

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики

(інститут)

Факультет інформаційних технологій

(факультет)

Кафедра Програмного забезпечення комп'ютерних систем

(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня

бакалавра

(назва освітньо-кваліфікаційного рівня)

студента

Пеліна Владислава Олександровича

(ПІБ)

академічної групи

122-19-2

(шифр)

спеціальності

122 Комп'ютерні науки

(код і назва спеціальності)

освітньої програми

Комп'ютерні науки

(назва освітньої програми)

на тему:

*Створення 3D-моделі та її впровадження у Unreal Engine
за допомогою мови програмування C++*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтингов ою	інституцій ною	
кваліфікаційної роботи	<i>проф. Бердник М.Г.</i>			
розділів:				
спеціальний	<i>проф. Бердник М.Г.</i>			
економічний	<i>проф. Вагонова О.Г.</i>			
Рецензент	<i>доц. Шедловський І.А.</i>			
Нормоконтролер	<i>доц. Гуліна І.Г.</i>			

Дніпро
2023

Міністерство освіти і науки України
НТУ «Дніпровська політехніка»

ЗАТВЕРДЖЕНО:

завідувач кафедри
програмного забезпечення комп'ютерних систем

(повна назва)

М.О. Алексєєв

(підпис)

(прізвище, ініціали)

« » 2023 року

ЗАВДАННЯ

на кваліфікаційну роботу
бакалавра

(назва освітньо-кваліфікаційного рівня)

студента 122-19-2 Пелін В.О.
(група) (прізвище та ініціали)

тема кваліфікаційної роботи Створення 3D-моделі та її впровадження у
Unreal Engine за допомогою мови програмування C++

затверджена наказом ректора НТУ «ДП» від 16.05.2023 № 350-с

Розділ	Зміст виконання	Термін виконання
Спеціальний	На основі матеріалів проектно-технологічної практики та інших науково-технічних джерел провести аналіз стану рішення проблеми та постановку задачі. Обґрунтувати вибір та здійснити реалізацію методів вирішення проблеми	13.05.2023 р.
Економічний	Провести розрахунок трудомісткості розробки програмного забезпечення, витрат на створення ПЗ й тривалості його розробки	27.05.2023 р.

Завдання видав _____ проф. Бердник М.Г.
(підпис) (посада, прізвище, ініціали)

Завдання прийняв до виконання _____ Пелін В.О.
(підпис) (прізвище, ініціали)

Дата видачі завдання: 14.01.2023 р.

Термін подання кваліфікаційної роботи до ЕК: 12.06.2023 р.

РЕФЕРАТ

Пояснювальна записка випускної кваліфікаційної роботи містить: 85 ст., 53 рис., 4 табл., 3 дод., 21 джерел.

Об'єкт розробки: 3D-моделі до Unreal Engine.

Мета кваліфікаційної роботи: розробити програмний застосунок, котрий зможе, використовуючи створені 3D-моделі, створити на базі двигуна Unreal Engine гру – платформер, яка дозволить користувачеві весело провести час за нею та відволіктися від своїх проблем.

У вступі розглядається та аналізується поточний стан проблеми, з'ясовується головна мета роботи та сфера її застосування, аргументується актуальність теми.

У першому розділі аналізується предметна галузь, наводяться необхідні теоретичні відомості, визначається доцільність підстави для розробки, формується завдання та визначаються вимоги для впровадження застосунку.

Другий розділ обирає необхідні платформи для розробки, надає характеристику технологіям та інструментарію, котрий був використаний у ході процесу створення, описує програму, її процес створення, реалізацію необхідних алгоритмів, визначає виклик і завантаження програми.

У третьому розділі, котрий ще називають економічним, визначається доцільність розробки інформаційної системи, розрахунок вартості робіт зі створення програмного застосунку та час створення, що буде затрачений на роботу.

Практичне значення полягає у створенні застосунку, що надає можливість користувачеві отримати весь спектр яскравих моментів та відійти від усіх негативних емоцій, котрі гравець міг накопичувати у собі.

Актуальність інформаційної системи визначається великим попитом на таких майданчиках для продажу ігрових застосунків, як: Steam, Epic Games Store, Origin, Uplay та інші. Мільйони доларів, котрі кожен рік заробляють ці майданчики за надання місця ігровим розробникам говорять краще за все.

Список ключових слів: ТРИВИМІРНЕ МОДЕЛЮВАННЯ, ТЕКСТУРА, ІГРОВИЙ ДВИГУН, ВПРОВАДЖЕННЯ МОДЕЛІ.

ABSTRACT

The explanatory note of the final qualification work contains: 85 p., 53 pictures., 4 tables, 3 add., 21 sources.

Object of development: implementation of 3D model to Unreal Engine.

The goal of the qualification work: to develop a software application that will be able to use the created 3D models, to create a platformer game based on the Unreal Engine engine, which will allow the user to have fun with it and distract himself from his problems.

In the first section, the subject area is analyzed, the necessary theoretical information is given, the expediency of the basis for development is determined, the task is formed, and the requirements for the realization of the application are determined.

The second section selects the necessary platforms for development, characterizes the technologies and tools used during the creation, describes the program, its creation process, the implementation of the necessary algorithms, defines the call and download of the program.

The third section, which is also called economic, determines the expediency of developing an information system, calculating the cost of work on creating a software application, and the creation time that will be spent on the work.

The practical sense is to create an application that will allow the user to get the full range of bright moments and move away from all the negative emotions that the player could accumulate in himself.

The relevance of the information system is determined by the high demand on such platforms for the sale of game applications, such as: Steam, Epic Games Store,

Origin, Uplay and others. This is best illustrated by the millions of dollars these sites earn each year for hosting game developers.

List of keywords: 3D MODELING, TEXTURE, GAME ENGINE, MODEL IMPLEMENTATION.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПК/PC – персональний комп'ютер;

ОС – операційна система;

VRAM – відеопам'ять;

GB – гігабайт. Одиниця вимірювання кількості інформації;

GHz – частотна одиниця, яка вимірює кількість циклів в секунду;

ПЗ – програмне забезпечення;

Платфóрмер (англ. Platformer) – жанр відеоігор, ігровий процес в якому складається зі стрибків персонажа по різноманітним платформам та через перешкоди, збирання предметів;

RTS – Стратегія в реальному часі;

RPG – Рольова відеогра;

Gamedev – це розробник, котрий створює ігрові застосунки на різноманітних платформах: PC, Play Station, Nintendo, Xbox та інші. Для цього він займається візуалізацією та концепцією гри, створенням програмного коду, вибирає засоби та інструменти для вирішення поставлених завдань;

КР – кваліфікаційна робота.

ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT.....	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ	
1.1. Загальні відомості з предметної галузі.....	10
1.2. Призначення розробки та галузь застосування.....	22
1.3. Підстава для розробки	25
1.4. Постановка завдання.....	25
1.5. Вимоги до програми або програмного виробу	26
1.5.1. Вимоги до функціональних характеристик.....	26
1.5.2. Вимоги до інформаційної безпеки.....	26
1.5.3. Вимоги до складу та параметрів технічних засобів.....	27
1.5.4. Вимоги до інформаційної та програмної сумісності	27
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ	
2.1. Функціональне призначення системи.....	28
2.2. Опис застосованих математичних методів.....	28
2.3. Опис використаних технологій та мов програмування.....	29
2.3.1. Порівняння редакторів 3d моделювання.....	29
2.3.2. Порівняння редакторів для створення текстур об'єктів.....	41
2.3.3. Порівняння ігрових двигунів для розробки.....	46
2.3.4. Створення візуальної частини застосунку.....	54
2.4. Опис структури системи та алгоритмів її функціонування	64
2.5. Обґрунтування та організація вхідних та вихідних даних програми.....	66
2.6. Опис розробленої системи	66
2.6.1. Використані технічні засоби.....	66
2.6.2. Використані програмні засоби.....	67
2.6.3. Виклик та завантаження програми.....	67

2.6.4. Опис інтерфейсу користувача.....	68
РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ	
3.1. Розрахунок трудомісткості та вартості розробки програмного продукту.....	70
3.2. Рахунок витрат на створення програми.....	74
ВИСНОВОК.....	76
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	77
Додаток А. Код програми.....	79
Додаток Б. Відгук керівника економічного розділу.....	84
Додаток В. Перелік файлів на диску.....	85

ВСТУП

У наш час комп'ютерні технології є однією з галузей, що швидко розвиваються. За останні десятиліття прогрес у цій галузі досягнув значних розмірів. Якщо раніше комп'ютери і телефони у тому вигляді, який ми знаємо зараз, були рідкістю, фільми знімалися майже без віртуальних спецефектів, а ігри взагалі не представлялися в цифровому вигляді, то зараз неможливо уявити собі сучасні галузі виробництва, науки, культури, спорту та економіки, де не застосовувалися б комп'ютери.

Однією із значущих областей комп'ютерних технологій є комп'ютерна графіка, зокрема її розділ – тривимірна графіка, або 3D-моделювання. Головне завдання 3D-моделювання – це показати візуальне об'ємне уявлення будь-якого об'єкта: вже існуючого або тільки задуманого. Без моделювання важко уявити наше життя: архітектурна візуалізація перейшла у спеціальний напрямок в житті архітекторів; інженери за допомогою систем автоматизованого проектування швидше та ефективніше справляються зі поставленим їм завданням; в медицині також виділяються окремі напрями: точкова томографія, конструювання та створення протезів тощо. Важко не погодитися з тим, що процес моделювання, який захоплює все більше і більше галузей промисловості та науки, йде лише на користь розвитку суспільства.

На даний момент 3D-моделі використовують майже всі сфери життя: телебачення, кінотеатри, рекламні відео, продуктові фірми, медичні компанії тощо.

Однак найбільш зацікавленими в 3D-моделюванні є розробники віртуальних ігор. Адже їх робота – це втілення неіснуючого в наш реальний світ, нехай і лише у вигляді віртуальних 3D-моделей. Вони буквально існують завдяки тому, що можуть використовувати ці моделі, так як без них відтворити саме ту реальність, яка була би найкращою реалізацією їх думок буде дуже складно.

Тому моделювання тривимірних об'єктів та використання їх на ігровому двигуні(в чому й полягає створення більшості ігор) є достатньо актуальною темою для випускної кваліфікаційної роботи.

Метою даної роботи є створення 3D-моделі та її впровадження у Unreal Engine.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1. Загальні відомості з предметної галузі

Тривимірна графіка - розділ комп'ютерної графіки, сукупність прийомів та інструментів (як програмних, так і апаратних), покликаних забезпечити просторово-часову безперервність одержуваних образів. Найбільше застосовується для створення зображень в архітектурній візуалізації, кінематографі, телебаченні, комп'ютерних іграх, друкованої продукції, а також у науці.

Щоб отримати тривимірне зображення, слід пройти два етапи:

- етап моделювання;
- етап рендерингу.

Моделювання - це процес створення математичної моделі сцени і об'єктів, що знаходяться в ній. Приклад наведено на рис. 1.1.



