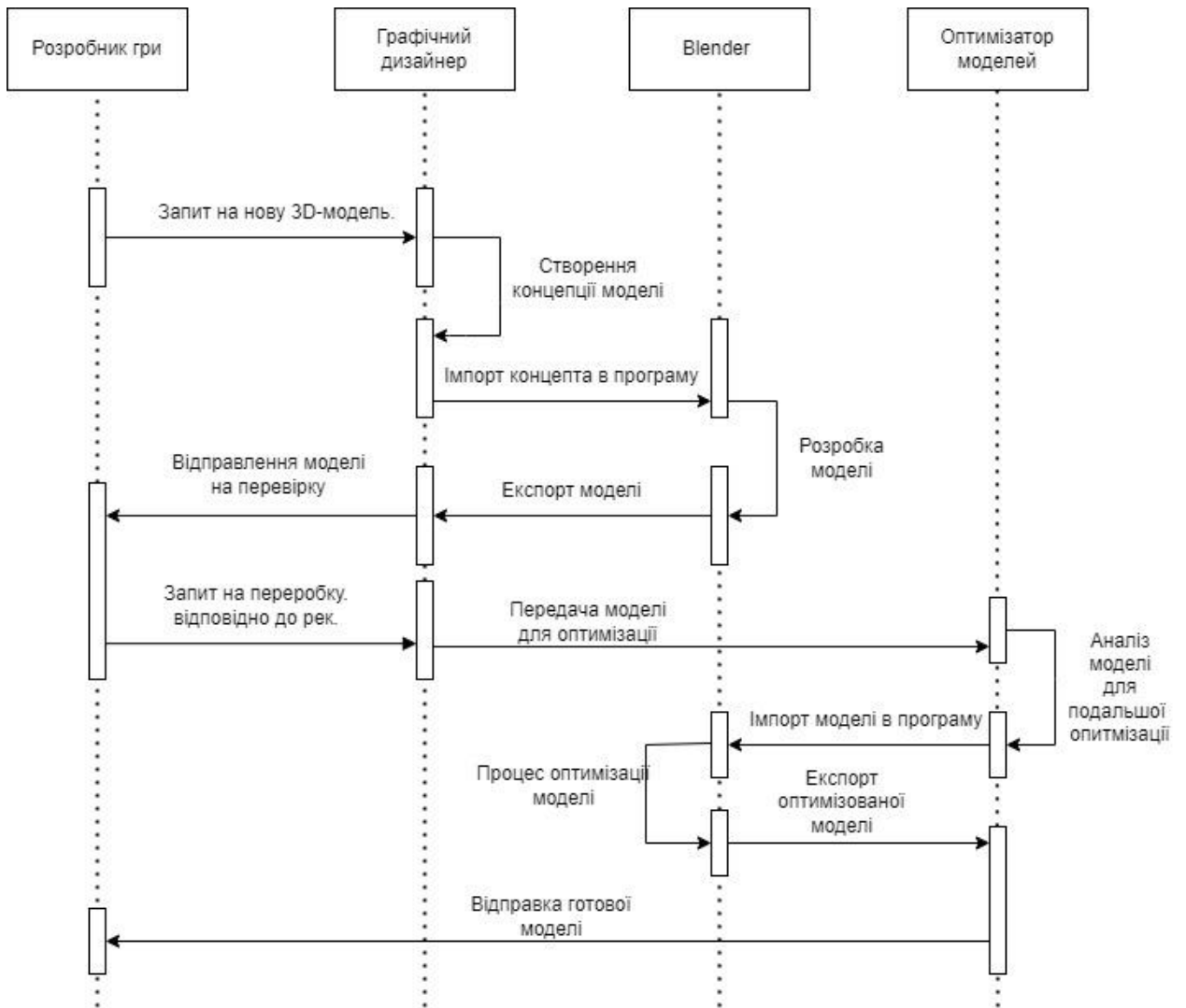


Рис. 2.8 – UML-діаграма послідовності

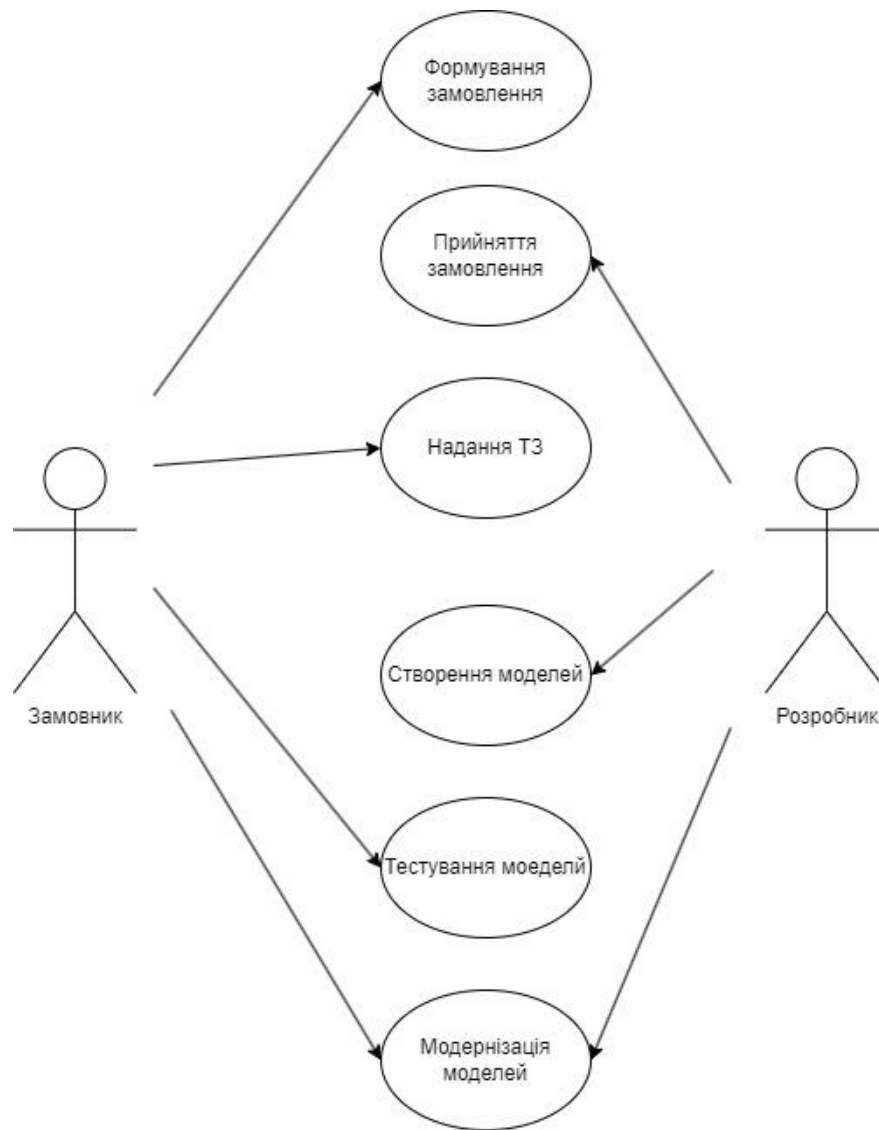


Рис. 2.9 – UML-діаграма прецедентів

