

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

Крисквич Дмитро Андрійович

УДК 005.382.713.421.1

КВАЛІФІКАЦІЙНА РОБОТА

3D моделювання сцен для комп'ютерних ігор

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,
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

Крисевич Д.А. 3D моделювання сцен для комп'ютерних ігор – кваліфікаційна робота на правах рукопису.

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

Розробка ігор є динамічною сферою, яка постійно пропонує нові технології.

Мета цієї роботи полягає в тому, щоб дослідити процес 3D моделювання сцен для комп'ютерних ігор та описати ключові етапи цього процесу.

Ключові слова: Blender, 3D моделювання, Текстурування, Освітлення, Рендеринг.

ABSTRACT

Krysevych D.A. 3D modeling of the scene for computer games - qualifying work on the rights of the manuscript.

Qualification work for obtaining a bachelor's degree in specialty 122 - Computer science. – Polis National University, Zhytomyr, 2024.

Game development is a dynamic field that constantly offers new technologies.

The purpose of this work is to investigate the process of 3D scene modeling for computer games and to describe the key stages of this process.

Keywords: Blender, 3D modeling, Texturing, Lighting, Rendering.

Зміст

ВСТУП.....	6
РОЗДІЛ 1. ТЕОРЕТИЧНІ ЗАСАДИ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	8
1.1. Аналіз інформаційних потреб і визначення предметної області дослідження	8
1.2. Аналіз алгоритмів побудови 3D-сцен для комп'ютерних ігор	10
1.3. Вибір технічного забезпечення для розробки та 3-D моделювання сцен.	17
Висновки до першого розділу	21
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ МЕТОДИКИ 3-D МОДЕЛЮВАННЯ СЦЕН.....	23
2.1. Алгоритми побудови реалістичних 3D-сцен.....	23
2.2. Методика 3-D моделювання сцен для комп'ютерних ігор	25
2.3. Процес реалізації 3-D сцени для комп'ютерних ігор за авторською методикою.....	28
Висновки до другого розділу	30
РОЗДІЛ 3. ТЕСТУВАННЯ МЕТОДИКИ 3-D МОДЕЛЮВАННЯ СЦЕН.....	31
3.1. Тестування 3D-моделей сцен.....	31
3.2. Аналіз результатів.....	32
Висновки до третього розділу.....	33
ЗАГАЛЬНІ ВИСНОВКИ	35
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	37
ДОДАТКИ.....	39

ВСТУП

Актуальність. 3-D моделювання є одним з найважливіших етапів розробки комп'ютерних ігор. Від якості 3-D моделі залежить візуальна привабливість гри, її реалістичність та ігровий процес. В даній дипломній роботі буде досліджено сучасні методи та інструменти 3-D моделювання, а також розроблена методика 3-D моделювання сцен для комп'ютерних ігор, що відповідає сучасним вимогам.

Мета кваліфікаційно роботи присвячена розробці методики для створення високоякісних 3-D сцен для комп'ютерних ігор.

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

- 1) проаналізувати предметну область дослідження;
- 2) визначити вимоги до розробок такого типу;
- 3) розробити методику 3-D моделювання сцен для комп'ютерних ігор;
- 4) спроектувати прототип 3D-моделей сцен для комп'ютерних ігор;
- 5) протестувати розроблені 3D-моделі сцен.

Предмет дослідження є процес розробки 3D-моделей сцен для комп'ютерних ігор.

Об'єктом дослідження є теоретичні та практичні аспекти 3-D моделювання сцен для комп'ютерних ігор.

У процесі дослідження були використані наступні **методи дослідження**: *аналізу* (для дослідження предметної області та визначення вимог до 3D-моделей сцен); *моделювання* (для розробки методики 3-D моделювання сцен для комп'ютерних ігор); *тестування* (для оцінки працездатності та ефективності розробленої методики моделювання реалістичних 3-D сцен).

Практичне значення досліджуваної теми полягає в тому, що результати дослідження можуть бути використані для підвищення ефективності розробки 3D-моделей сцен у відеоіграх; зменшення часу та зусиль, необхідних для розробки візуалізації; підвищення продуктивності комп'ютерних ігор, які використовують 3D-моделі.

Теоретичне значення досліджуваної теми полягає у розробці методики 3-D моделювання сцен для комп'ютерних ігор, що буде враховувати сучасні

вимоги до візуальної якості та ігрового процесу. Розширенні області застосування теорії розробки 3D-моделей сцен. **Унікальністю** кваліфікаційної роботи є розроблена методика реалізації 3D-моделей сцен.

Робота апробована на двох конференціях: Міжфакультетській науково-практичній інтернет-конференції здобувачів вищої освіти і молодих вчених “Безпека, технології, інновації: нові горизонти”, (м.Житомир, 2023 р.) та XXIV Міжнародна науково практична конференція «Modern technologies among us in the environment», (Рим, Італія, 2024 р.).

Структура та обсяг роботи. Кваліфікаційна робота складається з вступу, трьох розділів, висновку та списку використаних джерел. Загальний обсяг роботи становить 35 сторінок.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ЗАСАДИ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Аналіз інформаційних потреб і визначення предметної області дослідження

3D-моделювання сцен для комп'ютерних ігор є складним і багатогранним процесом, який охоплює широкий спектр завдань. Цей процес включає створення тривимірних моделей об'єктів, їх текстуровання, освітлення, анімацію та оптимізацію для забезпечення високої продуктивності гри. Розглянемо детальніше основні етапи цього процесу та проблеми, з якими стикаються розробники.

Для цього проаналізуємо наукову літературу за напрямом дослідження. Зокрема у книзі Еріка Ленгеля "3D Game Design: A Practical Introduction" [5] описано різні аспекти дизайну 3D-ігор, включаючи проблеми 3D-моделювання сцен. Автор розглядає такі проблеми, як оптимізація моделей, створення анімації, наповнення сцен та дизайн рівнів. У науковому доробку Джонатана Райта "Unreal Engine 4 Game Development Essentials" [22] увага фокусується на розробці ігор за допомогою Unreal Engine 4, популярного рушія для 3D-ігор, він описує проблеми 3D-моделювання сцен, пов'язані з Unreal Engine 4, такі як інтеграція 3D-сцен з двигуном та оптимізація продуктивності. Майк Кері у посібнику "3D Modeling and Rendering with Blender" [3] описує проблеми 3D-моделювання сцен, пов'язані з Blender, такі як створення 3D-моделей, текстуровання та освітлення. Андрій Гласнер у своїй фундаментальній праці "The Art of 3D Computer Graphics and Animation" [8] розкриває основи 3D-комп'ютерної графіки та анімації, що є важливою основою для 3D-моделювання сцен. Автор описує такі проблеми, як 3D-моделювання, 3D-текстуровання, 3D-освітлення та 3D-анімація. Олександр Гусєв у книзі "Створення 3D-світів: практичний посібник" [24] надає практичні рекомендації щодо створення 3D-світів, включаючи 3D-моделювання та візуалізацію. Він наголошує на ряді проблем 3D-моделювання сцен для комп'ютерних ігор, зокрема на таких, як – оптимізація моделей, текстуровання та освітлення. Сергій Ковальчук у посібнику "3D-моделювання та візуалізація: теорія та практика" [28] зазначає на важливості теоретичної бази з 3D-моделювання та візуалізації, що важливо для

розуміння принципів роботи програмних пакетів. Автор описує такі проблеми, як 3D-моделювання, 3D-текстурування та 3D-освітлення. У доробку Андрія Кузьменка "Unreal Engine 4: розробка ігор та віртуальної реальності" [29] увага фокусується на розробці ігор та віртуальної реальності за допомогою Unreal Engine 4. Основною проблемою 3D-моделювання сцен, пов'язаних з Unreal Engine 4, є інтеграція 3D-сцен з двигуном та оптимізація продуктивності.

Отже, основними проблемами 3D-моделювання сцен є – *оптимізація моделей* (3D-моделі повинні бути оптимізовані, щоб зменшити їх полігональність та покращити швидкість рендеру); *створення реалістичних сцен та їх наповнення* (створення реалістичних та деталізованих сцен може потребувати значних ресурсів часу та пам'яті); *інтеграція 3D-сцен з ігровим двигуном*.

Вважаємо за необхідне розглянути виділені проблеми, з точки зору поетапної розробки 3D-сцен. На першому етапі 3D-моделювання створюються тривимірні моделі всіх елементів сцени, включаючи персонажів, об'єкти, будівлі, ландшафти та інші елементи. Для цього використовуються спеціалізовані програмні засоби, такі як Blender, Autodesk Maya або 3ds Max. Основна складність полягає у створенні реалістичних та детальних моделей, які б візуально відповідали реальним об'єктам. Це потребує значних знань та навичок у галузі 3D-моделювання, а також розуміння пропорцій та анатомії.

Етап текстурування включає нанесення текстур на 3D-моделі для надання їм реалістичного вигляду. Текстури можуть включати кольорові карти, нормальні карти, карти висот та інші типи. Ключовою складністю на цьому етапі може бути створення високоякісних текстур, які б гармонійно поєднувалися з моделями та відповідали візуальному стилю гри.

Розробка освітлення сцени є ключовим етапом, який впливає на атмосферу та настрій гри, для цього використовуються різні типи світильників та техніки освітлення, такі як глобальне освітлення та освітлення на основі фізики. Створення реалістичного освітлення, яке б візуально відповідало реальному світу – потребує розуміння принципів фізичного освітлення та знання технік рендерингу.

На етапі створення анімацій для елементів сцени, головна складність полягає у розробці анімацій, які б візуально відповідали реальним рухам, що потребує знань анатомії та принципів анімації.

Процес оптимізації 3D-моделей, текстур, анімацій та інших ресурсів для забезпечення високої продуктивності гри вимагає балансування між якістю графіки та продуктивністю гри: зниження кількості полігонів, оптимізація текстур та зменшення навантаження на процесор і графічний процесор.

Таким чином, 3D-моделювання сцен для комп'ютерних ігор є складним процесом, який вимагає знань у багатьох областях, включаючи моделювання, текстурування, освітлення, анімацію та оптимізацію. Дослідники та розробники постійно працюють над створенням нових інструментів та методів для 3D-моделювання сцен для комп'ютерних ігор. Ці інструменти та методи мають бути ефективними та зручними у використанні, щоб спростити процес створення високоякісних 3D-сцени.