Рис. 1.1. 3D-модель автомобіля

Рендеринг (що з англійської перекладається як «візуалізація») у комп'ютерній графіці – процедура отримання зображення моделі за допомогою комп'ютерної програми. У даному контексті, модель – це опис будь-яких об'єктів чи явищ на якійсь з мов програмування або за допомогою структури даних. Говорячи про опис, маються на увазі геометричні дані, положення точки спостерігача, інформація про освітлення та інше.

Нижче на рис. 1.2 показана модель автомобіля, отримана в процесі рендерингу.



Рис. 1.2. Візуалізована модель автомобіля

Моделювання ж поділяється на декілька видів, кожен з яких буде окремо розібраний.

Полігональне моделювання виникло тоді, коли для визначення розташування точки вручну вводилися її координати по трьох осях XYZ. Якщо всі три точки координат задати як вершини та з'єднати їх ребрами, то вийде трикутник, який у 3D-моделюванні називають полігоном. Будь-який 3D-об'єкт має сітку, що складається з вершин, ребер, граней. Сам полігон складається з граней, але в системах, що підтримують багатосторонні грани, полігони та грані будуть рівнозначними. Приклад полігонального моделювання наведено на рис. 1.3.

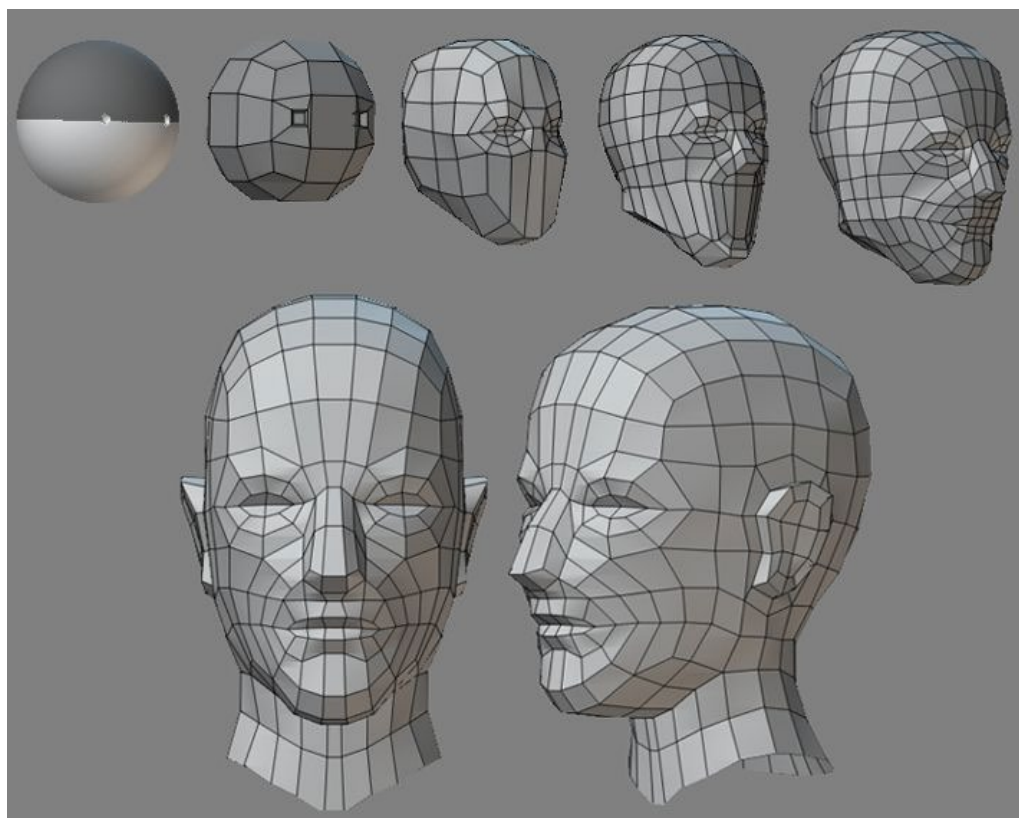


Рис. 1.3. Моделювання голови людини методом полігонів

Моделювання за допомогою полігонів дає можливість проводити різні маніпуляції із сіткою 3D-об'єкта. Цей спосіб є першим і одночасно основним видом моделювання — можна створити об'єкт будь-якої проблеми шляхом з'єднання груп полігонів.

Полігональне моделювання поділяється на три типи:

- низькополігональне моделювання, чи Low-Poly, призначене до створення об'єктів з невеликим числом полігонів. Зазвичай це дозволяє економити ресурси, якщо немає потреби у високій деталізації;
- середньополігональне моделювання, чи Mid-Poly, спрямовано необхідний результат при рендерингу; полігональна сітка не оптимізується або зміни мінімальні;
- високополігональне моделювання, або High-Poly, дозволяє здавати об'єкт з великою кількістю полігонів, що значно підвищує комп'ютерну продуктивність.

Сплайнове моделювання – це створення 3D-об'єктів за допомогою сплайнів – кривих ліній. Сплайни – це лінії різної форми: прямокутники, кола, дуги, текст тощо. Об'єкти у своїй можуть бути грубої, і плавної форми, що дозволяє створювати органічні моделі, наприклад рослини чи живі істоти. Головна перевага сплайнового моделювання – гнучкість зміни форми сплайну. (рис. 1.4)

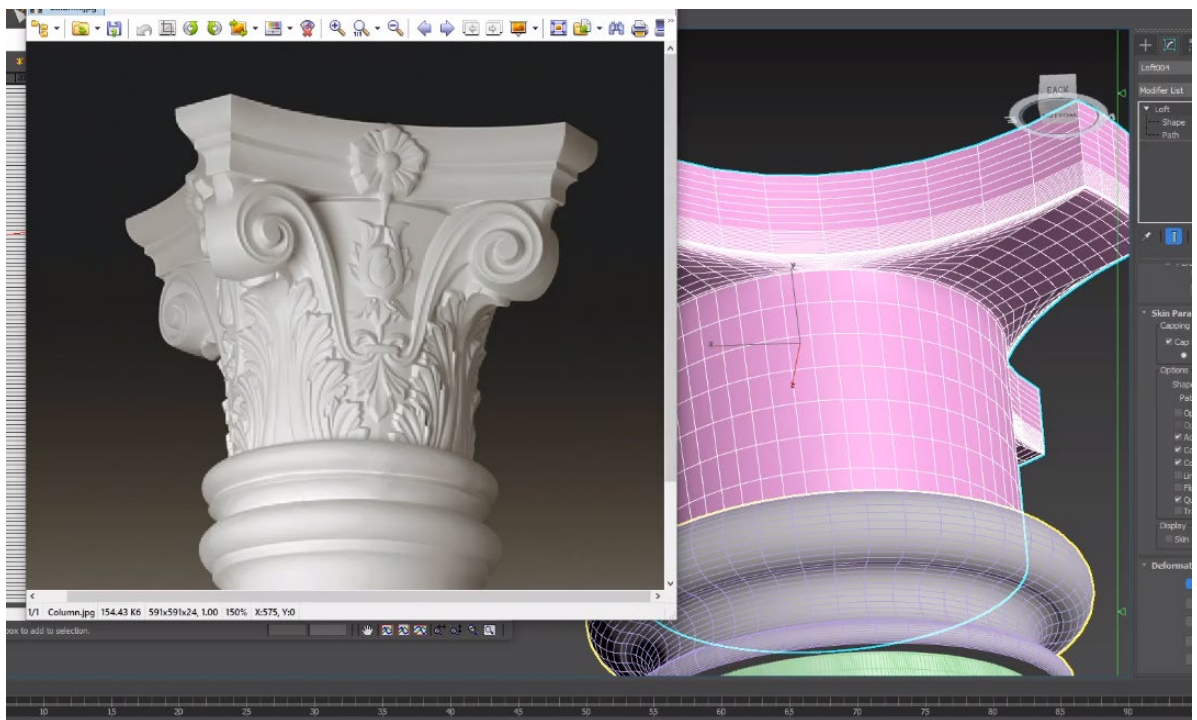


Рис. 1.4. Створення колони шляхом сплайнового моделювання

Цей вид моделювання нерідко порівнюють з полігональним, як векторну графіку з растровою. У векторній графіці зі збільшенням об'єкта його якість не змінюється, на відміну від растрового, де стають видні пікселі. Так само і при збільшенні об'єкта, створеного сплайнами, його якість залишається незмінною, а при полігональному моделюванні буде видно полігони.

Звідси впливають такі переваги сплайнового моделювання:

- можливість будь-якої миті змінити форму конкретного сплайна, тобто готовий об'єкт відредагувати нескладно;
- збереження якості при масштабуванні, і саме з цієї причини перевага сплайнового моделювання надається все частіше і частіше, особливо при розробці тривимірних ігор.

Тобто, якщо потрібно створити об'єкт, який зберігатиме свою якість при наближенні, потрібно вибрати саме цей спосіб.

Non-Uniform Rational B-Spline (укр. «неоднорідний раціональний В-сплайн») – це технологія створення 3D-об'єктів за допомогою спеціальних кривих, які називаються В-сплайнами. Деякі фахівці виділяють даний вид моделювання в окремий, інші – як підвид сплайнового моделювання.

Принцип цього моделювання полягає в наступному: за допомогою В-сплайнів, розташованих по вертикалі та горизонталі, будується відповідна форма об'єкта, а потім все це з'єднується за допомогою полігонів.

Є два різновиди цього моделювання (рис. 1.5):

- за допомогою Р-кривих (Point), форму яких можна змінювати за допомогою вершин, які розташовані безпосередньо на самій лінії;
- за допомогою CV-кривих (Control Vertex), форму яких можна змінювати за допомогою вершин, які розташовані за межами смуги.

Слід зазначити, що принцип побудови таких кривих полягає у використанні певних математичних формул.

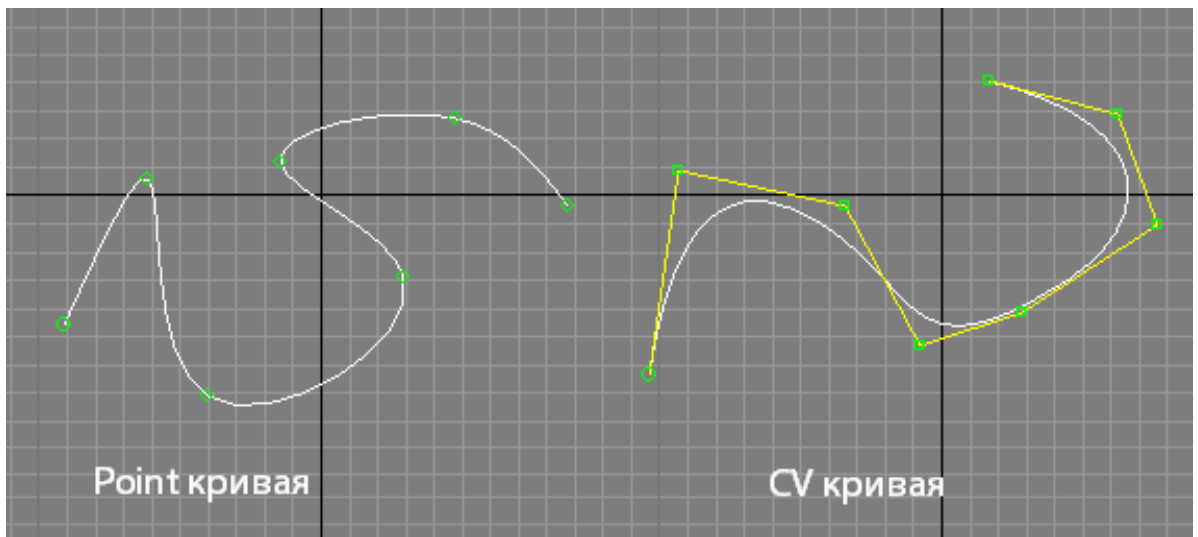


Рис. 1.5. Різновиди моделювання В-сплайнами

Головна перевага цієї технології полягає в тому, що вона призначена для створення плавних органічних форм та моделей, так як заснована на складному математичному апараті. Усього існує близько 1500 рівнянь для опису всіх геометричних елементів, від простих кривих до складних поверхонь.