## Процес реалізації системи скриптів для анімації 3D-моделей у відеогрі



Рис. 2.10 – Physics Asset, що прив'язується до скелету моделі

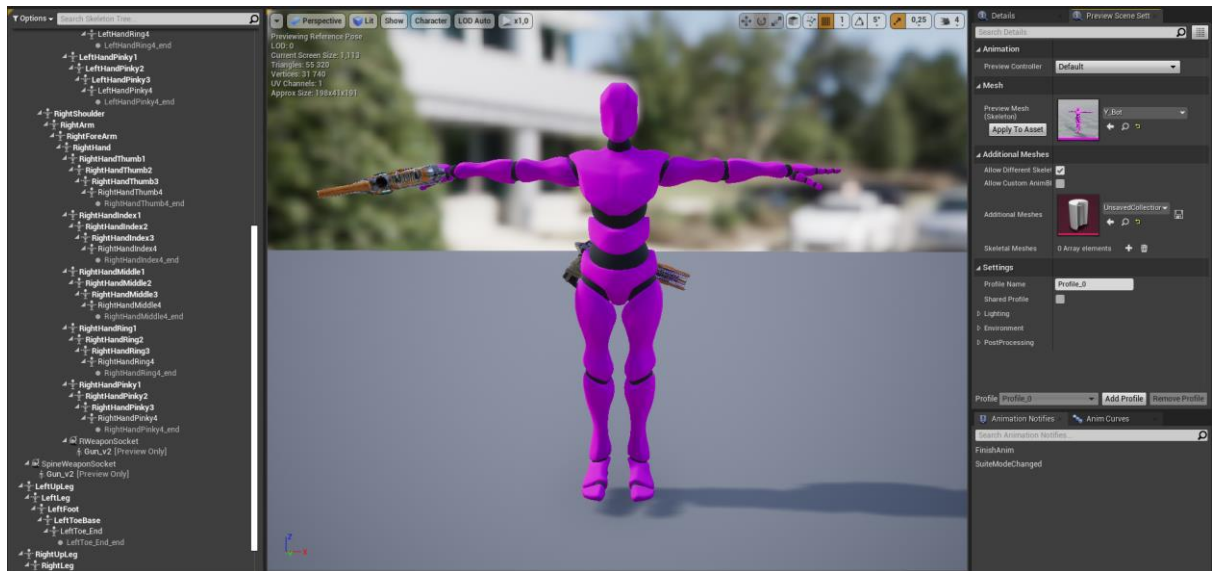


Рис. 2.11 – Skeleton y Unreal Engine

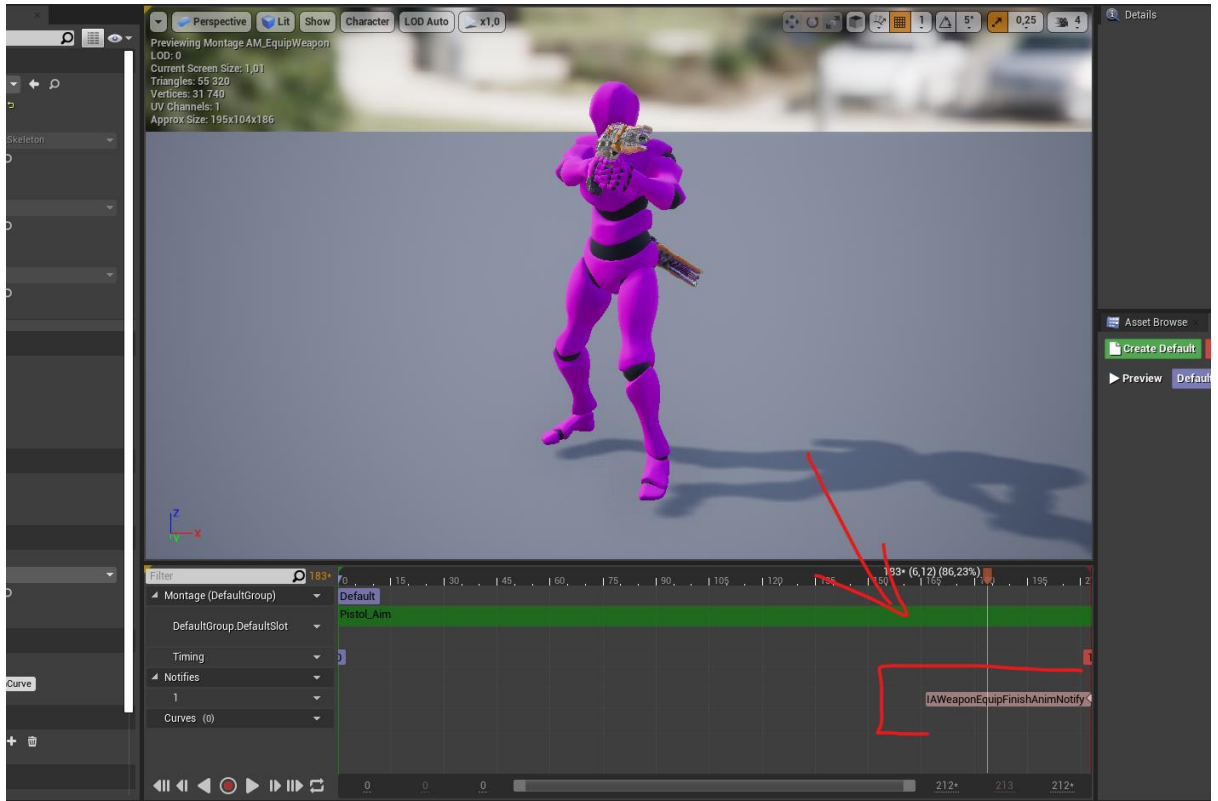


Рис. 2.12 – Вигляд AnimNotify на анімації