1.2. Аналіз алгоритмів побудови 3D-сцен для комп'ютерних ігор

Основні сучасні методи моделювання включають: полігони, сплайни, фрактали, хмари точок та вокселі. Полігональні моделі, або полігональні сітки, мають найширше застосування в комп'ютерній графіці. Вони представляють поверхні геометричних об'єктів у вигляді набору плоских полігонів, що з'єднані між собою. Традиційний опис полігональної моделі в комп'ютерній графіці є ієрархічним і містить список вершин, ребер та полігонів об'єкта. Основним недоліком полігональних моделей є необхідність створення великої кількості полігонів для відображення складних, особливо криволінійних поверхонь. Реалістичне відтворення просторової сцени може вимагати мільйон і більше полігонів. Це означає, що при синтезі динамічних зображень геометричні параметри потрібно обчислювати з великої кількості примітивів у режимі реального часу.

Загальний вигляд зображення, побудованого з використанням описаної моделі, представлено на рис. 1.1. Незважаючи на те, що кількість полігонів у

сучасних 3D моделях може досягати мільйонів, ця модель залишається майже монополістом у сфері комп'ютерної графіки завдяки простоті створення користувачами та графічним процесорам, спеціально розробленим для її використання. Іншим методом створення поверхонь геометричних об'єктів є використання сплайнів. Це більш сучасний підхід, зумовлений зростанням обчислювальних потужностей цифрової техніки.

Сплайни – це криві, які задаються функціями, визначеними на певних проміжках, де кожен з цих проміжків описується поліномом. При створенні поверхонь за допомогою сплайнів, вони складаються з великої кількості кривих, що разом формують цілісне зображення. Основним недоліком сплайнів є їхня висока обчислювальна складність, що робить їх непридатними для використання в реальному часі при 3D моделюванні. Однак сплайни широко застосовуються у фільмопроизводстві, технічному проектуванні та цифровому скульптингу (Digital sculpting).

Більш детально, кожен сегмент сплайна визначається поліномом, який забезпечує гладкість і точність моделі. Це дозволяє досягти високого рівня деталізації та реалістичності візуалізацій. У фільмах сплайни допомагають створювати плавні переходи і динамічні ефекти, в технічному проектуванні – забезпечують точні геометричні форми, а в цифровому скульптингу – дають можливість художникам створювати складні форми та текстури. Приклад зображення побудованого за допомогою цієї технології зображено на рис.1.2.

Описані вище методи створення 3D моделей є способами формування поверхонь. Вони забезпечують високоякісну візуалізацію, але самі об'єкти втрачають цілісність. Такі об'єкти складаються з елементарних одиниць і насправді є колекціями кривих або полігонів. В результаті, операції зі зміни одного об'єкта перетворюються на безліч операцій зі зміни групи елементів, що значно підвищує обчислювальну складність і ставить під сумнів можливість використання в реальному часі. Часто зустрічаються об'єкти, особливо природні, які мають досить складну форму, що ускладнює створення універсального аналітичного опису. Їх форма визначається набором характерних (опорних) точок, розташованих на поверхні об'єкта. Наприклад, це може бути складена

геодезистами карта висот ділянки земної поверхні. У процесі геометричного моделювання необхідно відновити вихідну поверхню з заданою точністю, так щоб вона проходила якомога ближче до опорних точок або навіть через них.

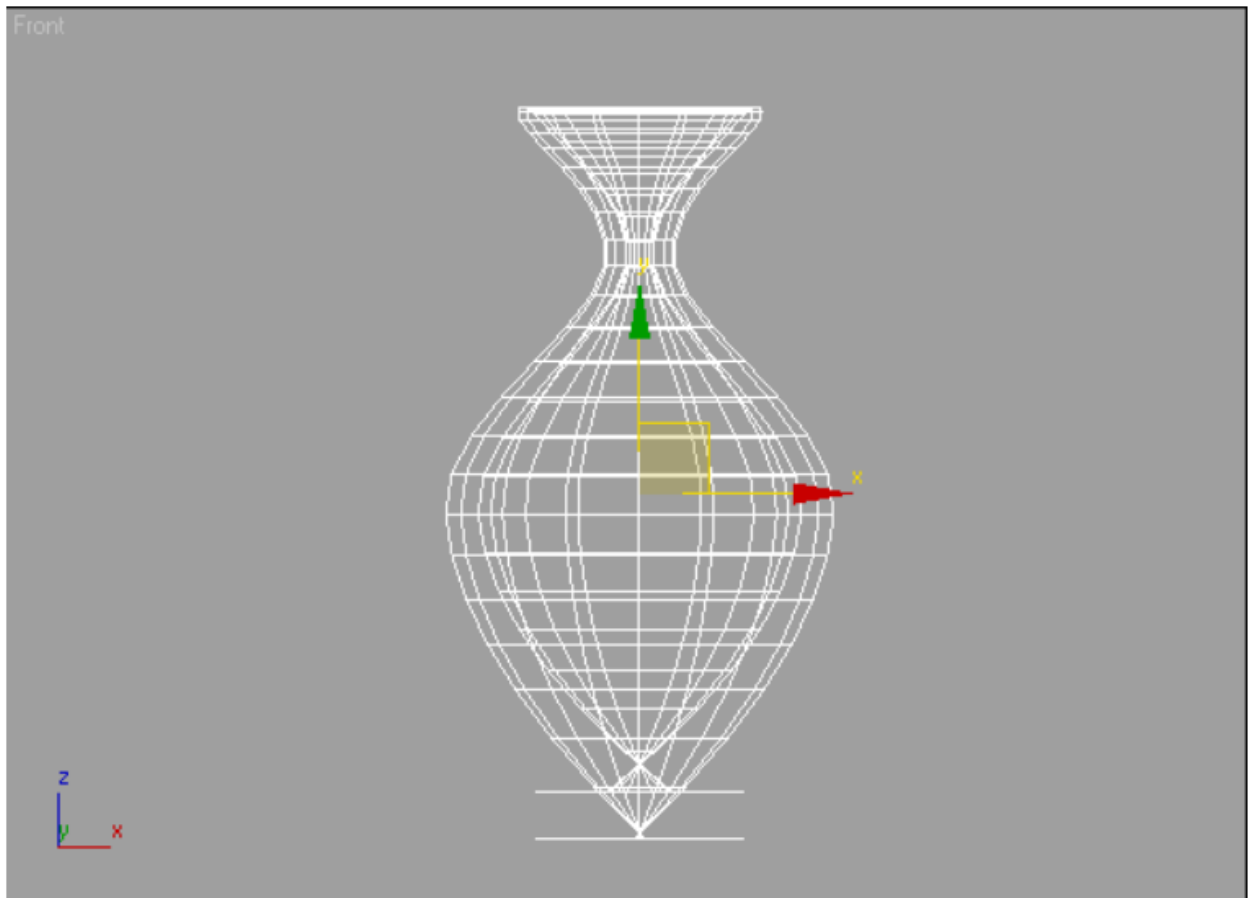


Рисунок 1.2 – Приклад побудови за допомогою сплайнів

Фрактальні ландшафти можуть бути створені за допомогою фракталів, використовуючи метод випадкового зміщення середньої точки. На рисунку 1.3 наведено приклад цього процесу. Спочатку середні точки сторін трикутника (a) зміщуються вгору або вниз від площини зображення і з'єднуються з вершинами (b). Це призводить до утворення чотирьох менших трикутників, до яких знову застосовується та ж сама процедура. Величина зсуву визначається функцією розподілу ймовірності, що впливає на ступінь гладкості фрактального ландшафту. Потім графічна програма комп'ютера зафарбовує трикутники, створюючи різні відтінки (d). В результаті отримується дуже реалістичний зображення (e). Цей метод дозволяє створювати детальні та природні ландшафти, що вражають своєю реалістичністю та складністю.

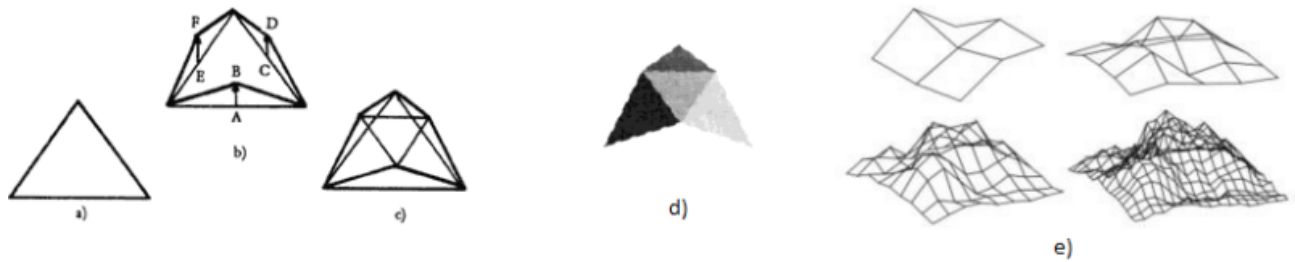


Рисунок 1.3 – Побудова фрактального рельєфу.

Фрактали представляють собою відносно простий метод створення різноманітних складних структур, які мають властивість масштабованості. Водночас, фрактальна модель є ситуативною і залежить від конкретних умов. Справді, у природі часто можна знайти об'єкти, які можуть бути описані за допомогою фракталів, такі як гори, річки, морські узбережжя, рослини тощо. Однак для більшості природних об'єктів побудова фрактальної моделі є неможливою, тому ця технологія розглядається лише як додаток до інших методів.

Хмари точок — це відносно нова технологія, що виникла завдяки розвитку цифрової техніки. Вони являють собою результати роботи 3D-сканерів і мають широке застосування, зокрема для створення 3D-моделей деталей, контролю якості та численних застосунків у візуалізації, анімації та рендерінгу. Хоча хмари точок можуть бути безпосередньо відрендерені та перевірені, їх зазвичай не використовують для створення 3D-моделей. Замість цього вони часто перетворюються в полігональні моделі, NURBS-моделі (Non-uniform rational B-spline) або САПР-моделі за допомогою процесу реконструкції поверхні.