Через особливості будови NURBS поверхні завжди гладкі (у них немає гострих країв, властивих полігонам), тому вони широко використовуються в органічному моделюванні (подібному до створення рослинних форм), для створення моделей тварин, людей, машин і так далі. NURBS поверхні не складаються з сітки прямокутників, розбиття поверхонь на багатокутники відбувається лише на етапі рендерингу та передбачає використання оптимального алгоритму для збереження гладкості. Тому за будь-якого наближення дотримується гладкість поверхні.

3D-скульптинг, він же «цифрова скульптура», показаний на рис. 1.6 є імітацією процесу «ліплення» 3D-моделі, тобто деформування її полігональної сітки спеціальними інструментами – кистями. Можна провести аналогію з ліпленням фігур руками із пластиліну чи глини.

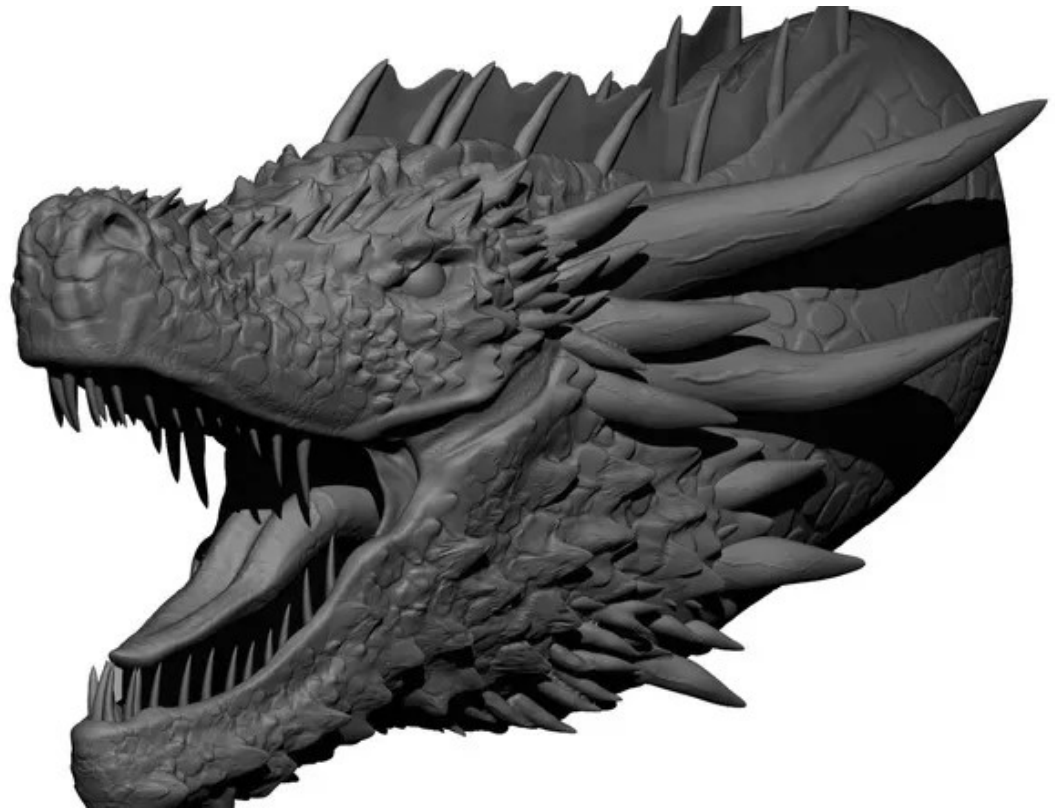


Рис. 1.6. 3D-скульптура голови дракона

3D-скульптинг вплинув на такі ключові сфери 3D-моделювання:

- саме моделювання. Скульптинг спростив безліч моментів, а також привніс багато зручності у створення складних за структурою форм;
- текстурування. Тепер можна малювати та редагувати текстурні карти прямо на поверхні 3D-об'єктів;
- низькополігональне моделювання. Тепер низькополігональну сітку можна будувати, створювати безпосередньо на поверхнях високополігональних об'єктів – ретопологія. Або програми скульптингу під час експорту до інших пакетів автоматично переводять hi-poly нижчу роздільну здатність – автоматична ретопологія.

Промисловість – сфера, у якій 3D-моделювання предметів, вузлів та його функціональності найбільш актуально. Помилки проектування, які неминуче виникають при втіленні механізму креслення або схеми, наочно відображаються на моніторі. Вносячи поправки в цифрову модель можна побачити результати цих корективів, домагаючись максимальної працездатності та естетичності.

Для бізнесу ці технології – величезна економія. Цю проекцію можна використовувати для виготовлення точної копії зразка, щоб оцінити її ще повніше або продемонструвати потенційним інвесторам. Готельні види виробництв повністю перейшли на масовий випуск продукції з електронної моделі: і це не лише сувеніри, іграшки чи декоративні аксесуари, а й сегменти механізмів та високотехнологічної апаратури.

Використання 3D-моделювання для вирішення завдань промислового підприємства дає деякі переваги:

- зниження витрат на виготовлення будь-яких виробів;
- дуже швидка розробка нових виробів;
- простота використання готових виробів.

Для створення 3D-моделей промислового призначення застосовують системи автоматизованого проектування (САПР) або англійською мовою Computer-Aided Design (CAD), що показано на рис. 1.7. Вони призначені для створення точних копій реальних об'єктів.

У даному виді моделювання враховуються не тільки найменші проміжки, але і властивості матеріалу об'єкта, що моделюється. У зв'язку з цим цей вид моделювання знайшов широке застосування в інженерній справі. Особливість цього моделювання в тому, що для створення використовують не полігони, а цілісні форми.

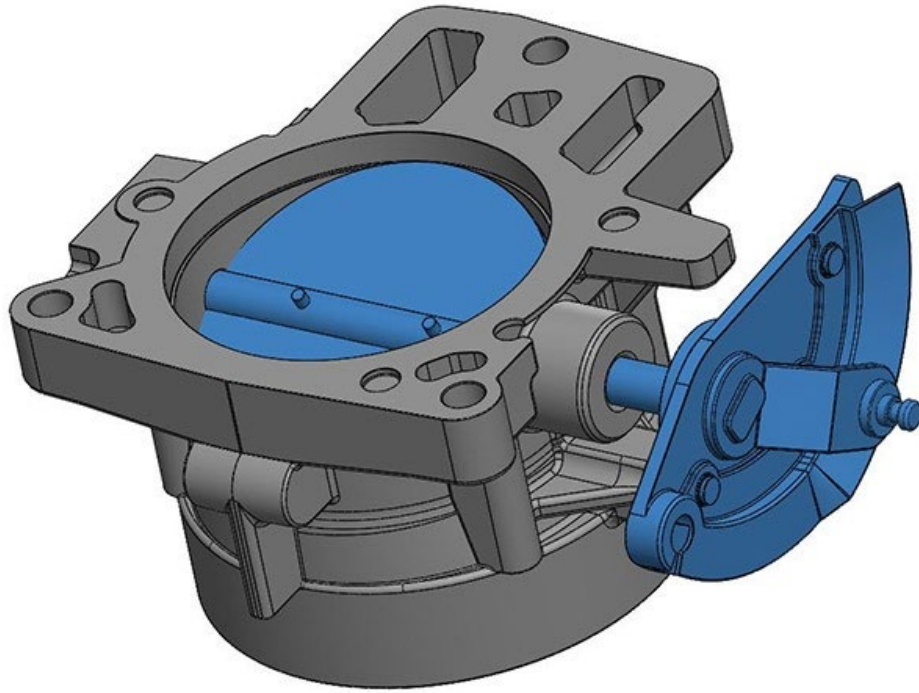


Рис. 1.7. 3D-модель промислового призначення

Параметричне моделювання на рис. 1.8. здійснюється шляхом введення необхідних параметрів елементів моделі, а також співвідношень між ними. Іншими словами, створюється математична модель з потрібними параметрами, змінюючи які можна створити різні комбінації моделей і тим самим уникнути помилок, замість необхідних коригувань.

Є досить старим і найпростішим способом проектування промислових деталей та механізмів.

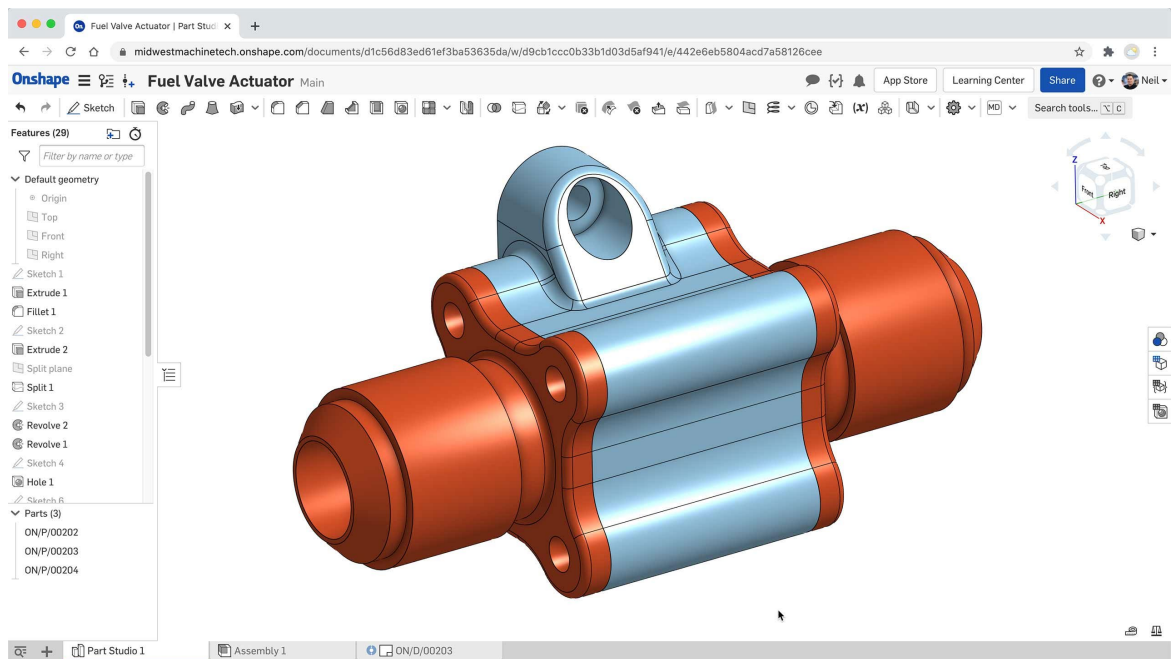


Рис. 1.8. Параметричне моделювання у системі OnShape

Параметричне тривимірне, або його ще називають двовимірним моделюванням, суттєво відрізняється від звичайного креслення або 3D-моделювання. У випадку параметричного моделювання створюється математична модель з параметрами, зміна яких тягне за собою зміну абсолютно всієї конфігурації деталі, переміщення деталей у складання та інші схожі трансформації.