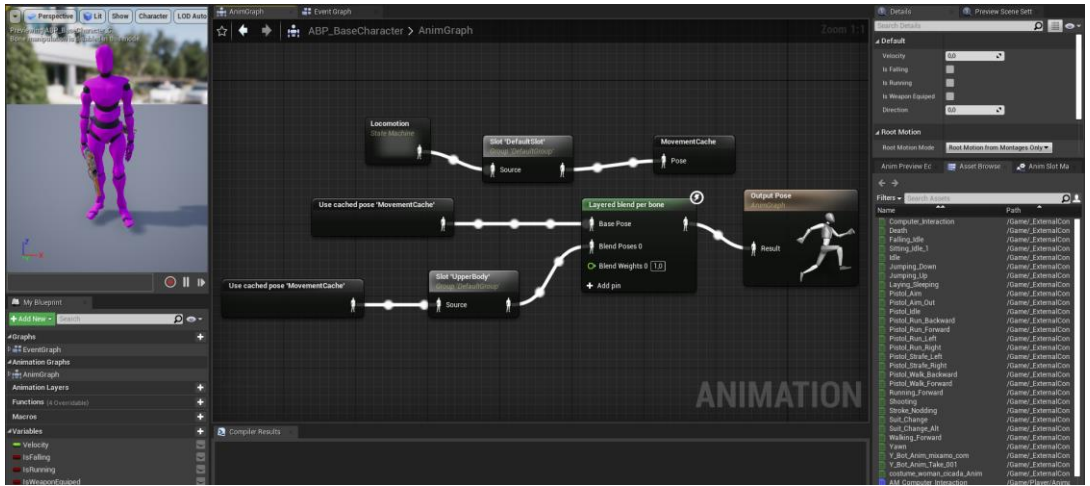


Рис. 2.13 – AnimGraph з прив'язаними анімаціями

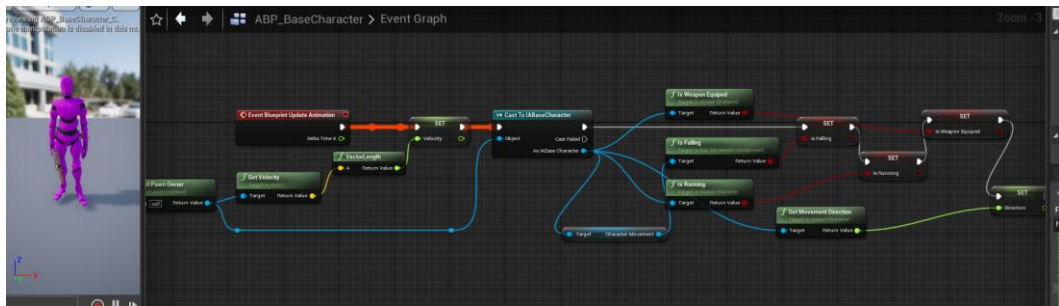


Рис.