Існує багато методів перетворення хмар точок у 3D-поверхні. Зокрема, у статті М. Берже та його колег представлена порівняльна таблиця з 35 методів реконструкції поверхні з хмари точок [8]. Деякі методи, такі як триангуляція Делоне, Альфа-форма та метод поворотних куль (ball pivoting) [9], створюють трикутну або полігональну сітку на основі існуючих вершин хмари точок. Інші методи використовують об'ємні таблиці відстаней або реконструюють неявну поверхню за допомогою алгоритму крокуючих кубиків (marching cubes). Приклад побудови за допомогою описаної моделі зображено на рис.1.4

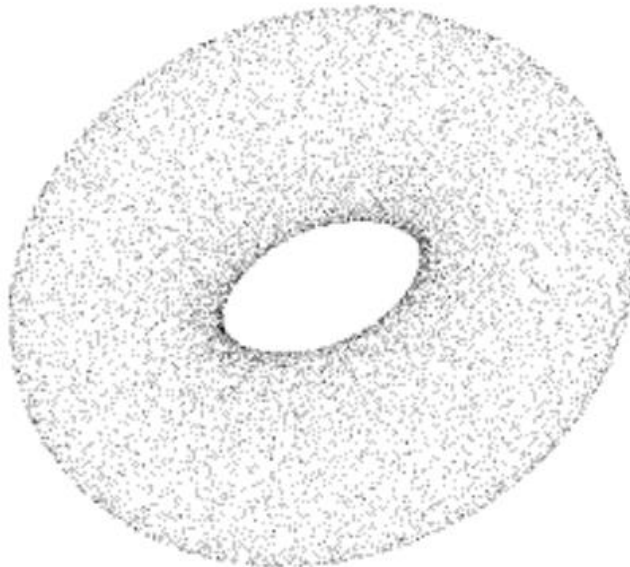


Рисунок 1.4 – Хмара точок моделі тора

Воксельний метод моделювання 3D об'єктів полягає у розбитті простору на маленькі кубики (вокселі), кожен з яких представляє елемент тривимірної матриці. Простота цього підходу є його головною перевагою, але є і суттєвий недолік - величезні потреби у пам'яті для зберігання моделей високої роздільності.

Наприклад, модель з роздільною здатністю 1024x1024x1024 вокселів, де кожен воксель кодується 4 байтами (3 для кольору RGB і 1 для прозорості), вимагатиме цілих 4 Гб пам'яті. Це в рази перевищує обсяги, необхідні для інших методів 3D моделювання.

Тим не менш, воксельна технологія знайшла своє застосування у реальних проектах. Наприклад, у грі Outcast (1999) використовувалася спрощена воксельна модель для створення ландшафту, а персонажі були полігональними. Ще одним прикладом є демонстрація Джона Оліка на SIGGRAPH 2008, де була представлена повноцінна 3D воксельна модель...

Ця воксельна модель, приклад якої зображено на рис. 1.5, вражала своєю деталізацією, складаючись з неймовірних 35 184 372 088 832 вокселів. Примітно, що це було досягнуто на відеокарті NVIDIA GeForce GTX280 з обмеженим обсягом відеопам'яті всього 1 Гб, що свідчить про ефективність використовуваних алгоритмів та оптимізацій.

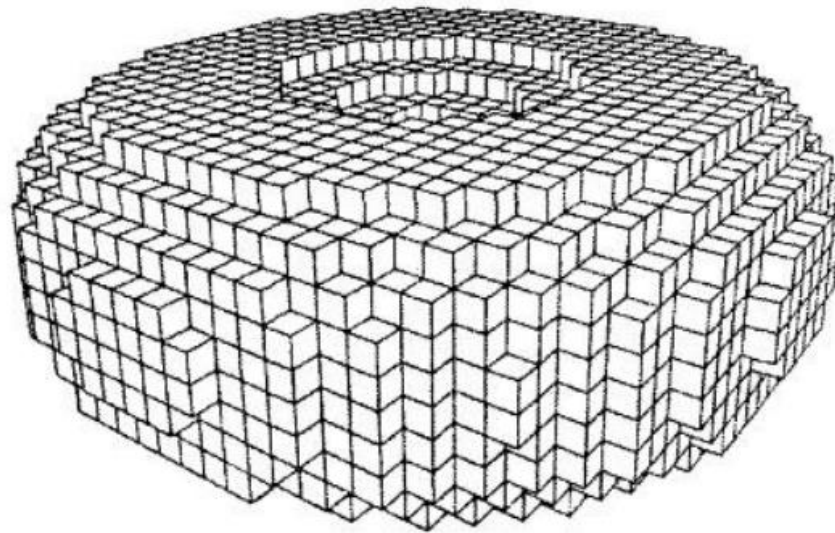


Рисунок 1.5 – Воксельна модель тору

Серед усіх розглянутих моделей, вокселі є найточнішим відображенням реальності, представляючи простір у вигляді тривимірної сітки з кубів. Однак, зберігання інформації про кожну комірку вимагає значних обсягів пам'яті. На щастя, об'єкти рідко складаються з окремих вокселів, і немає потреби зберігати дані про порожній простір.

Для ефективного зберігання та побудови воксельних моделей використовуються спеціальні структури даних. Однією з найперспективніших є розріджене воксельне октодерево (SVO), що забезпечує високу деталізацію при оптимізованому використанні пам'яті.

Октодерево - це ієрархічна структура даних, схожа на файлову систему, де корінь містить гілки (папки), які, своєю чергою, можуть містити інші гілки (вкладені папки), доки не дійдемо до "листя" (файлів). Така організація дозволяє швидко знаходити потрібні дані.

Кожен вузол октодерева може мати до восьми гілок. Це дозволяє ефективніше використовувати пам'ять, деталізуючи лише ті області, де це необхідно. Для ілюстрації принципу побудови октодерева розглянемо квадродерево на площині.

Стандартний підхід (рис. 1.6(a)) передбачає розбиття площини на однакові комірки та заповнення тих, що належать фігурі. Квадродерево ж (рис. 1.6(b)) починає з базового зображення, яке ділиться на чотири квадранти. Якщо квадрант порожній або повністю заповнений, поділ припиняється. Інакше,

квадрант ділиться далі, поки всі квадранти не стануть однорідними або не буде досягнуто заданої глибини.

Навіть у цьому простому прикладі квадродерево дає результат, ближчий до оригіналу, використовуючи менше даних. Октодерево переносить цей принцип у тривимірний простір, забезпечуючи ефективне представлення воксельних моделей.

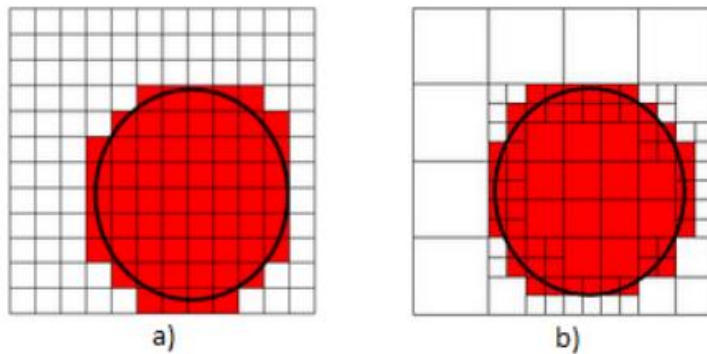


Рисунок 1.6 – Варіанти заповнення

На практиці, вираш у використанні пам'яті при застосуванні октодерев не такий значний, як може здатися на перший погляд. Це пояснюється тим, що кожен вузол октодерев повинен зберігати посилання на свої дочірні елементи, що вимагає додаткової пам'яті порівняно зі звичайною координатною сіткою, де розташування вокселів визначено заздалегідь.

Проте, октодерев мають ряд переваг при побудові воксельних моделей. По-перше, вони дозволяють ефективно виконувати прокладку променів (ray casting), що є швидшим методом, ніж трасування променів (ray tracing). Октодерево одночасно слугує сховищем даних (геометрія та текстури) і структурою прискорення обчислень перетинів.

По-друге, октодерев спрощують вирішення проблеми рівня деталізації (LOD) як для геометрії, так і для текстур. Оскільки кожен вузол містить кольорову інформацію, можна обійтися без традиційних 2D-текстур. Рівень деталізації вибирається автоматично залежно від розміру вокселя відносно розміру пікселя на екрані, що ідеально підходить для потокових систем візуалізації.

Однак, потокові системи візуалізації, включаючи воксельні октодерева, мають спільний недолік: вони погано справляються з раптовими змінами сцени. При різкому переході в іншу частину сцени може знадобитися час для завантаження необхідних даних, що призведе до тимчасового відображення низькодеталізованої геометрії. Хоча ця проблема не унікальна для октодерев, вона є більш помітною у випадку воксельних моделей, де низькодеталізоване октодерево виглядає особливо непривабливо.

1.3. Вибір технічного забезпечення для розробки та 3-D моделювання сцен

Програмне забезпечення для 3D-моделювання охоплює програми, що створюють базові тривимірні моделі об'єктів або персонажів, надаючи додаткові функції для деталізації моделей з реалістичними елементами. Ці продукти використовують різні підходи та інструменти для створення моделей, часто включаючи можливості 3D-малювання для нанесення текстур і кольору. Такі програми знаходять застосування не лише в індустрії відеоігор, але й у телебаченні, кіно, маркетингу та віртуальній реальності. Після створення моделей за допомогою програмного забезпечення для 3D-моделювання, їм можна надати життя за допомогою інструментів 3D-рендерінгу та 3D-анімації. Графічні дизайнери можуть використовувати грубі 3D-моделі без візуалізації для візуального оформлення веб-сайтів і медіа-матеріалів. Інструменти з можливостями 3D, спеціально призначені для проектування та архітектури будівель, цивільного будівництва чи інших функцій, пов'язаних із САПР, не розглядаються в цьому розділі, оскільки вони не підходять для створення 3D-моделі, необхідної для бакалаврської кваліфікаційної роботи.

Індустрія 3D-моделювання постійно розвивається під впливом різних факторів. Одним з ключових факторів є зростання попиту на якісний та деталізований дизайн. Споживачі очікують 3D-моделей, які не лише реалістично виглядають, але й враховують всі дрібниці. Це суттєво відрізняє професійний дизайн від посереднього.