Ідея створення параметричного моделювання з'явилася досить давно, але, на жаль, її втілення було неможливо через недостатню продуктивність комп'ютерів. 1989 рік став датою народження параметричного моделювання, оскільки саме цього року були випущені перші САПРи з функціями параметризації.

Формування та використання залежностей та маніпуляції з ними по суті є процесом проектування. Тому, як було зазначено, параметричне моделювання є найпростішим, зручним способом проектувати об'єкти, оскільки саме ця технологія надає фахівцю повний доступ до контролю залежностей. Параметризація як спосіб проектування об'єктів для професіоналів є настільки ж легким методом як і редагування тексту в MS Word.

Якщо при полігональному моделюванні куб розрізати навпіл, там усередині буде порожнеча. При твердотільному моделюванні, як на малюнку, якщо розрізати куб, то не буде порожнечі, ніби розрізали реальний твердий предмет.

При побудові моделі працюють відразу з усією оболонкою, а чи не з окремими поверхнями. Спочатку створюється проста форма оболонки, наприклад сфери, та був до неї застосовують різні операції: різання, об'єднання коїться з іншими тілами, булеві операції, як на рис. 1.9.

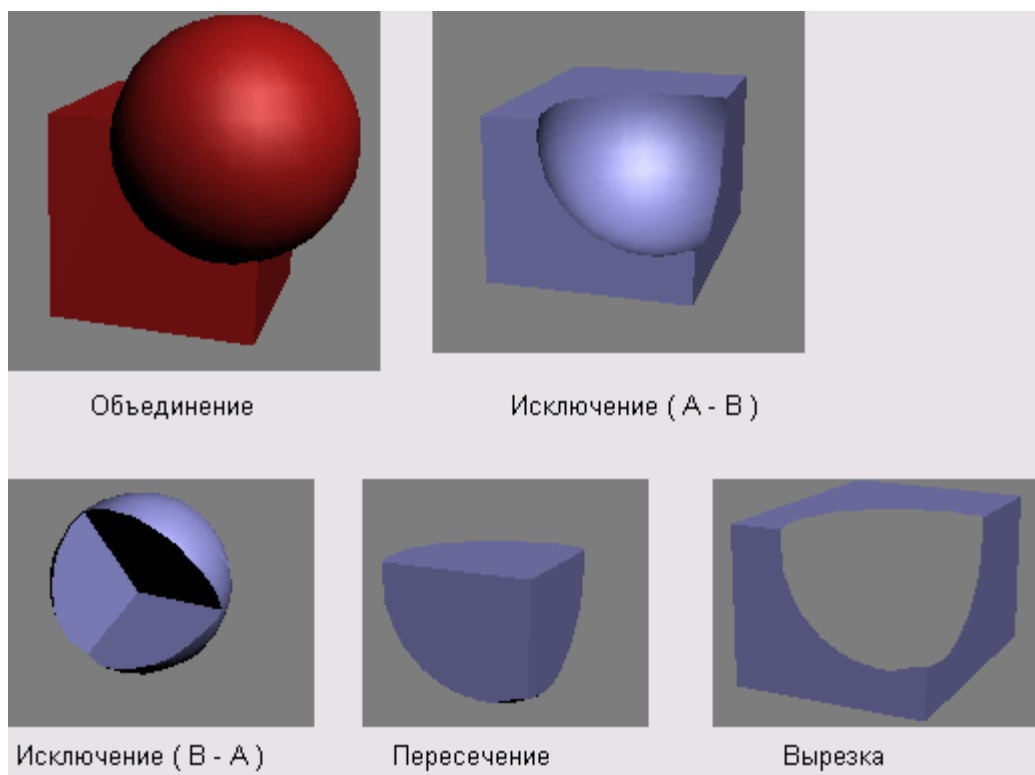


Рис. 1.9. Приклад булевих операцій при твердотільному моделюванні

Твердотільне моделювання ідеально підходить для створення жорстких 3D-моделей нескладної форми: шестерня, двигун, і так далі, але не застосовується до створення м'яких: м'ятого одягу, тварин тощо.

Переваги твердотільного моделювання:

- найкраща візуалізація та сприйняття створеної моделі. Тривимірна модель із застосуванням сучасних технологій виглядає більш ніж реалістично;

- автоматичне формування креслень – одне з найголовніших переваг цієї технології. Побудова моделі та формування креслень за нею з використанням твердотільного моделювання – справа кількох секунд;
- швидкість та легкість у процесі внесення змін та коригувань у модель – не потрібно заново формувати креслення, достатньо змінити потрібні пункти та оновити програму. Також можна використовувати шаблони, що значно скоротить час виконання роботи;
- поєднання з різними додатковими програмами – інтеграція дозволяє скоротити час, використовуючи відразу отримані результати на наступних стадіях роботи;
- швидкість під час проектування – твердотільне моделювання скорочує термін виконання проектування об'єкта. Швидкість моделювання позитивно впливає на швидкість повернення вкладених інвестицій.

Створення твердотільних моделей як ніколи сьогодні є актуальним. Важливо не тільки швидко створювати об'єкт, а й так само швидко редагувати його. Твердотільне моделювання має дані якості, тому воно вважається найдосконалішою технологією. Методи уявлень, а саме граничний та конструктивний, забезпечують максимально реалістичні моделі.

Маючи такі суттєві переваги, твердотільне моделювання визнано найшвидшим, якісним та ефективним методом при проектуванні складних об'єктів.

1.2. Призначення розробки та галузь застосування

На сьогоднішній день 3D-моделювання має доволі велике значення у житті людини. Його використовують для таких робіт як:

гірничя промисловість – створення тривимірної моделі шахти/кар'єра;
(рис. 1.10)

мультиплікація – створення тривимірних персонажів для мультфільмів;
(рис. 1.11)

медична сфера – для пластичної хірургії, моделювання протезів та імплантатів; (рис. 1.12)

промисловість – Кожну складну деталь збирають по готовій та продуманій 3D-моделі; (рис. 1.13)

відеогеймінг – створення 3d моделей для реалізації певних комерційних та безкоштовних проєктів. (рис. 1.14)

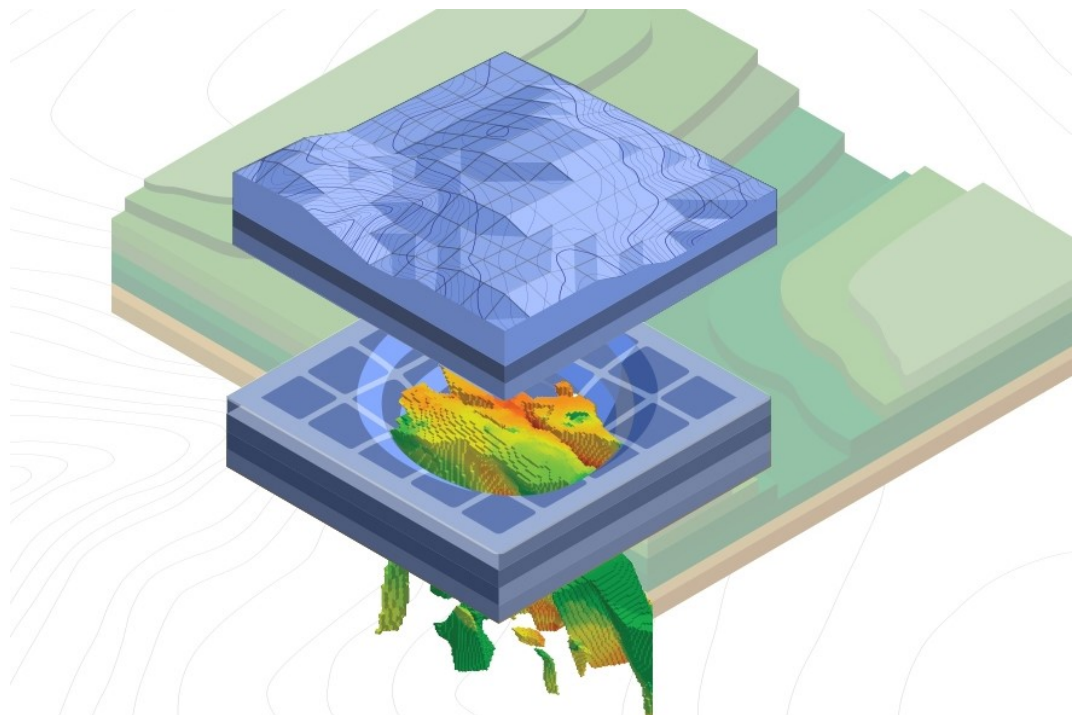


Рис.1.10. Тривимірна модель в гірничій промисловості

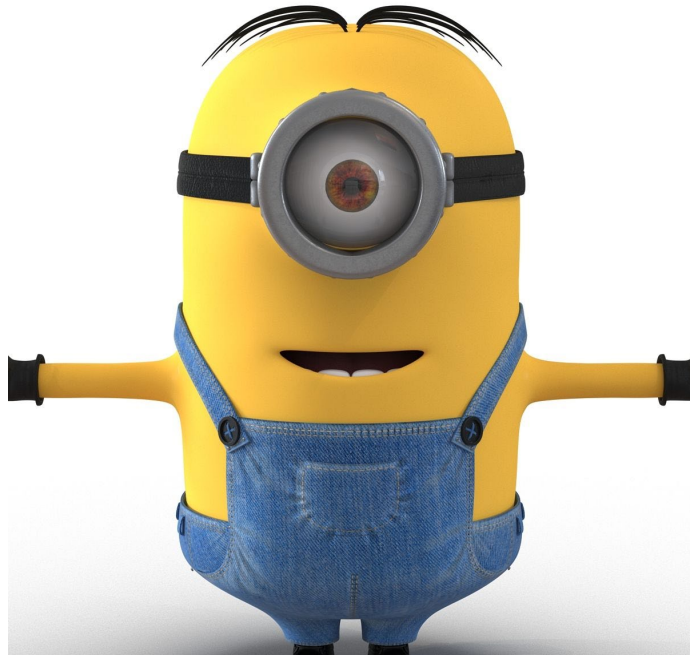


Рис.1.11. Персонаж «мінйон» з мультиплікації «Гадкий я»