2.14 – Event Graph, де оброблюється логіки викилику анімацій

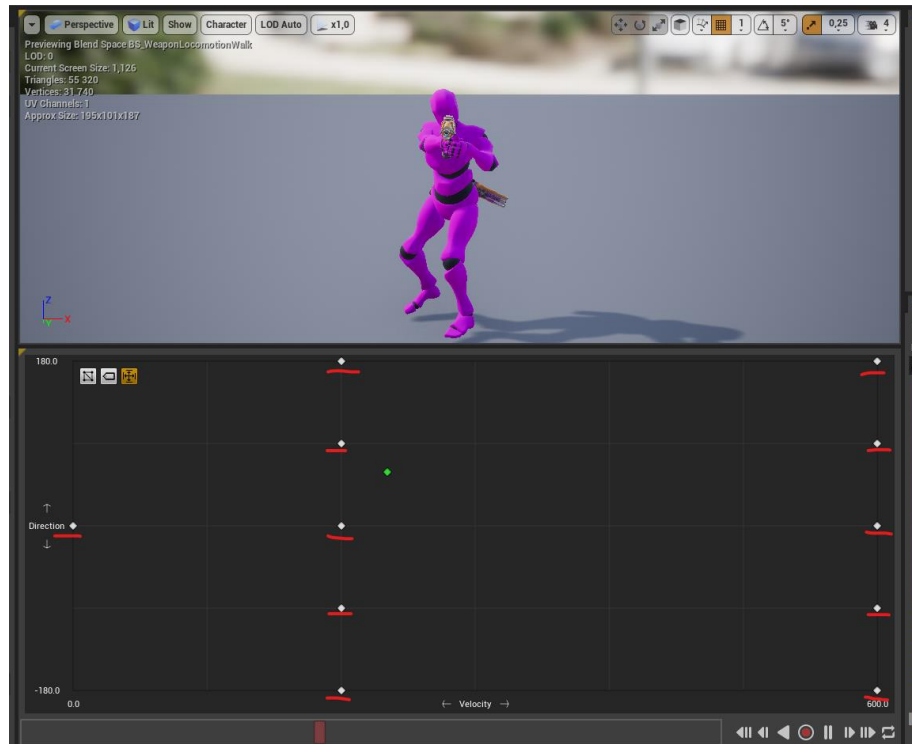


Рис. 2.15 – Вигляд Blend Space



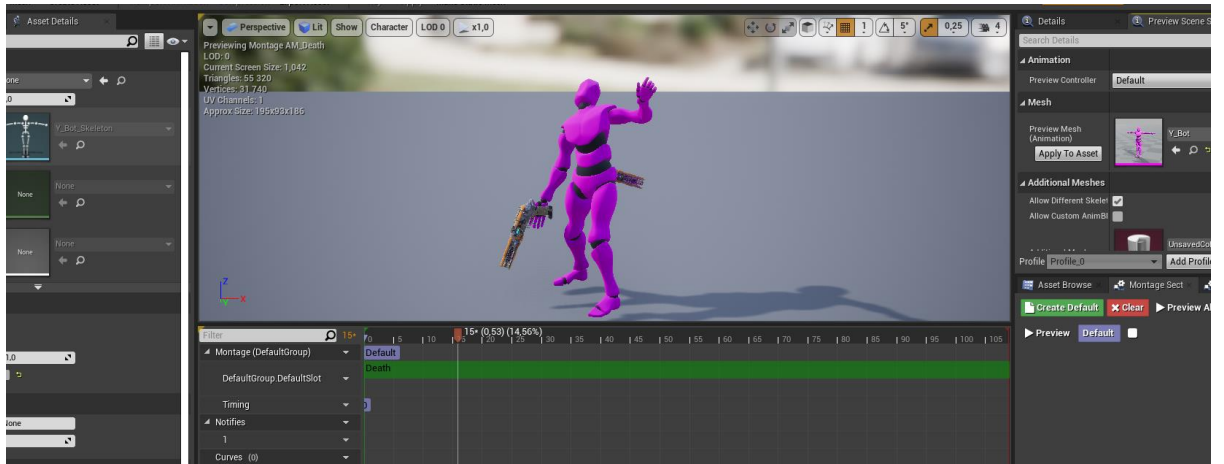


Рис. 2.16 – Налаштування Animation Montage