Варто також відзначити, що індустрія 3D-анімації та рендерингу переживає стрімкий розвиток. Популярні потокові сервіси та дизайнерські фірми все частіше потребують 3D-графіку VFX та 3D-моделі для дизайну продуктів. Це стимулює розробку нових інструментів та програмного забезпечення, що робить процес 3D-моделювання більш доступним та ефективним.

Maya

Maya – це не просто програмне забезпечення для 3D-моделювання, а й потужний інструмент, який активно використовується в індустрії 3D для створення реалістичного та динамічного контенту. Завдяки широкому набору інструментів для 3D-моделювання, анімації та візуалізації, Maya дозволяє втілювати в життя найскладніші задуми.

Одним з ключових факторів успіху Maya є її універсальність. Програмне забезпечення підтримує як візуальне програмування Vifrost, так і традиційні методи, що робить його ідеальним для роботи з волоссям, частинками, тканинами та іншими елементами, значно полегшуючи процес анімації. Крім того, Maya дає можливість перемикатися між CPU та GPU режимами, що дозволяє їй ефективно справлятися з найскладнішими завданнями, будь то створення об'ємних та реалістичних 3D-світів або формування 3D-об'єктів та сцен. Maya доступна для Windows, Linux та macOS, що робить її доступною для широкого кола користувачів.

Звичайно, Maya має й свої недоліки. Її вартість може здатися високою, а при роботі з великими проектами можуть виникати проблеми з продуктивністю. Проте, враховуючи широкий спектр можливостей та інструментів, Maya залишається одним з кращих виборів для професіоналів у сфері 3D-дизайну, візуальних ефектів та анімації.

Autodesk 3Ds Max

Autodesk 3Ds Max – це одне з найкращих програмних забезпечень для 3D-моделювання на платформі Windows, яке пропонує широкий спектр функцій для кіноіндустрії, архітектури, дизайну продуктів та інших сфер. Його переваги роблять його лідером серед професіоналів. 3Ds Max пропонує вражаючі методи прямого маніпулювання та моделювання, а також багату бібліотеку 3D-ресурсів,

що робить його універсальним інструментом для різних задач. Програмне забезпечення забезпечує швидку та плавну роботу, він порівняно простий у освоєнні. Це робить його доступним як для новачків, так і для досвідчених користувачів, на відміну від Maya. 3Ds Max відомий своїми вражаючими можливостями візуалізації та ефектами затінення, що робить його ідеальним для створення реалістичних зображень та анімації.

Однак, як і будь-яке програмне забезпечення, 3Ds Max має й недоліки. Програма може бути досить ресурсомісткою, тому для її комфортного використання потрібна потужна система. Деякі користувачі вважають інтерфейс 3Ds Max дещо застарілим та складним для навігації. Робота над 2D-проектами в цій програмі може бути менш зручною, порівняно з спеціалізованим програмним забезпеченням.

ZBrush

ZBrush – це потужна програма для 3D-моделювання, яка дозволяє створювати вражаючі скульптури та фігурки. Завдяки інтуїтивному інтерфейсу та різноманітним пензлям, він дозволяє створювати органічні форми з неймовірною деталізацією. Окрім ліплення, ZBrush також чудово підходить для створення жорстких поверхонь та складних моделей. Функціональність програми не обмежується лише геометрією. Завдяки інструментам для текстурювання та створення UV-карт, ви можете додати своїм моделям реалістичні текстури та тканини, роблячи їх візуально вражаючими. ZBrush спеціально розроблений для 3D-друку, пропонує безліч інструментів для підготовки моделей до друку, таких як оптимізація сітки, додавання опор та автоматичне виявлення помилок.

ZBrush дозволяє створювати моделі з неймовірною деталізацією та реалістичністю, які неможливо досягти за допомогою інших програм. Інструменти ZBrush економлять час та зусилля, роблячи процес 3D-моделювання більш ефективним.

Однак, ZBrush не є найкращим вибором для ригінгу та анімації персонажів. Для цих завдань краще використовувати інші програми. ZBrush – це досить дорога програма, що може бути недоступною для деяких художників.

Blender

Blender – це безкоштовний 3D-додаток, який пропонує широкий спектр функцій для моделювання, текстурування, анімації та візуалізації. Завдяки своїм можливостям та активній спільноті розробників, Blender став популярним вибором як для новачків, так і для досвідчених 3D-художників.

Blender може похвалитися вражаючим набором інструментів для 3D-моделювання та скульптури, що робить його життєздатною альтернативою платним програмам. Завдяки широкому спектру функцій, Blender підходить для різноманітних VFX та завдань моделювання, роблячи його цінним активом у будь-якому 3D-робочому процесі.

Останніми роками Blender стає все більш популярним у великих студіях, що свідчить про його здатність відповідати потребам професійних 3D-художників. Інтерфейс Blender може здатися складним для початківців, але з часом до нього можна звикнути.

Mixamo

Mixamo пропонує потужне рішення для 3D-анімації, надаючи величезну бібліотеку високоякісних 3D-персонажів та сотні анімацій. Хоча Mixamo не є повноцінним 3D-редактором, він вирізняється зручністю автоматичної реалізації скелету та надання готових анімацій для персонажів, які можна експортувати в різних форматах до популярних ігрових рушіїв та Blender.

Розглянуті приклади програмного забезпечення є лідерами на ринку 3D-моделювання, мають багаторічну історію та велику спільноту користувачів та розробників. Вони всі задовольняють базові потреби 3D-моделювання, але мають свої сильні та слабкі сторони, включаючи функціонал, зручність використання та ціну.

Щоб визначити, яке програмне забезпечення найкраще підходить для конкретного завдання, необхідно порівняти їх за ключовими критеріями, представленими у таблиці 1.1.

Аналіз даних з Таблиці 1.1 привів до вибору Blender як основного програмного забезпечення для розробки 3D-моделей. Blender - це потужний і універсальний редактор, що відповідає всім сучасним вимогам 3D-моделювання.

Він працює навіть на комп'ютерах з невисокими технічними характеристиками, має велику спільноту користувачів та розробників, які створюють численні навчальні матеріали. Але найважливіше, Blender є повністю безкоштовним і з відкритим кодом, що не заважає йому конкурувати за якістю з іншими комерційними редакторами.

Таблиця 1.1. – Характеристика програмного забезпечення для розробки 3D моделей

Програмне забезпечення для 3D-моделювання	Maya	Autocad 3ds Max	ZBrush	Blender
Ключові особливості	Обробка захоплення руху, моделювання nurbs, шари	Мова скриптів, модифікатор редагування полі, сплайн-система	Настроюванні пензлики, шари ліплення	Модифікатор шкіна, комбінації клавіш, доповнення
Найкраще підходить для	Анімація та візуальні ефекти	Моделювання та ригінг	High-poly моделювання, ліплення	Анімації, незалежних проєктів
Складність	Середньо	Середньо	Середньо	Важко

Як додаткове програмне забезпечення було обрано Міхато, який спрощує та автоматизує роботу з анімацією та скелетами персонажів. Міхато також є безкоштовним і дозволяє швидко створювати анімації для персонажів.

Висновки до першого розділу

Вимоги до розробок включають можливість створення реалістичних та детальних моделей, які візуально відповідають реальним об'єктам, та забезпечення високої продуктивності гри шляхом оптимізації моделей, текстур, анімацій та інших ресурсів. Алгоритми побудови 3D-сцен включають полігони, сплайни, фрактали, хмари точок та вокселі, кожен з яких має свої переваги та підходить для різних типів завдань. Полігони є найпоширенішим методом, який підходить для створення реалістичних моделей, але потребує багато ресурсів. Сплайни підходять для створення плавних кривих та органічних форм, фрактали

– для складних ландшафтів, хмари точок – для моделей реальних об'єктів, а вокселі – для процедурно-генерованих моделей.

Вибір методу 3D-моделювання залежить від поставлених завдань, ресурсів та необхідного рівня деталізації. Актуальні дослідження спрямовані на розробку нових методів оптимізації 3D-моделей для забезпечення високої продуктивності гри, створення нових інструментів для реалістичних 3D-сцен та вивчення можливостей використання нових технологій, таких як штучний інтелект та віртуальна реальність, у 3D-моделюванні.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ МЕТОДИКИ 3-D МОДЕЛЮВАННЯ СЦЕН

2.1. Алгоритми побудови реалістичних 3D-сцен

Розглянемо три алгоритми виявлення та опису ключових точок на зображеннях: SIFT, SURF та ORB. Ці алгоритми відіграють важливу роль у створенні реалістичних 3D-сцен, оскільки дозволяють знаходити та співставляти відповідні точки на різних зображеннях. Проаналізуємо особливості кожного алгоритму та їх дескриптори, щоб у подальшому визначити принципи власного алгоритму.

SIFT (Scale-Invariant Feature Transform)

Алгоритм SIFT виявляє ключові точки на еталонних зображеннях, зберігає їх у базі даних та порівнює з новими зображеннями для розпізнавання об'єктів. Відповідність визначається шляхом обчислення евклідової відстані між векторами ознак. SIFT використовує ефективну хеш-таблицю для швидкого пошуку відповідностей та ретельного фільтрування результатів, щоб забезпечити високу точність розпізнавання.

Процес SIFT можна розбити на такі кроки:

1. **Масштабно-просторове визначення екстремумів:** Виявлення особливостей зображення, таких як кути, за допомогою масштабно-просторової фільтрації та різниці гаусіанів (для оптимізації обчислень).
2. **Локалізація ключових точок:** Уточнення розташування ключових точок за допомогою розширення ряду Тейлора та відкидання точок з низькою інтенсивністю.
3. **Задача координування:** Побудова орієнтаційної гистограми для забезпечення інваріантності до поворотів зображення.
4. **Дескриптор ключових точок:** Створення унікального опису кожної ключової точки.
5. **Узгодження ключових точок:** Порівняння дескрипторів ключових точок між різними зображеннями для знаходження відповідностей.

SURF (Speeded Up Robust Features)

SURF - це покращений алгоритм, що базується на SIFT. Він також використовує різницю гаусіанів, але наближує її до Box Filter, що дозволяє швидше обчислювати згортки та паралельно обробляти різні масштаби зображення.

У наступних розділах ми детальніше розглянемо алгоритми ORB та порівняємо їх з SIFT та SURF, щоб визначити найбільш підходящий для нашого завдання.