Рис.1.12. 3d модель медичного протезу ноги людини

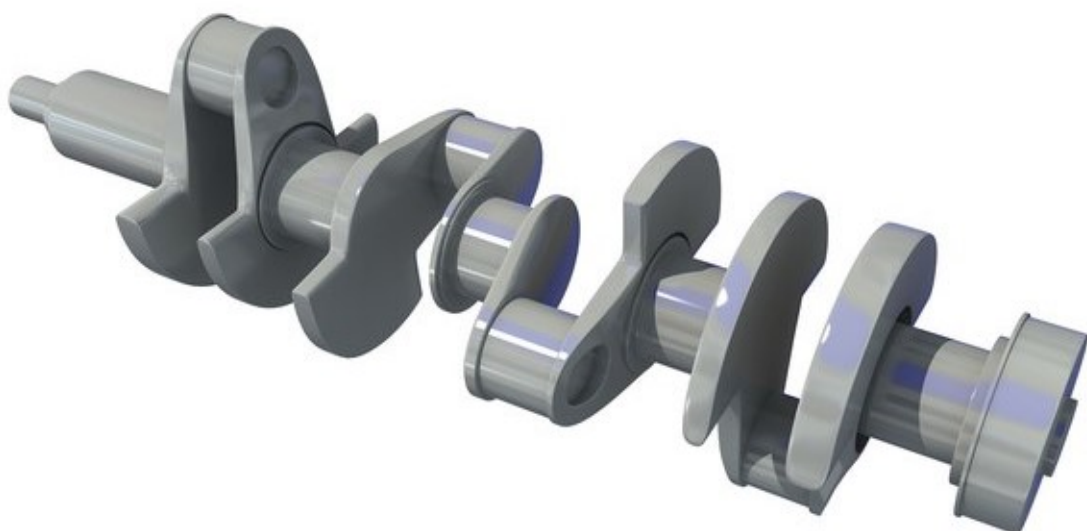


Рис.1.13. 3d модель колінчастого валу



Рис.1.14. 3d модель автомата АК-74 для гри FPS

1.3. Підстава для розробки

Підстави для розробки та виконання кваліфікаційної роботи :

- Освітня програма 122 Комп'ютерні науки;
- Навчальний план та графік навчального процесу;
- Наказ ректора Національного технічного університету «Дніпровська політехніка» №350-с від 16.05.2023 р;
- Завдання на кваліфікаційну роботу на тему «Створення 3D-моделі та її впровадження у Unreal Engine за допомогою мови програмування C++».

1.4. Постановка завдання

Розробити застосунок, з використанням тривимірної графіки на базі певного ігрового двигуна.

Цей застосунок повинен буде підходити до вимог замовника та приватного користувача.

До вимог замовника належать такі :

- актуальність теми – застосунок повинен бути актуальним на сьогоднішній день;
- вартість/зусилля – готовий виріб повинен мати розумну ціну, яку підприємство виділить на його створення;
- окупність – кінцевий виріб повинен окупити себе та принести додаткову вигоду, інакше він не має сенсу в створенні;
- безпечність – застосунок має бути захищений від втрати даних чи викрадання його певними заінтересованими особами;

До вимог клієнта належать такі критерії:

- розумна ціна – кінцевий виріб повинен мати адекватну ціну покупки, інакше його не будуть купувати;
- зручний інтерфейс – застосунок повинен бути зручним у використанні ;

- цікавість – виріб повинен бути цікавим, виконувати головну функцію (розважати клієнта) на 100%;
- функціональність – застосунок повинен виконувати всі функції, що були оголошені на соціальних сторінках, у рекламі тощо;
- безпечність – готовий виріб має бути захищеним від типових помилок клієнтів.

1.5. Вимоги до програми або програмного виробу

1.5.1. Вимоги до функціональних характеристик

Програма, виконана у рамках даної кваліфікаційної роботи повинна вміти:

- зчитувати натискання користувача на клавіатуру;
- обробляти ці натискання та виконувати певну функцію (переміщення на ігровому полі тощо);
- дозволяти користувачеві отримувати винагороду за гру (монети-бонуси які «зникають» при зіткненні з ними);
- не давати гравцю зламати самому собі гру (виліт за межі ігрового поля тощо).

1.5.2. Вимоги до інформаційної безпеки

Кінцевий виріб повинен мати:

- стійке(безперебійне) функціонування – повинна буди самодостатність для виконання функцій, закладених в неї;
- контроль вхідної та вихідної інформації – захист від неправильного занесення/обробки інформації;
- час відновлення після відмови – виріб мусить відновити свою працездатність в найкоротші строки;

- захист від несанкціонованого доступу – ніхто, окрім уповноважених осіб, не повинен мати доступ до приватних частин програми.

1.5.3. Вимоги до складу та параметрів технічних засобів

Програма розроблена для використання на ПК на базі ОС Windows (64-розрядна система).

Для її ефективної працездатності встановленні мінімальні та рекомендовані параметри системи.

Мінімальні параметри:

- ОС: Windows 10 (64-bit);
- процесор: Intel Core i3 3.0 GHz;
- оперативна пам'ять: 2 GB;
- відеокарта: GeForce GTX 660 2GB VRAM.

Рекомендовані параметри:

- ОС: Windows 10 (64-bit);
- процесор: Intel Core i5-4590 3.70 GHz;
- оперативна пам'ять: 4 GB;
- відеокарта: GeForce GTX 1050 4GB VRAM;
- місце на жорсткому диску: 3 GB.

1.5.4. Вимоги до інформаційної та програмної сумісності

Програма повинна мати:

- зрозумілий, лаконічний та працездатний вихідний код;
- актуальне програмне середовище для створення;
- адаптивність для переходу на іншу операційну систему;
- мова програмування має бути актуальною для світового ринку та ефективною для праці з нею.

РОЗДІЛ 2.

ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Функціональне призначення системи

Дана інформаційна система являє собою віртуальний застосунок, розроблений на ігровому двигуні, з метою отримання яскравих та добрих відчутті від процесу його використання.

Для виконання цієї мети дуже сильно допоможуть:

- тривимірна графіка, яка втягне користувача у події, що відбуваються на екрані на більш глибокий рівень усвідомлення;
- текстуровані моделі, які поділяться з гравцем всіма яскравими кольорами, що мають у собі;
- різнорівневий ландшафт рівня, котрий наділе користувача можливістю відчуті усі перешкоди, які долатиме персонаж та накопичити симпатію до нього та до гри узагалі;
- цікавий геймплейний момент у вигляді збирання монет, розташованих по всьому ігровому рівні, який надає змогу для отримання користувачем певного азарту від гри, а отже і радості.

2.2. Опис застосованих математичних методів

В даній кваліфікаційній роботі не було використано математичних методів. Ігровий двигун надає більш ніж достатньо можливостей для автоматизування процесу навіть без додавання додаткових методів. Звідси й виходить, що для даної роботи у використанні математичних методів не має сенсу.

2.3. Опис використаних технологій та мов програмування

2.3.1. Порівняння редакторів 3d моделювання.

Оскільки кінцевий виріб повинен містити у собі тривимірні моделі, то потрібно обрати, з яким саме редактором 3d моделювання буде зручніше працювати у даній кваліфікаційній роботі.

Blender – професійне, вільне і відкрите програмне забезпечення для створення тривимірної комп'ютерної графіки, що включає в себе засоби моделювання, анімації, рендерингу, постобробки та монтажу відео зі звуком, компонування за допомогою «вузлів» (Node Compositing), а також для створення інтерактивних ігор. В даний час вона користується найбільшою популярністю серед безкоштовних 3D-редакторів у зв'язку з його швидким та стабільним розвитком, якому сприяє професійна команда розробників.

Однією з переваг для широкого користувача, у тому числі працюючого на аматорському рівні, є те, що він поширюється вільно. Але це не означає, що програма підходить новачкам і є простою для професіоналів. Blender 3D використовували в деяких фільмах, таких як «Людина павук 2», «Хардкор», «Світ дикого заходу», а також програма повністю використовувалася при створенні деяких мультфільмів і комп'ютерних ігор, які випускали самі розробники софту для розкриття переваг та недоліків Blender 3D.

На рис. 2.1 показано робоче середовище програми та вже змодельований об'єкт.

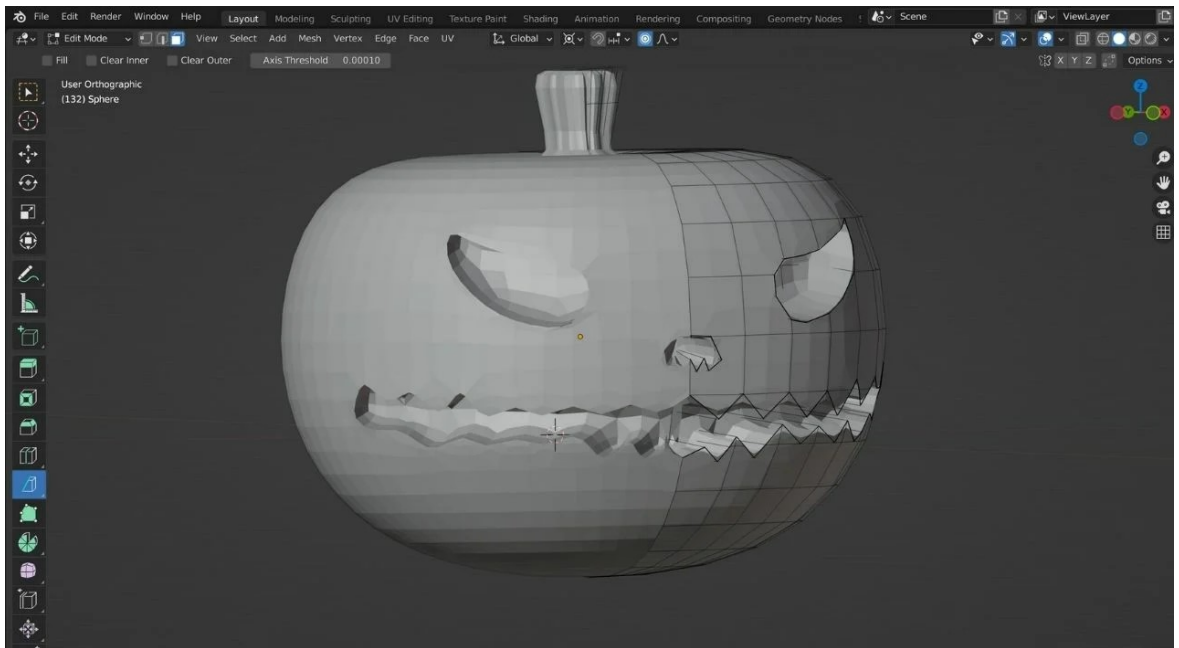


Рис. 2.1. Робоче середовище програми Blender 3D

Інструмент надає своїм користувачам такі можливості:

- рендеринг;
- моделювання;
- створення анімації;
- монтаж та обробка відео.

Це середовище активно використовують професійні розробники для створення інтерактивних ігор. Зображення при цьому виходять максимально реалістичними та ефектними.

Професійна команда розробників даного програмного забезпечення постійно працює над його вдосконаленням, що забезпечує популярність цього пакета для користувачів по всьому світу.