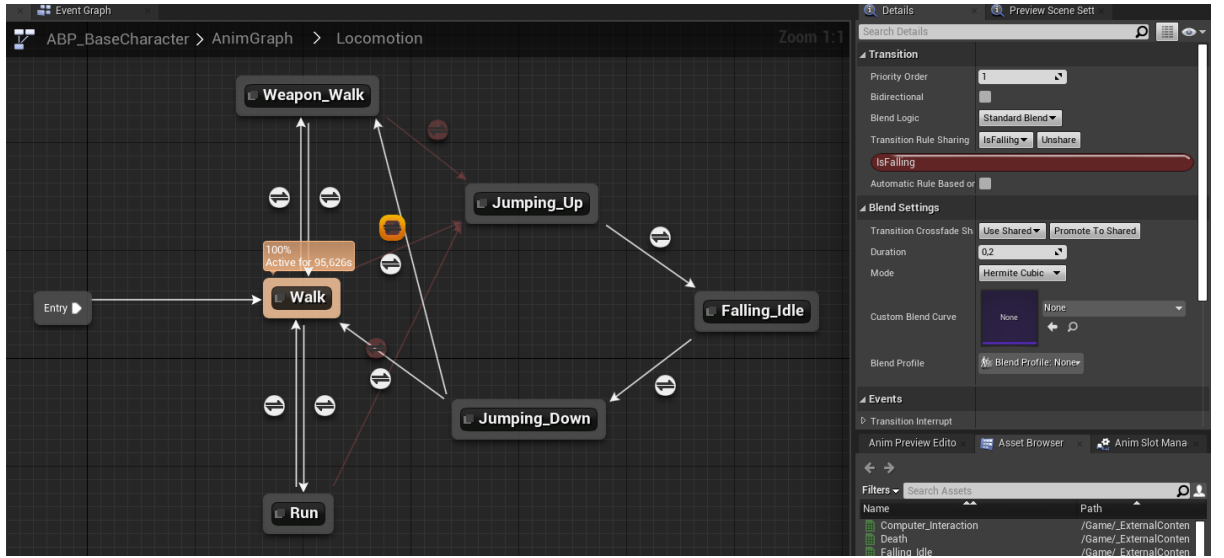


Рис. 2.17 – State Machine для анімації руху персонажа

## Тестування системи скриптів для анімації 3D-моделей та аналіз результатів



Рис. 3.1 – Типове навантаження при анімації (40,62 FPS)