Масштабів [1]

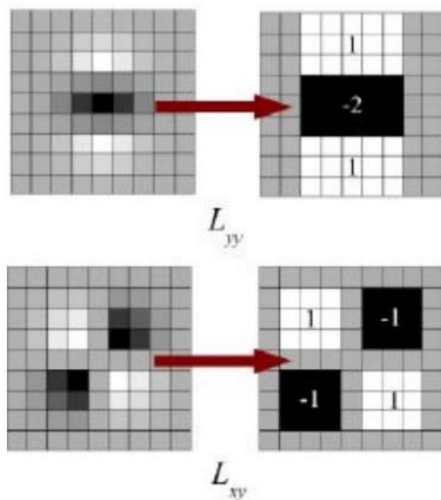


Рисунок 1.2 - Box Filter [1]

Виявлення особливих точок у SURF базується на обчисленні детермінанта матриці Гессе (гессіана), який допомагає знайти локальні екстремуми яскравості зображення. Для цього використовуються спеціальні фільтри, зображені на Рис. 1.2.

Для визначення орієнтації SURF застосовує вейвлет-відповіді в горизонтальному та вертикальному напрямку, з урахуванням гаусових ваг. Цей підхід, відомий як Upright-SURF (U-SURF), прискорює обчислення та забезпечує надійність при поворотах зображення до ± 15 градусів.

Для опису особливостей SURF також використовує вейвлет-відповіді, але вже у вікні розміром 20x20 навколо ключової точки. Це вікно ділиться на менші субрегіони, що прискорює обчислення. Важливим удосконаленням є використання знаку Лапласіана, що дозволяє уникнути зайвих обчислень.

Загалом, SURF можна розглядати як покращений варіант SIFT, який швидше працює з розмитими та повернутими зображеннями, але гірше справляється зі змінами освітлення.

ORB (Oriented FAST and Rotated BRIEF) - це комбінація детектора ключових точок FAST та дескриптора BRIEF з додатковими модифікаціями для підвищення продуктивності. ORB використовує піраміду для створення багатомасштабних ознак та спеціальний алгоритм для визначення орієнтації. За твердженням авторів, ORB значно швидший за SURF та SIFT, а його дескриптор працює краще, ніж у SURF. ORB є гарним вибором для пристроїв з обмеженими ресурсами та завдань, де потрібна швидкість, наприклад, для панорамного зшивання.

Хоча SIFT та SURF є ефективними алгоритмами, вони запатентовані та не можуть бути використані безкоштовно. Крім того, існують інші підходи, такі як метод на основі лісу рандомних дерев, які можуть давати кращі результати, але вимагають більше пам'яті та часу. ORB є швидкою та безкоштовною альтернативою, що робить його привабливим вибором для багатьох застосувань.

Враховуючи все вищесказане, можна зробити висновок, що існує багато ефективних алгоритмів виявлення та опису ключових точок. Ми можемо використовувати їх безпосередньо або запозичити деякі ідеї для створення власної реалізації.

2.2. Методика 3-D моделювання сцен для комп'ютерних ігор

Основною метою 3D-моделювання сцен для комп'ютерних ігор є створення віртуальних світів, які є візуально привабливими, реалістичними та інтерактивними. Ці віртуальні світи слугують основою для ігрового геймплею, надаючи гравцям середовище, в якому вони можуть досліджувати, взаємодіяти та виконувати завдання.

Методика – це система взаємопов'язаних правил, прийомів та принципів, що використовуються для досягнення певної мети в певній галузі діяльності. Вона охоплює послідовність дій, технологічні підходи та інструменти, які

використовуються для виконання конкретних завдань. Методика базується на теоретичних знаннях та практичному досвіді, що дозволяє систематизувати та оптимізувати процес роботи [26].

3D-моделювання – це процес створення тривимірних цифрових моделей об'єктів. Ці моделі використовуються в різних сферах, таких як комп'ютерна графіка, анімація, дизайн, інженерія та медицина [28].

Розглянемо базову методіку 3D-моделювання описану у працях Akenine-Möller, T., Haines, E., Jensen, H. W («Computer graphics: Principles and practice»), [1] Watt, A., Watt, M. («Advanced 3D graphics with OpenGL»). [21] та Hughes, J. F., Van Dam, A. («3D computer graphics: Principles and practice»). [11] Дослідники виділяють наступні етапи:

1. Планування – визначення мети моделювання; створення ескізів або концептуальних зображень, які описують модель; визначення технічних обмежень та вимог.

2. Моделювання – полягає у виборі відповідного 3D-моделюючого програмного забезпечення; створенні базової геометрії моделі; додаванні деталей та уточнення форми.

3. Текстурування – цей етап включає застосування текстур до моделі для створення реалістичного вигляду. Вибір та налаштування текстурних карт. Створення карт нормалей та інших карт деталей.

4. Освітлення – додавання джерел світла до сцени; налаштування параметрів освітлення для досягнення бажаного ефекту; використання тіней та відблисків для додання реалістичності.

5. Рендеринг – полягає у виборі типу рендерингу (наприклад, растровий, векторний); налаштуванні параметрів рендерингу для досягнення бажаної якості та збереженні зображення або анімації [24].

Виходячи із вищезначеного, визначимо поняття «методики 3D-моделювання сцен для комп'ютерних ігор» як набір правил і принципів, які використовуються для створення реалістичних, візуально привабливих та оптимізованих для ігрових двигунів 3D-моделей.

Ця методика включає в себе наступні етапи:

1. Планування та концептуалізація. На цьому етапі чітко формулюється мета створення 3D-сцени. Це може бути створення певного ігрового середовища, візуалізація архітектурного проекту або розробка анімаційного ролика. Створюється ескіз, описи, мудборди та інші візуальні матеріали, які описують загальний вигляд та атмосферу сцени. Це допомагає візуалізувати кінцевий результат і визначити ключові елементи сцени. Визначаються можливості ігрового двигуна, який буде використовуватися для реалізації сцени. Це дозволяє визначити оптимальний рівень деталізації та складності 3D-моделей.

2. Створення 3D-моделей. Під час другого етапу вибираються відповідні програмні інструменти для моделювання, текстурювання та анімації. Створюються 3D-моделі всіх елементів, які повинні бути присутніми в сцені, включаючи персонажів, будівлі, меблі, декор та інші деталі. До 3D-моделей додаються текстури, матеріали та освітлення, які надають їм реалістичного вигляду та естетичної привабливості. В нашій роботі було використано Blender.

3. Наповнення сцени. 3D-моделі розміщуються в сцені відповідно до концепції, звертаючи увагу на композицію, баланс та візуальну ієрархію. Якщо сцена містить ландшафт, він створюється за допомогою 3D-моделювання, текстурювання та скульптування. Додаються візуальні ефекти, такі як освітлення, тіні, туман, дим та інші, щоб зробити сцену більш реалістичною та атмосферною.

4. Оптимізація та тестування. Кількість полігонів у 3D-моделях зменшується для покращення продуктивності гри без шкоди для візуальної якості. Текстури оптимізуються для зменшення їх розміру та покращення продуктивності. Перевіряється її працездатність, візуальний ефект та продуктивність. За потреби проводяться додаткові налаштування освітлення, текстур, ефектів та інших параметрів для досягнення оптимального результату.

Авторська методика розроблена з урахуванням досвіду автора в 3D-моделюванні та розробці ігор. Її мета – надати чіткий та структурований підхід до створення 3D-сцен, який може використовуватися як професіоналами, так і початківцями.

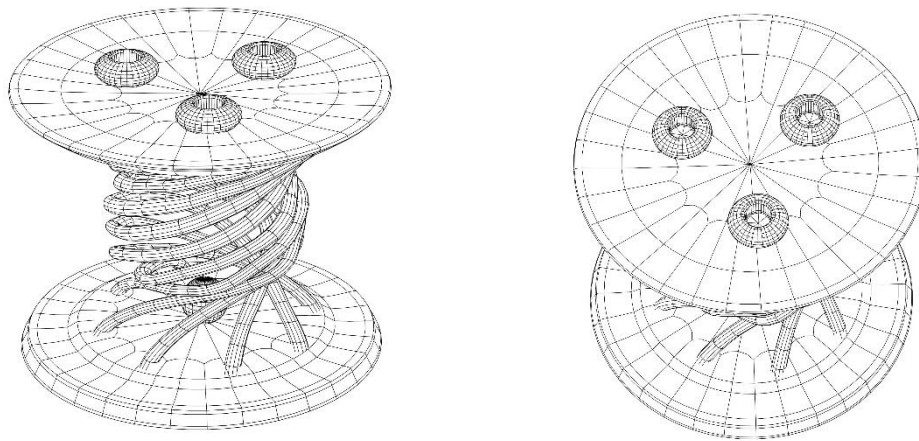
Ця методика може бути використана для створення 3D-сцен для різних жанрів комп'ютерних ігор, таких як шутери, RPG, симулятори та багато інших.

2.3. Процес реалізації 3-D сцени для комп'ютерних ігор за авторською методикою

У цьому підрозділі опишемо процес створення 3D-сцени для комп'ютерних ігор за авторською методикою. Ця методика охоплює всі аспекти створення реалістичних та атмосферних 3D-сцен, від планування та концептуалізації до оптимізації та тестування.

1. Планування та концептуалізація

На цьому етапі було визначено мету створення 3D-сцени — візуалізацію затишного куточка з природними елементами в інтер'єрі. Було створено ескіз, що описує загальний вигляд та атмосферу сцени.



Концептуальний дизайн. Сцена повинна була мати затишну та природну атмосферу. Сцена зображала невеликий інтер'єрний садок з меблями та рослинами. На передньому плані розміщено стіл з голубом, а на задньому плані — вікна з жалюзі та поличка зі свічкою. Загальна атмосфера теплої, спокійної та затишної обстановки. Визначено ключові елементи: стіл з голубом, рослини, вікна з жалюзі, поличка зі свічкою, стіна з текстурою, підлога з травою, декор у вигляді каменів.

Технічні можливості. Сцена була розроблена для використання в ігровому двигуні Unity, що дозволяло використовувати 3D-моделі середньої деталізації та текстури невисокої роздільної здатності.