Blender 3D базується на відкритому коді. А це означає, що теоретично будь-хто може взяти участь у його вдосконаленні та розвитку, якщо має додатково навички програмування. У розробці цього програмного середовища беруть участь коди, написані мовами програмування Python, C, C++.

Maya – програма для 3D-анімації, моделювання та візуалізації, яка надає потужний інтегрований інструментарій, котрий можна використовувати для створення анімації, графіків руху, віртуальної реальності та персонажів. Приклад наведено на рис. 2.2.

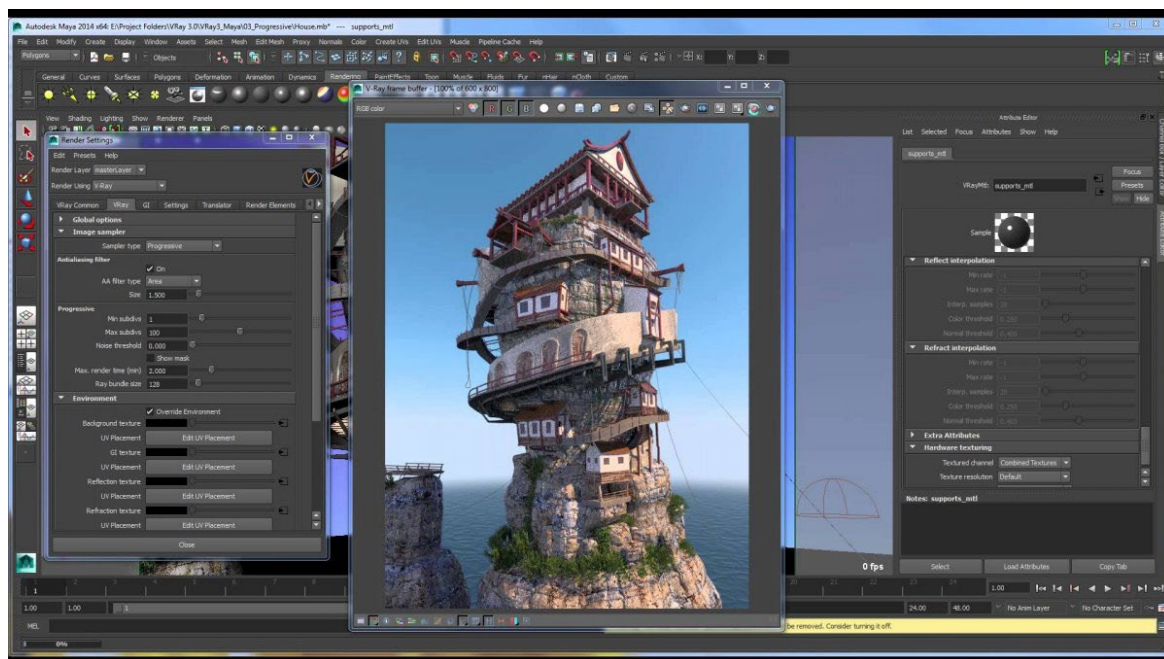


Рис. 2.2. Робоче середовище Autodesk Maya

Насамперед – це потужна зброя в руках 3D-аніматорів, працівників кіноіндустрії та телебачення. Maya цінується за величезний набір інструментів для анімації, текстурування, а також створення різноманітних спецефектів. Це серйозний редактор тривимірної графіки, який широко застосовується у професійних колах. У ній грамотно реалізована функція візуалізації готових моделей: програма оснащена чотирма вбудованими візуалізаторами, до того ж є можливість встановлення різних плагінів, зокрема V-Ray.

За способом роботи Autodesk Maya наближена до другого продукту компанії – 3Ds Max. Але порівняно з ним у Майї функції анімації реалізовані ґрунтовніше. Також програма цікава своєю можливістю роботи з динамікою твердих і м'яких тіл, простотою накладання текстур і зручним.

Якщо говорити про побудову 3D–моделей для 3D–друку, варто зазначити, що Autodesk Maya на це зовсім не орієнтована. Однак різноманітність її можливостей дозволяє використовувати програму навіть з цією метою.

Autodesk Maya оснащена різноплановими інструментами для якісної та зручної роботи з тривимірною графікою. У ній можна знайти все, що завгодно вимогливому модельєру, а також підібрати власний алгоритм дій. Ось короткий перелік того, що пропонує цей додаток:

- робота з кривими, включаючи NURBS;
- полігональне моделювання; можливість ретопології;
- кілька способів накладання матеріалів, текстур, а також зручний UV-мапінг (тобто розгортку);
- у версії Autodesk Maya 2016 є інструменти для скульптингу, подібні до пензлів Zbrush, причому скульптинг тут застосовний також для моделювання простого ландшафту;
- ряд інструментів для створення високоякісної анімації, є можливість проанімувати навіть окреме пасмо волосся;
- динаміка жорстких та м'яких тіл;
- унікальні інструменти для створення реалістичного волосся та вовни;
- симуляція води;
- створення спецефектів (дим, хмари, атмосферні ефекти);
- візуалізація як за допомогою вбудованих інструментів, так і додаткових плагінів, наприклад, плагін V-ray дозволяє досягти фотореалістичної візуалізації.

Щодо навчання роботі з програмою, то це справа не з легких. Що, втім, не заважає масі ентузіастів вивчати Autodesk Maya самостійно і досягати значних успіхів. Всесвітня мережа пропонує достатньо безкоштовних навчальних матеріалів у текстовому та відео-форматі.

Autodesk 3Ds Max (раніше 3D Studio MAX) – професійне програмне забезпечення для 3D-моделювання (рис. 2.3), анімації та візуалізації під час створення ігор та проектування. В даний час розробляється та видається компанією Autodesk.

В даний час програма 3Ds Max є найпопулярнішою серед розробників як професіоналів, так і любителів. Більше того, вона надає студентам можливість безкоштовно користуватися програмним забезпеченням протягом трьох років.

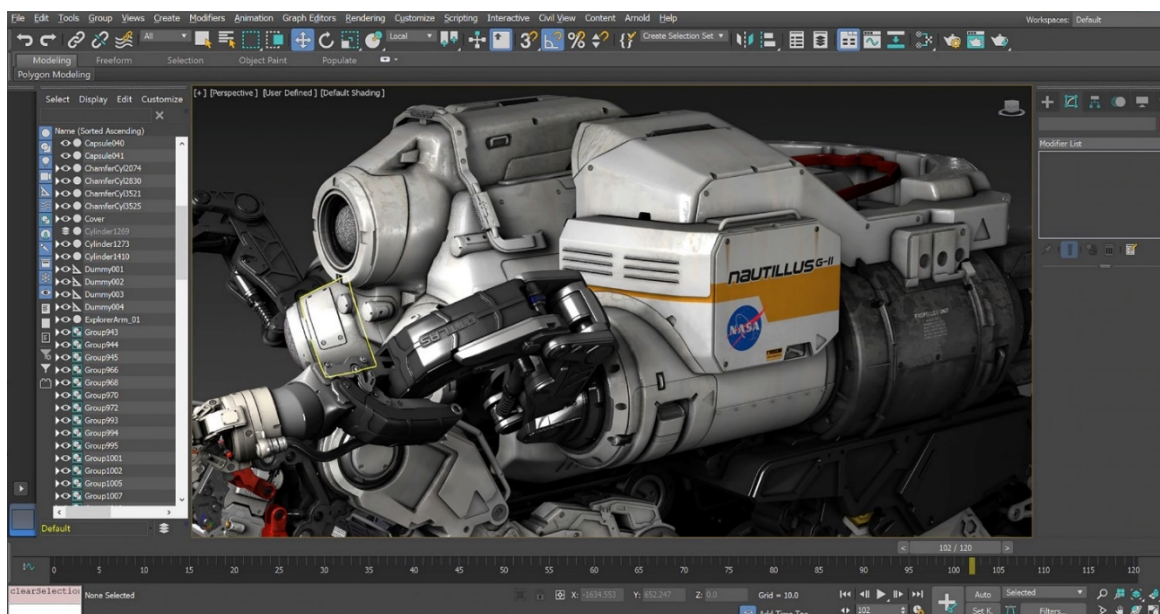


Рис.2.3. Робоче середовище Autodesk

У 3Ds Max наявна досить велика бібліотека тривимірних примітивів як стандартних, так і розширених. Побудова таких геометричних форм, як куб, циліндр, піраміда, призма, капсула не забирає багато часу – потрібно лише вибрати примітив і ввести його параметри, наприклад, висоту і радіус. Є інструменти для роботи зі сплайнами, які також легкі у вивченні. Нескладно освоєння полігонального моделювання. Можливість редагування сітчастих поверхонь на рівнях вершини або ребер полегшує роботу зі складними поверхнями і дозволяє досягти максимальної наочності в їх поданні.

У програму вбудований простий та зручний редактор матеріалів. Створення скляних чи дзеркальних поверхонь займе лічені секунди. Будь-який матеріал легко налаштовується, оскільки редактор має безліч параметрів і навіть є можливість створювати його з одного і більше матеріалів.

На рис. 2.4 наведено приклад накладання матеріалів об'єкта.