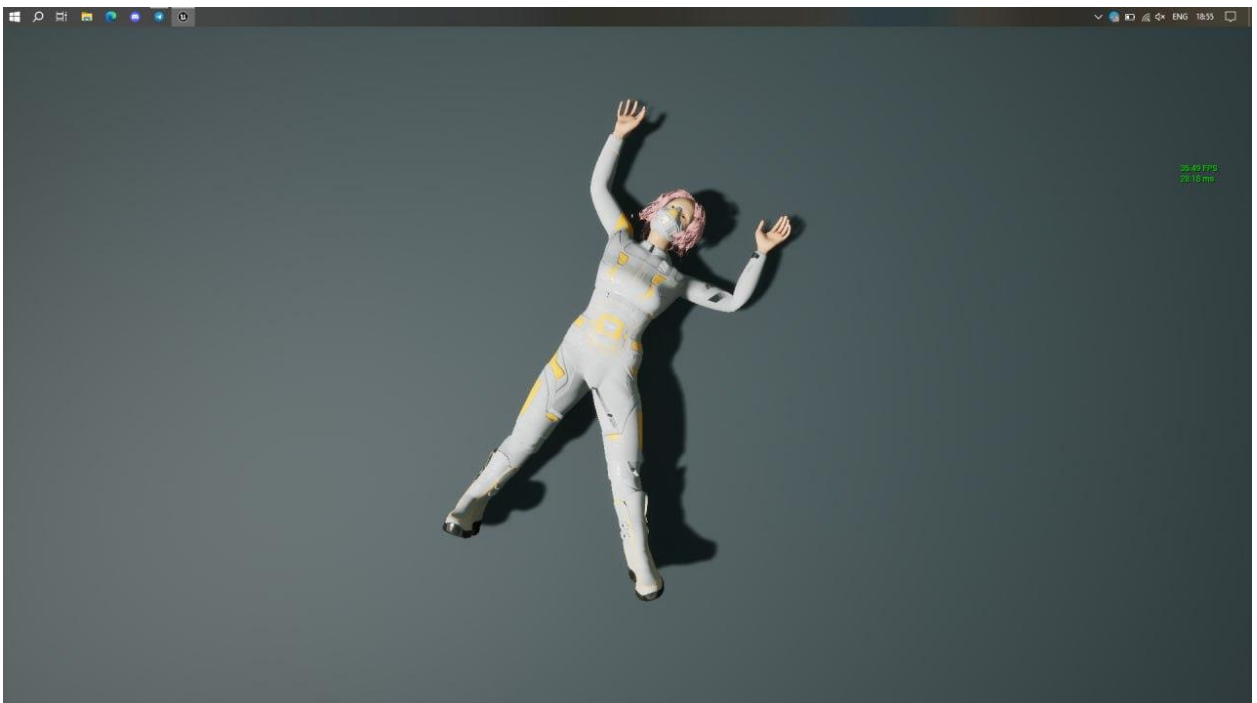


Рис. 3.2 – Навантаження при змінах анімацій (35,49 FPS)

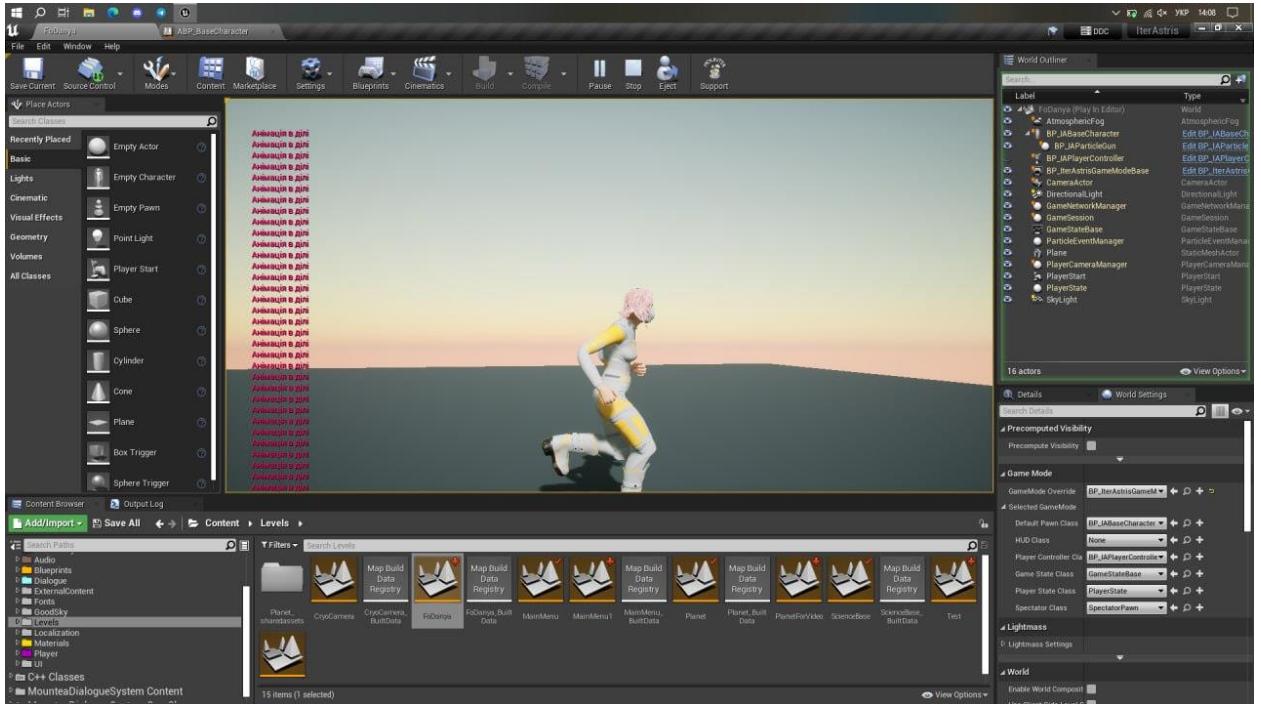


Рис. 3.3 – Тестування на реагування анімацій на натискання клавіш керування



Рис. 3.4 – Налаштування і тестування Blend Space