2. Створення 3D-моделей. У якості основного ПЗ використано Blender. Базова форма столу була створена за допомогою геометрії примітивів, таких як куб і циліндр. За допомогою інструментів скульптування та редагування вершин

була додана деталізація, така як закруглені краї, виступи та текстура дерева. Базова форма свічки була створена за допомогою циліндра. За допомогою інструментів скульптування була додана деталізація, така як нерівна поверхня та гніт. Базова форма підставки була створена за допомогою циліндра. Використано готові 3D-моделі рослин з бібліотеки Blender. Створено прямокутну раму вікна з прозорим склом.

До всіх 3D-моделей додано текстури, що імітують дерево, метал, скло та інші матеріали. Для столу використовувалась текстура дерева з високою роздільною здатністю, що імітує натуральне дерево. Для свічки використовувалась текстура воску з глянцем. Використано UV-розгортку для оптимального розміщення текстур на 3D-моделях.

Додано освітлення, щоб підкреслити форму та текстуру 3D-моделей. В сцені використовувалось декілька джерел світла, включаючи точкове світло, що імітує свічку, та м'яке сонячне світло, що проникає через вікно.

3. Наповнення сцени. Моделі столу, свічки та підставки були розміщені в сцені відповідно до ескізу. Додано фонову текстуру, що імітує стіну та підлогу. Стіл розміщено в центрі сцени, на ньому сидить голуб. Поличку зі свічкою розміщено на задньому плані. Рослини розміщено навколо столу та по кутках кімнати. Вікна з жалюзі розташовані на стіні з боків сцени. Підлога сцени покрита текстурою трави. Додано декілька каменів та інших елементів декору для створення більш реалістичного вигляду.

Додано м'яке сонячне світло, що проникає через вікно, а також точкове світло від свічки для створення затишної атмосфери. Додано тіні від 3D-моделей та легкий туман для створення атмосфери.

4. Оптимізація та тестування. Кількість полігонів у 3D-моделях була оптимізована для покращення продуктивності без шкоди для візуальної якості. Текстури були оптимізовані за розміром та зменшені до мінімального необхідного рівня. Сцена була протестована в Blender, щоб переконатися, що вона візуально приваблива та має гарну продуктивність.

Висновки до другого розділу

У цьому розділі було описано процес створення 3D-сцени для комп'ютерних ігор за авторською методикою. Ця методика охоплює всі аспекти створення реалістичних та атмосферних 3D-сцен, від планування та концептуалізації до оптимізації та тестування.

Авторська методика складається з чотирьох основних етапів:

1. Планування та концептуалізація: на цьому етапі визначається мета сцени, створюється ескіз та опис, а також визначаються технічні можливості.

2. Створення 3D-моделей: на цьому етапі створюються 3D-моделі всіх елементів сцени, додаються текстури, матеріали та освітлення.

3. Наповнення сцени: на цьому етапі 3D-моделі розміщуються в сцені, додається фон, декор та інші елементи.

4. Оптимізація та тестування: на цьому етапі оптимізується кількість полігонів, текстури та інші параметри для покращення продуктивності, а також тестується сцена.

Її переваги полягають у чіткому розбитті процесу створення 3D-сцени на етапи, що полегшує роботу та робить її більш зрозумілою. Вона є універсальною, й може бути використана для створення 3D-сцен для різних жанрів комп'ютерних ігор. Під час реалізації проєкту, було доведено що запропонована методика дозволяє створювати реалістичні та атмосферні 3D-сцени з високою продуктивністю.

РОЗДІЛ 3. ТЕСТУВАННЯ МЕТОДИКИ 3-D МОДЕЛЮВАННЯ СЦЕН

3.1. Тестування 3D-моделей сцен

Тестування 3D-моделей сцен, створених за розробленою методикою, стало ключовим етапом у нашій роботі. Цей комплексний процес дозволив переконатися, що створені сцени відповідають всім поставленим вимогам та очікуванням, а також гарантують захоплюючий досвід для гравців.

Було здійснено кілька видів тестування.

Візуальне тестування. Полягало у порівнянні відповідності зовнішнього вигляду готових сцен з концептуальним дизайном та ескізом, адже саме вони лягли в основу візуального стилю проекту. Аналіз пропорцій допоміг переконатися, що розміщення та співвідношення 3D-моделей в межах сцени дозволяє створити візуально приємну та реалістичну картину. Здійснили перевірку налаштувань освітлення та тіні, щоб досягти бажаного ефекту та підкреслити красу сцен. Проаналізували відповідність текстур масштабу 3D-моделей, адже саме вони надають глибини та деталізації віртуальному світу.

Технічне тестування. Провели тестували сцени на різних конфігураціях комп'ютерів, щоб гарантувати оптимальну продуктивність та плавну роботу гри на будь-якому ПК. Переконалися, що 3D-моделі та сцена не містять помилок або проблем з геометрією, щоб уникнути візуальних артефактів або збоїв під час гри. 3D-моделі та текстури пройшли через процес оптимізації, щоб досягти балансу між якістю візуалізації та ресурсомісткістю. Здійснили перевірку сумісності з ігровим двигуном. Протестували сцени на наявність візуальних артефактів або помилок рендерингу, щоб забезпечити гравцям бездоганний візуальний досвід.

Функціональне тестування. Полягало у перевірці правильної роботи та відповідності задуму усіх елементів сцен. Переконалися, що сцени відповідають всім ігровим механікам та геймплею, адже саме вони роблять гру захоплюючою та динамічною.

Користувацьке тестування. Розроблені моделі були використанні в комп'ютерній грі «Burn the Witch». Це дозволило ідентифікувати потенційні проблеми та вдосконалити сцени, щоб зробити їх максимально захоплюючими для цільової аудиторії.

Проведення тестування на всіх етапах розробки 3D-моделей сцен дозволило: виявити та виправити помилки, проблеми з геометрією, візуальні артефакти та інші недоліки на ранніх стадіях розробки, що значно економило час та ресурси; досягти високого рівня візуальної якості, продуктивності та функціональності сцен.

3.2. Аналіз результатів

Отримані результати тестування авторської методики 3D-моделювання сцен для комп'ютерних ігор в комп'ютерній грі «Burn the Witch», свідчать про її ефективність та дозволяють зробити наступні висновки: використання високоякісних 3D-моделей, створених за допомогою запропонованої методики, значно покращило візуальну складову гри «Burn the Witch». Завдяки оптимізації ресурсомістких операцій, методика не вимагає потужного обладнання.

Позитивні результати:

1) Швидке створення базової сцени – завдяки використанню розробленої методики вдалося створити базову сцену значно швидше, ніж очікувалося. Це свідчить про її ефективність у генеруванні простих геометричних форм та їх розміщенні. Цей результат підтверджує, що методика дійсно полегшує та прискорює процес створення 3D-сцен, роблячи його більш доступним та економним.

2) Чітка структура та логіка – методика чітко структурована та логічно побудована, що робить її зрозумілою та простою у використанні. Це полегшує процес навчання та вдосконалення навичок роботи з 3D-моделюванням, навіть для початківців.

3) Універсальність – методика може бути застосована для створення різних типів 3D-сцен, що робить її універсальним інструментом для розробників ігор, дизайнерів та інших фахівців, які працюють з візуальним контентом.

Напрямки для покращення:

- **Матеріали:** додавання різноманітних матеріалів (дерево, тканина, метал, пластик, скло тощо) значно підвищить реалістичність сцен. Це зробить їх більш візуально привабливими та захоплюючими для глядачів.
- **Освітлення:** використання складніших типів освітлення (сонячне, точкове, об'ємне, динамічне) дозволить створити більш виразну атмосферу та підкреслити певні елементи сцени. Це допоможе краще передати емоції та задум авторів.
- **Текстури:** застосування високоякісних текстур з відповідною роздільною здатністю та деталізацією значно покращить візуальне сприйняття сцен. Це зробить їх більш фотореалістичними та приємними для очей.
- **Деталізація:** додавання деталей, таких як меблі, декор, особисті речі, рослини тощо, зробить сцену більш цікавою та живою. Це допоможе створити відчуття життєвого простору та занурити глядачів у віртуальний світ.
- **Анімація:** додавання анімації елементів сцени (рухомі механізми, динамічні об'єкти, рух персонажів) зробить її більш динамічною та захоплюючою. Це допоможе краще розповісти історію та залучити глядачів.
- **Звук:** додавання звукових ефектів (шум вітру, спів птахів, кроки людей тощо) створить більш реалістичну та атмосферну сцену. Це допоможе краще занурити глядачів у віртуальний світ та посилити їхні емоції.

Вдосконалення цих аспектів дозволить створювати не лише базові, але й фотореалістичні та емоційно насичені 3D-сцени, які справді захоплюватимуть та вражатимуть глядачів.

Висновки до третього розділу

Тестування 3D-моделей сцен, створених за розробленою методикою, стало ключовим етапом у процесі розробки. Цей комплексний процес дозволив переконатися, що створені сцени відповідають всім поставленим вимогам та очікуванням, а також гарантують захоплюючий досвід для гравців.

Було здійснено кілька видів тестування: візуальне тестування; технічне тестування; функціональне тестування; користувацьке тестування.

Проведення тестування на всіх етапах розробки 3D-моделей сцен дозволило: виявити та виправити помилки на ранніх стадіях розробки, що значно економило час та ресурси; досягти високого рівня візуальної якості, продуктивності та функціональності сцен.

Аналіз результатів показав, що завдяки використанню розробленої методики вдалося створити базову сцену значно швидше, ніж очікувалося.

ЗАГАЛЬНІ ВИСНОВКИ

3D-моделювання сцен для комп'ютерних ігор є складним процесом, який охоплює широкий спектр завдань, включаючи створення 3D-моделей об'єктів, їх текстуровання, освітлення, анімацію та оптимізацію. Цей процес вимагає знань у багатьох областях, включаючи моделювання, текстуровання, освітлення, анімацію та оптимізацію. З розвитком технологій, вимоги до 3D-моделей сцен також зростають – 3D-моделі мають бути реалістичними, деталізованими, інтегровані з ігровим двигуном, а сцени – оптимізовані для забезпечення високої продуктивності гри.