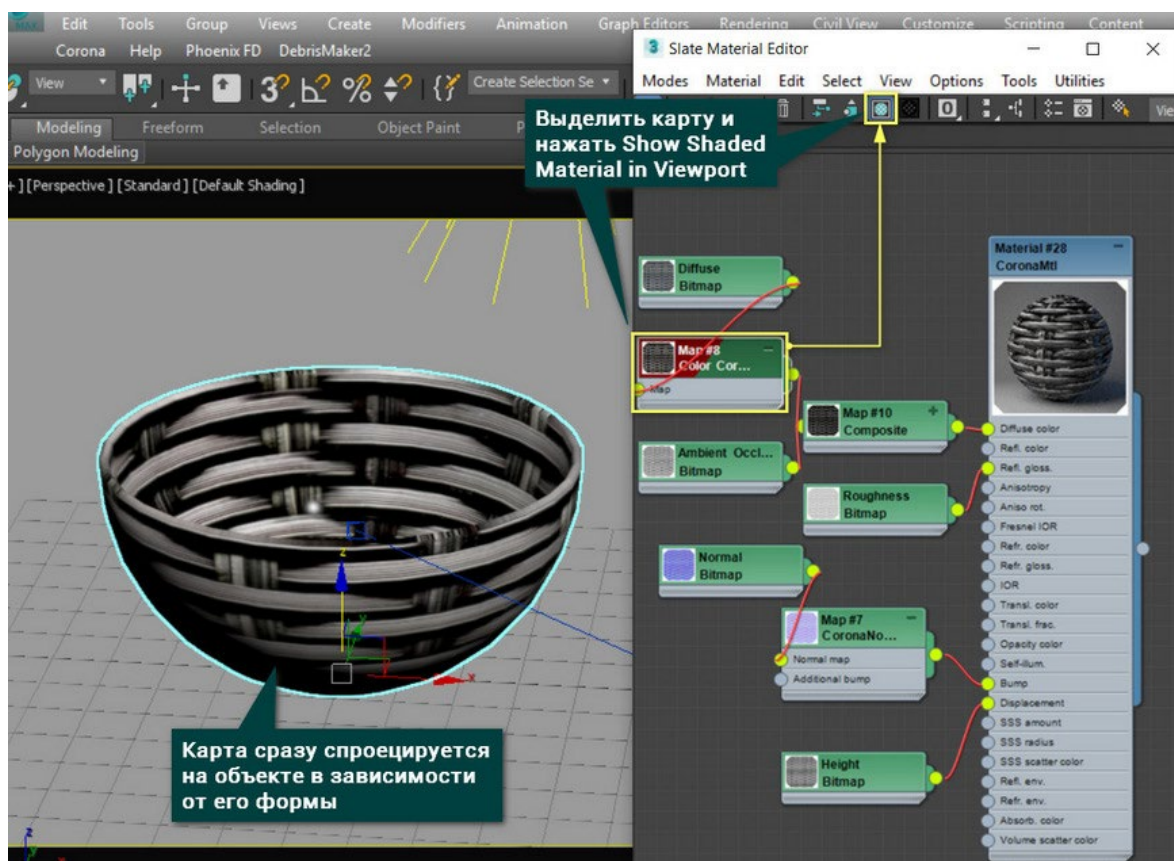


Рис. 2.4. Приклад накладання матеріалів у 3Ds Max

Щоб досягти ефекту реалістичності, застосовуються різні засоби та методи візуалізації. Наприклад, у програмі використовується метод трасування променів, за допомогою якого формується правдоподібне відображення та заломлення світла. Можна створювати атмосферні ефекти (такі, як туман чи вогонь), ефекти природного та штучного фотореалістичного освітлення. І говорячи про системи рендерингу, мається можливість використовувати або вбудований в 3Ds Max візуалізатор (такий як Arnold), або сторонні візуалізатори, створені незалежними розробниками (наприклад, Mental Ray від NVIDIA).

У 3Ds Max досить дружній інтерфейс – на робочій панелі розміщується мінімальна кількість клавіш, необхідних у роботі. Також можна додавати і видаляти клавіші або створювати своє робоче середовище або вибрати з наявних.

Велика увага приділяється розвитку інструментарію для створення анімації. Анімація з ключових кадрів, процедурна анімація, обмежена анімація – це неповний список всіх можливих варіантів змусити рухатися об'єкти. Є можливості управління скелетною деформацією, створення швидкої анімації двоногих істот, управління фізичними силами, які діють персонажів. І чого стоїть можливість створення поведінкової моделі натовпу для анімації відразу сотень об'єктів.

У 3Ds Max міститься величезна кількість модифікаторів. Об'єкти можна згладжувати, скручувати, створювати фаску, відбивати, накладати шуми, редагувати сітку лише на рівні полігонів, проводити різні деформації тощо. Більше того, кожен модифікатор дозволяє змінювати свої параметри, що відкриває неймовірний простір для творчості та реалізації будь-якої ідеї.

3Ds Max містить модулі для роботи з різними системами частинок, будь то сніг або бризки. В основу управління їх характеристиками та динамікою покладено реальні фізичні закони. Саме середовище 3Ds Max дозволяє не тільки моделювати персонажів, але і створювати дуже реалістичні предмети одягу. Причому, крім створення та дизайну одягу, спеціальні вбудовані модулі дозволяють анімувати будь-які об'єкти одягу, створюючи при цьому потрібні візуальні ефекти (створення складок та деформацій на згинах, ефект мокрого або липкого одягу, різні механічні пошкодження). Також програма має модифікатори для імітації волосяного та хутряного покриву. Можливості створення ефектів стрижки та зачісування, руху відповідно до заданих параметрів жорсткості, вологості тощо, а кожну сцену при анімації можуть супроводжувати звукові ефекти. Причому програма підтримує різні звукові формати.

Cinema 4D (скорочено – C4D), що на рис. 2.5 – це універсальна комплексна програма тривимірного моделювання, що дозволяє створювати та редагувати тривимірні об'єкти та ефекти. Вона дозволяє проводити візуалізацію об'єктів методом Гуро. Це метод зафарбовування (затінення), який призначений для створення ілюзії гладкої криволінійної поверхні, описаної у вигляді полігональної сітки з плоскими гранями, шляхом інтерполяції кольорів граней, що примикають.

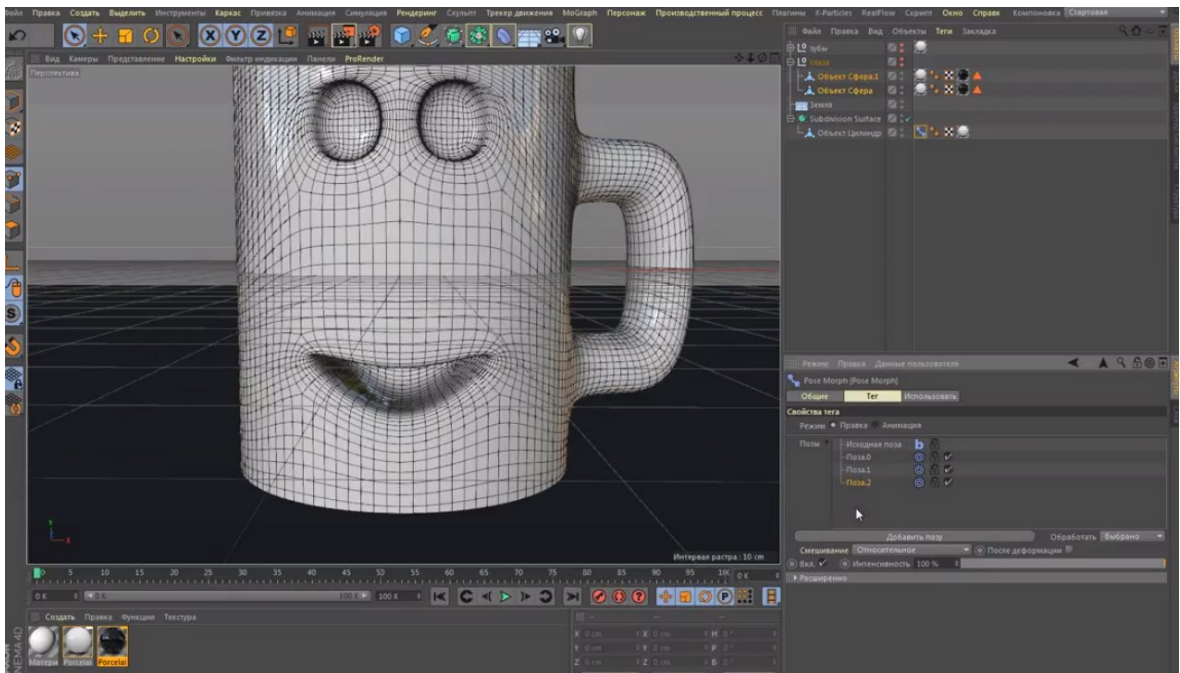


Рис.2.5. Рабочее окружение Cinema 4D

Інтуїтивні та прості для розуміння, функції Cinema 4D та логічно організований інтерфейс допоможуть початківцям оволодіти цим інструментом. Професійні користувачі ніколи не перестають дивуватися, наскільки легко працюється в Cinema 4D. Численні уроки та глобальна, висококваліфікована спільнота відіграють важливу роль у підвищенні професійного рівня користувачів-початківців.

Cinema 4D є основою робочого процесу користувачів. Тому розробники від початку намагаються, щоб нові функції працювали інтуїтивно і знаходилися саме там, де користувачі очікували їх знайти. Компонування Cinema 4D може бути легко змінено для точної відповідності потреб користувача і збережено для подальшого використання.

Cinema 4D надає численні настройки для недеструктивної роботи: параметричне моделювання, процедурні шейдери, текстури та інше. А система дублів Cinema 4D дозволяє зберігати та керувати безліччю версій сцени в одному єдиному файлі.

Cinema 4D є ідеальним пакетом для всіх художників тривимірної графіки, які хочуть досягати захоплюючих результатів швидко та без проблем. Початківці поряд з досвідченими професіоналами можуть скористатися широким набором інструментів і функцій Cinema 4D для швидкого досягнення приголомшливих результатів. Легендарна надійність Cinema 4D робить її відмінним додатком для вимогливої сфери 3D, що швидко змінюється. А асортимент різних пакетів із привабливими цінами відповідає будь-яким вимогам художників.

Також Cinema 4D підтримує високоякісний рендеринг та анімацію.

Відмінною рисою програми ZBrush є імітація «ліплення» 3D-скульптури. Цей процес посилений двигуном 3D-рендерінгу, який працює в режимі реального часу, що значно полегшує функцію створення моделей. ZBrush виділяється серед інших пакетів для 3D-моделювання ширшим і просунутим набором інструментів для скульптингу.

На рис. 2.6 наведено приклад того, як виглядає ZBrush.

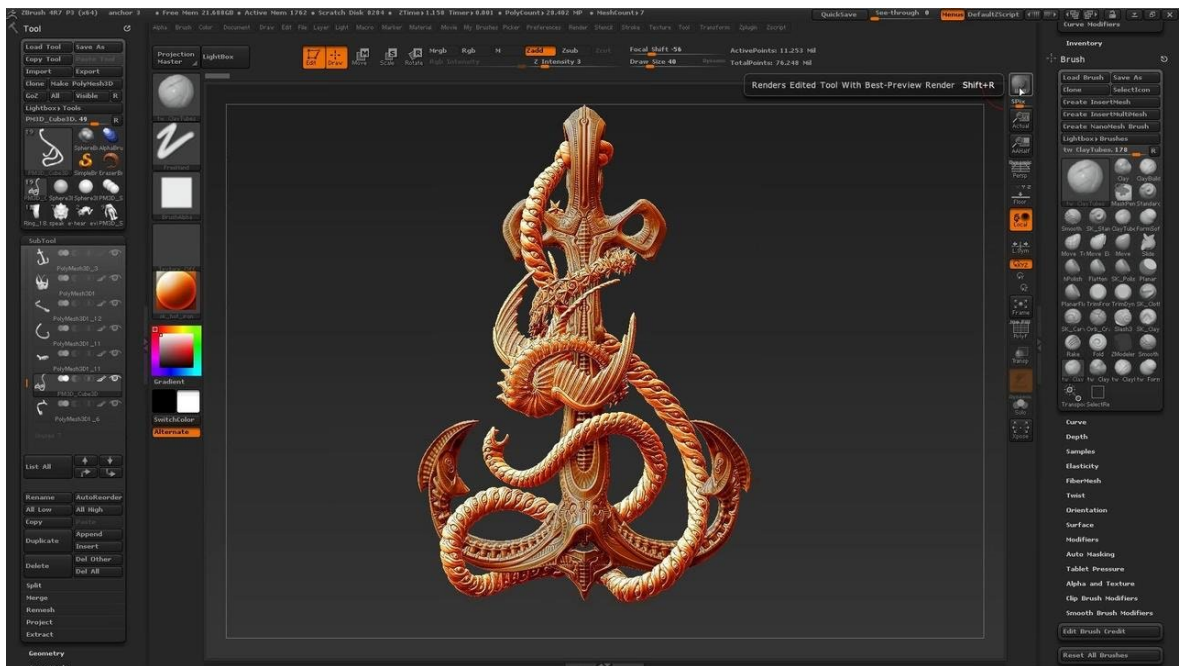


Рис. 2.6. Середовище 3d моделювання ZBrush

Цифрове ліплення є найбільш ефективним при створенні органічних об'єктів, таких як люди, тварини, і навіть рослини, однак це не єдине застосування запропонованих інструментів. Такий метод моделювання підходить для об'єктів, яким не потрібна висока точність деталей, проте необхідні форми, фактура і текстура поверхні. Так, величезний набір пензлів дозволяє досягти максимальної реалістичності під час створення різних «живих» персонажів, дозволяючи опрацювати текстури поверхні, і навіть розфарбувати модель у редакторі. Крім того, у програмі є інструменти для накладання текстур та фінальної візуалізації, що доповнює функціонал програми.