Дослідники та розробники постійно працюють над створенням нових інструментів та методів для 3D-моделювання сцен для комп'ютерних ігор. Розробка методики 3D-моделювання сцен для комп'ютерних ігор може бути використана для підвищення ефективності розробки 3D-моделей, зменшення часу та зусиль, необхідних для розробки візуалізації, та підвищення продуктивності комп'ютерних ігор, які використовують 3D-моделі.

Розроблено чітку та структуровану методику 3D-моделювання сцен для комп'ютерних ігор, яка охоплює всі етапи створення реалістичних та атмосферних 3D-сцен: планування та концептуалізація; створення 3D-наповнення сцени; оптимізація та тестування. Методика базується на кращих практиках 3D-моделювання та враховує особливості розробки ігрових сцен. Вона є універсальною та може бути використана для створення 3D-сцен для різних жанрів комп'ютерних ігор.

За допомогою розробленої методики створено прототип 3D-сцени для комп'ютерної гри "Burn the Witch". Сцена включає в себе реалістичні та атмосферні 3D-моделі меблів, рослин, декору та інших елементів. Моделі оптимізовані для забезпечення високої продуктивності в ігровому двигуні.

Проведено тестування прототипу 3D-сцени, яке підтвердило її візуальну привабливість та високу продуктивність. За результатами тестування проведено оптимізацію 3D-моделей та текстур для покращення продуктивності без шкоди для якості.

Переваги авторської методики полягають у чіткому розбитті процесу створення 3D-сцени на етапи; зрозумілості та простоті використання; універсальності та можливості застосування для різних жанрів ігор; створенні реалістичних та атмосферних 3D-сцен з високою продуктивністю.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Akenine-Möller, T., Haines, E., & Hoffman, N. (2018). Real-time rendering (4th ed.). CRC Press.
2. Bartel, T., Reuter, F., & Doll, A. (2019). Physically based shading and lighting in Filament. Google, Inc.
3. Carey, M. (n.d.). 3D Modeling and Rendering with Blender. <https://www.amazon.com/blender-3d/s?k=blender+3d>
4. Dunn, F., & Parberry, I. (2011). 3D math primer for graphics and game development (2nd ed.). Course Technology PTR.
5. Eberly, D. H. (2004). 3D game engine architecture: Engineering real-time applications with Wild Magic. Morgan Kaufmann.
6. Eberly, D. H. (2016). 3D Game Engine Design (2nd ed.). Morgan Kaufmann Publishers Inc.
7. Friedman, N. (2015). Catlike coding: An introduction to shader programming. No Starch Press.
8. Glassner, A. (2014). The Art of 3D Computer Graphics and Animation (3rd ed.). Focal Press. <https://www.amazon.com/Art-3D-Computer-Animation-Effects/dp/0470084901>
9. Gregory, J. (2017). Game engine architecture. CRC Press.
10. Gregory, J. (2018). Game Engine Architecture (3rd ed.). CRC Press.
11. Hughes, J. F., Van Dam, A. (2013). 3D computer graphics: Principles and practice. Addison-Wesley Professional.
12. Karis, B. (2013). Real shading in Unreal Engine 4. Epic Games, Inc.
13. Lengyel, E. (2012). Foundations of Game Engine Development, Volume 1: Mathematics. Terathon Software LLC.
14. Lengyel, E. (2012). Foundations of Game Engine Development, Volume 2: Rendering. Terathon Software LLC.
15. Lengyel, E. (2012). Mathematics for 3D Game Programming and Computer Graphics. Third Edition. Course Technology PTR. 552 p.
16. Lengyel, E., (2016). 3D Game Design: A Practical Introduction. Focal Press.
17. Nystrom, R. (2014). Game programming patterns. Genever Benning.

18. Nystrom, R. (2019). *Game Programming Patterns*. Genever Benning.
19. Pharr, M., Jakob, W., Humphreys, G. (2016). *Physically Based Rendering: From Theory To Implementation* (3rd ed.). Morgan Kaufmann Publishers Inc.
20. Watt, A., & Watt, M. (1992). *Advanced animation and rendering techniques: Theory and practice*. Addison-Wesley Professional.
21. Watt, A., Watt, M. (2019). *Advanced 3D graphics with OpenGL*. Pearson Education.
22. Wright, J. (2014). *Unreal Engine 4 Game Development Essentials*. Packt Publishing.
23. Wronski, B. (2016). *GPU Pro 7: Advanced rendering techniques*. CRC Press.
24. Гусєв, О. (2018). *Створення 3D-світів: практичний посібник*. БХВ.
25. Ковальчук, М. (2008). Зв'язок узагальнення з принципом наочності. *Наукові записки Тернопільського національного педагогічного університету. Серія: педагогіка*, (7), 113-117.
26. Ковальчук, М. О. (2015). *Методика створення мультимедійної книги. У Формування дидактичної компетентності педагогів дошкільної та початкової освіти: збірник науково-методичних праць* (с. 398-411). Житомир: Вид-во ЖДУ ім. І. Франка. http://eprints.zu.edu.ua/17595/1/dyd_komp_2015.pdf#page=398
27. Ковальчук, М. О., & Колесник, Н. Є. (2020). *Графічний дизайн та комп'ютерна графіка: монографія*. Житомир: ТОВ, 505, 440.
28. Ковальчук, С. (2019). *3D-моделювання та візуалізація: теорія та практика*. ЛітРес.
29. Кузьменко, А. (2020). *Unreal Engine 4: розробка ігор та віртуальної реальності*. БХВ.
30. Маєвський, О. М., & Ковальчук, М. В. (2023). *Сучасні технології підготовки мультимедійних творчих продуктів. Prospective And Priority Directions Of Scientific Research In Technical And Agricultural Sciences* (с. 191-214). Boston: International Science Group. – Primedia eLaunch.

ДОДАТКИ

Додаток А.

Код для завантаження 3D моделей в комп'ютерну гру

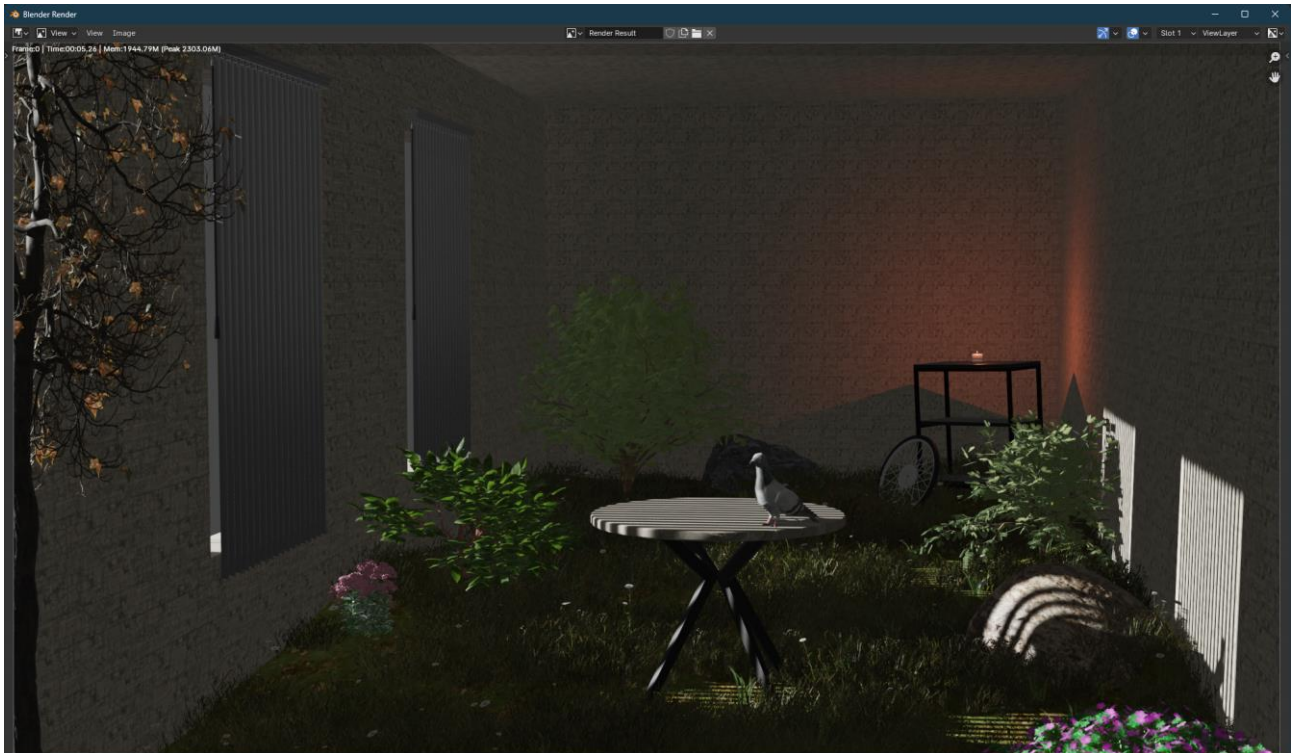
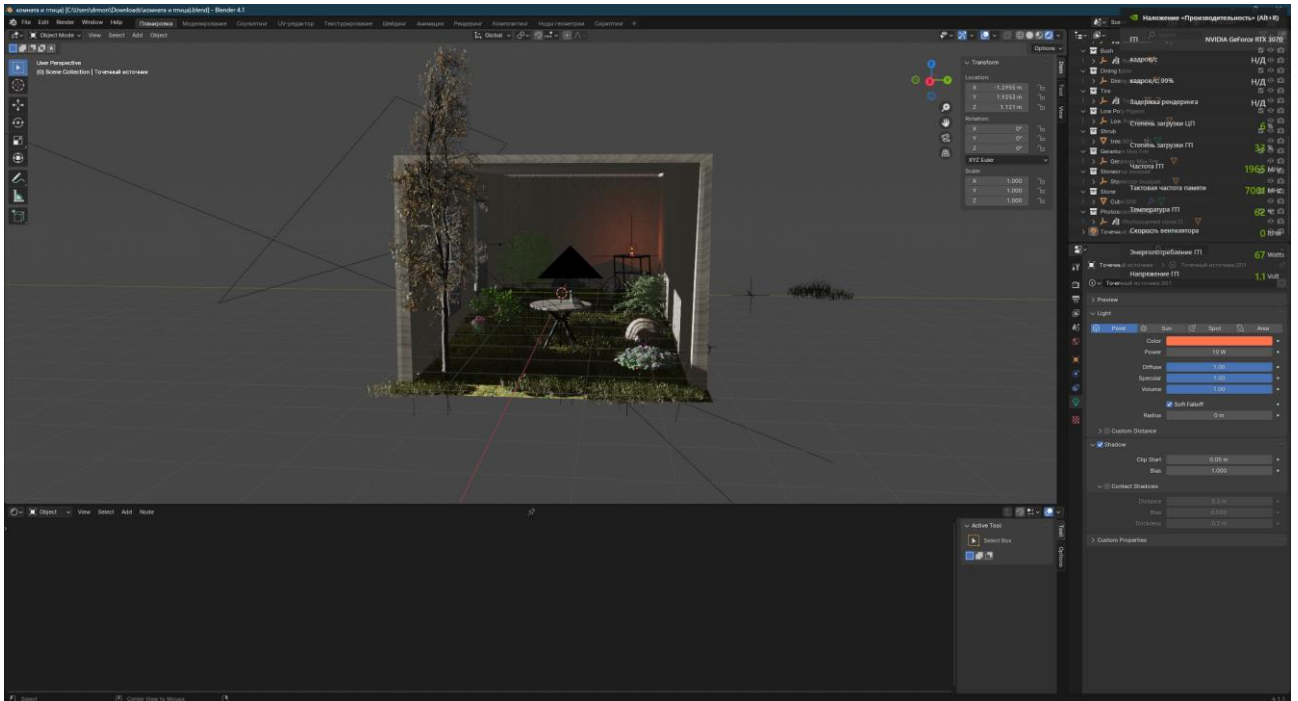
```
1  #include <iostream>
2  #include <vector>
3
4  // Бібліотеки для завантаження 3D-моделей.
5  #include <assimp/Importer.hpp>
6  #include <assimp/scene.h>
7  #include <assimp/postprocess.h>
8
9  using namespace std;
10
11 // Структура для зберігання інформації про 3D-модель.
12 ▾ struct Model {
13     vector<float> vertices; // Вершини 3D-моделі
14     vector<float> normals; // Нормалі вершин
15     vector<float> texCoords; // Текстурні координати
16     vector<unsigned int> indices; // Індеси для побудови трикутників
17 };
18
19 vector<Model> models; // Масив для зберігання 3D-моделей
20
21 // Функція для завантаження 3D-моделі з файлу.
22 ▾ void loadModel(const string& filename) {
23     // Створення екземпляра імпортера Assimp.
24     Assimp::Importer importer;
25
26     // Завантаження 3D-моделі з файлу.
27 ▾ const aiScene* scene = importer.ReadFile(filename,
28     | aiProcess_Triangulate | aiProcess_FlipUVs | aiProcess_GenNormals);
29
30     // Перевірка успішності завантаження.
31 ▾ if (!scene || scene->mFlags & AI_SCENE_FLAGS_INVALID || !scene->mRootNode) {
32     | cerr << "Error loading model: " << importer.GetErrorString() << endl;
33     | return;
34 }
35
```

```

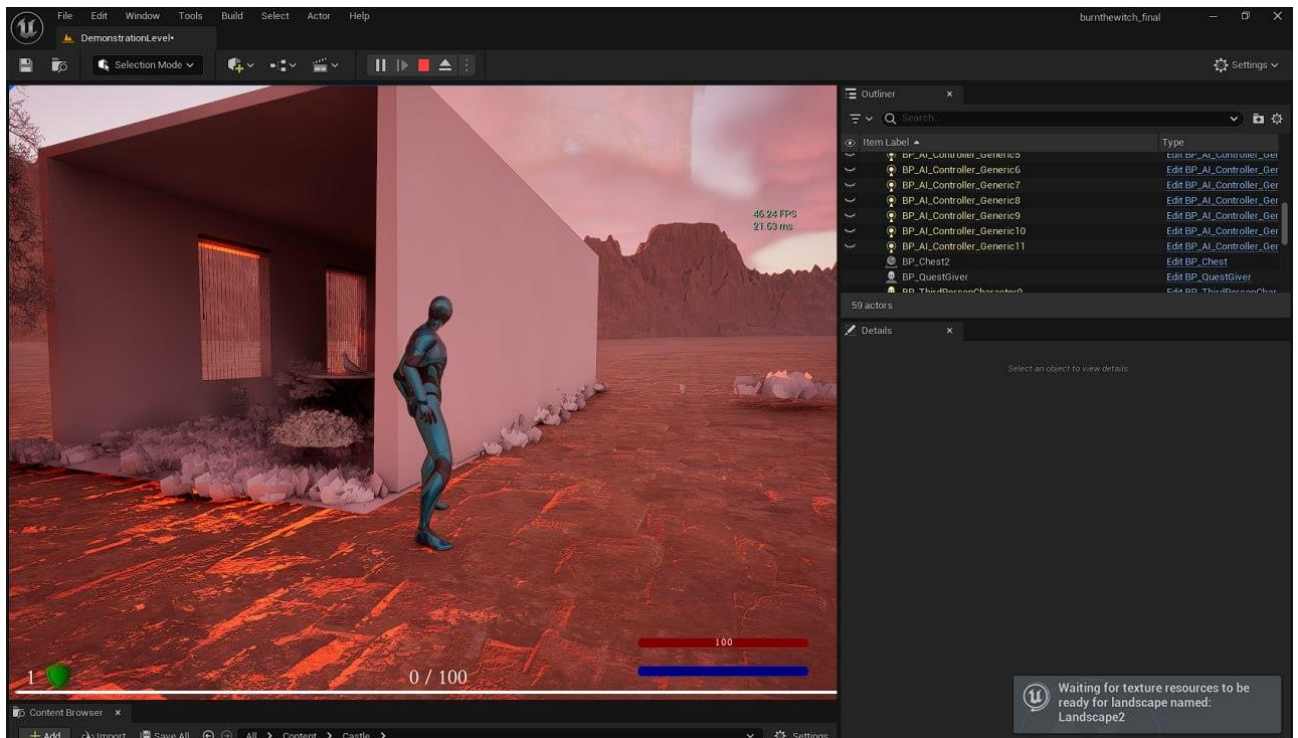
36     // Обробка 3D-моделі.
37     processNode(scene->mRootNode, scene);
38 }
39
40 // Функція для обробки вузла 3D-моделі.
41 void processNode(aiNode* node, const aiScene* scene) {
42     // Отримання 3D-даних з вузла.
43     const aiMesh* mesh = scene->mMeshes[node->mMeshIndex];
44
45     // Обробка вершин.
46     for (unsigned int i = 0; i < mesh->mNumVertices; ++i) {
47         models[currentModel].vertices.push_back(mesh->mVertices[i].x);
48         models[currentModel].vertices.push_back(mesh->mVertices[i].y);
49         models[currentModel].vertices.push_back(mesh->mVertices[i].z);
50     }
51
52     // Обробка нормалей.
53     if (mesh->HasNormals) {
54         for (unsigned int i = 0; i < mesh->mNumVertices; ++i) {
55             models[currentModel].normals.push_back(mesh->mNormals[i].x);
56             models[currentModel].normals.push_back(mesh->mNormals[i].y);
57             models[currentModel].normals.push_back(mesh->mNormals[i].z);
58         }
59     }
60
61     // Обробка текстурних координат.
62     if (mesh->HasTextureCoords[0]) {
63         for (unsigned int i = 0; i < mesh->mNumVertices; ++i) {
64             models[currentModel].texCoords.push_back(mesh->mTextureCoords[0][i].x);
65             models[currentModel].texCoords.push_back(mesh->mTextureCoords[0][i].y);
66         }
67     }
68
69     // Обробка індексів.
70     for (unsigned int i = 0; i < mesh->mNumFaces; ++i) {
71         const aiFace& face = mesh->mFaces[i];
72         for (unsigned int j = 0; j < face.mNumIndices; ++j) {
73             models[currentModel].indices.push_back(face.mIndices[j]);
74         }

```


Реалізація 3D сцени за авторською методикою



Тестування сумісності та інтегрованості 3D моделі сцени з ігровим рушієм



Зображення показує процес тестування сумісності та інтегрованості 3D моделі сцени з ігровим рушієм.

Вікно ігрового рушія: зображено 3D-модель сцени, імпортовану до ігрового рушія; модель розміщена в ігровому середовищі (наприклад, на карті) та взаємодіє з іншими елементами (персонажами, об'єктами); на панелях рушія можна бачити різні налаштування та дані про сумісність та інтеграцію.

У консолі розробника представляється текст логів, повідомлень про помилки та іншу інформацію, що генерується ігровим рушієм під час тестування. Ця інформація може допомогти розробникам виявити та виправити проблеми з сумісністю. На основі даних тестування можна буде виправити помилки та оптимізувати модель для кращої роботи в ігровому середовищі.

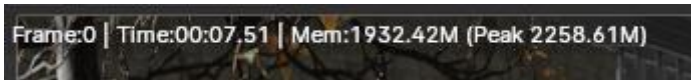
Зображення свідчить про те, що 3D-модель сцени може бути успішно інтегрована в ігровий рушій. Оскільки, на панелі інструментів тестування можна бачити, що не виявлено жодних критичних помилок. А це свідчить про те, що модель в цілому сумісна з ігровим рушієм.

Однак, є кілька попереджень, які рекомендують оптимізувати деякі аспекти моделі. Наприклад, можна зменшити кількість полігонів або покращити

текстури. Консоль розробника показує, що модель правильно анімується та взаємодіє з іншими об'єктами.

Це свідчить про те, що модель може бути використана в геймплеї.

Тестування продуктивності 3D моделі сцени



На панелі інструментів тестування можна бачити, що частота кадрів становить близько 30 кадрів на секунду. Це свідчить про те, що модель візуалізується плавно та без затримок. Однак, час рендерингу одного кадру становить близько 10 мілісекунд. Це може бути занадто довго для деяких застосунків, наприклад, для віртуальної реальності. Системний монітор показує, що процесор завантажений на 80%, а пам'ять - на 60%. Це свідчить про те, що система не перевантажена, але близька до граничних значень.

На основі цього можна зробити висновок, що 3D-модель сцени є досить складною, але комп'ютерна система може забезпечити прийнятну продуктивність під час роботи з нею.