Для текстурювання ZBrush використовує техніку поліпейнту, засновану на роздільній здатності моделі. Тому для детальної текстури потрібна високополігональна сітка. Крім того, текстури як такої не існує, поки її не експортувати. З іншого боку, не потрібно хвилюватися про розгорнення, а ліпити та розфарбовувати модель можна одночасно.

3D-Coat – це редактор, який підходить для таких методів, як твердотільне моделювання та скульптинг. Ця програма дозволяє створювати як грубі низькополігональні моделі, так і деталізовані органічні об'єкти. При роботі можна користуватися автоматичною або ручною топологією, створювати карти розгорток, текстурувати поверхню об'єкта в інтерактивному режимі, а також робити рендеринг у якості статичних сцен та створювати кругову анімацію моделей.

На рис. 2.7 наведено приклад того, як виглядає 3D-Coat.

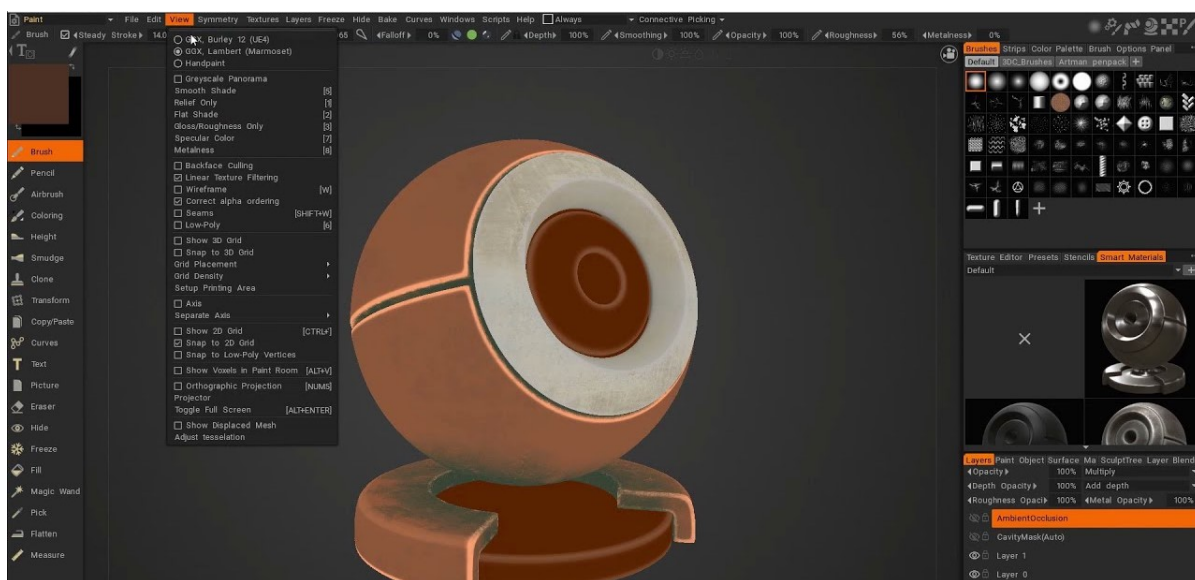


Рис. 2.7. Робоче середовище 3d моделювання 3D-Coat

У програмі 3D-Coat може здійснюватися імпорт моделей, створених за допомогою інших програмних пакетів, для подальшого перетворення. Даний редактор може використовуватися для уточнення або додавання різних деталей, створення розгорток та різних карток (заміщення, відблисків, дифузійного кольору тощо). Після додавання модифікацій вони можуть бути перенесені у вихідне програмне забезпечення: у файлі зберігатимуться розгортки, текстури та інша інформація.

Що ж, тепер можна створити порівняльну характеристику усіх названих до цього середовищ 3D-моделювання, завдяки якій нарешті визначиться, який саме 3D-редактор буде використано в даній кваліфікаційній роботі для створення тривимірних моделей.

Порівняльна характеристика буде наведена у вигляді таблиці 2.1.

Таблиця 2.1.

Порівняльна характеристика 3D-редакторів

Назва	Ціна (за 1 рік ліцензії)	Наявність пробної версії	Складність освоєння	Відкритий код	Специфіка
Autodesk 3Ds Max	1,875 USD	Студентська ліцензія на 3 роки	Так	Ні	Ігри та візуалізація для дизайнерів
Autodesk Maya	1,875 USD	Студентська ліцензія на 1 рік	Так	Ні	Анімація та спецефекти
Blender	Безкоштовно	–	Ні	Так	Моделювання, скульптинг, анімація
ZBrush	430 USD	Пробна версія на 30 днів	Так	Ні	Цифровий скульптинг для кіно- та ігрової промисловості
3D-Coat	186 USD	Пробна версія на 30 днів	Ні	Ні	Цифровий скульптинг
Cinema 4D	802 USD	Пробна версія на 14 днів	Так	Ні	моделювання, редагування об'єктів та створення ефектів

В результаті аналізу існуючих редакторів для створення 3D–моделей було обрано програму Autodesk 3Ds Max. Підставою для вибору стало те, що 3Ds Max є однією з найпопулярніших програм професійного рівня, а також має безкоштовну студентську версію. Окрім того на цей вибір вплинув той факт, що раніше вже був досвід ознайомлення із цим редактором.

2.3.2. Порівняння редакторів для створення текстур об'єктів.

Отже з редактором для створення тривимірних моделей розібралися. Однак використовувати голі 3d моделі не є гарною ідеєю. Набагато кращою ідеєю буде створити необхідні текстури, які будуть використані для створених моделей. Але без необхідного інструментарію виконати це буде тяжко. Тому й необхідно обрати середовище для створення цих самих текстур.

Почнемо з самого поняття, що таке текстури та текстурування:

Текстура – це плоске 2D–зображення, що накладається на полігони, з яких складаються 3D–структури, що знаходяться в кадрі.

Текстурування – один із найважливіших етапів моделювання об'єктів, незважаючи на те, що на ньому лише обчислюються та змінюються кольори двовимірної сітки різнокольорових блоків. Текстура зазвичай підбирається на етапі роботи з матеріалом і має вирішальне значення для того, як саме модель буде виглядати вкінці. Неякісні текстури з низькою роздільною здатністю можуть зіпсувати всю роботу, тоді як хороші безшовні текстури можуть наблизити до фотореалізму. Зазвичай під час роботи з матеріалом крім самої текстури – у плоскому зображенні створюються додатково карти нормалей, створюють на об'єкті мікрорельєф і додають йому реалістичності. Наприклад, якщо матеріал відполірованого металу може бути дуже гладким, то такі матеріали як метал, що зістарився, камінь, дерево тощо, мають певні нерівності, виступи, подряпини і так далі.

Для створення графічних елементів інтерфейсу користувача, а також для обробки текстур використовуються різні растрові редактори. Для розгляду будуть наведені деякі з них.

Adobe Photoshop – багатофункціональний графічний редактор, розроблений та розповсюджуваний фірмою Adobe. Продукт є лідером ринку в галузі комерційних засобів редагування растрових зображень та найвідомішим продуктом фірми Adobe.

Даний редактор містить безліч різних інструментів як для обробки фотографій, так і для створення колажів, фотоарту (вид цифрового образотворчого мистецтва, що поєднує фотографію та комп'ютерну графіку). Цей редактор підходить для цифрового живопису, так як містить велику бібліотеку пензлів та потужні налаштування для них, цифрові палітри та текстури. Завдяки наявності векторних інструментів, цей редактор підходить для створення елементів графічного дизайну, текстур та інтерфейсів користувача.

Приклад того, як виглядає цей редактор можна побачити на рис. 2.8.

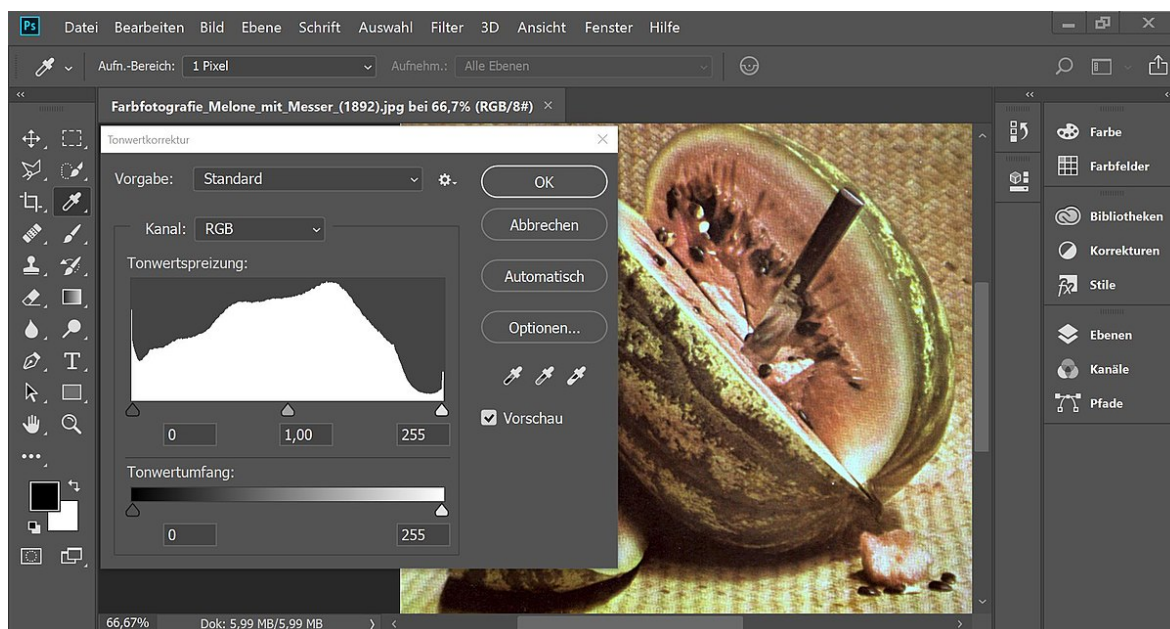


Рис. 2.8. Редактор зображень Adobe Photoshop

Adobe Photoshop – головний інструмент комп'ютерної графіки, який використовують для всіх видів комерційної та безкоштовної обробки медіа файлів. Головна суть Photoshop полягає в тому, щоб скоригувати зображення, покращити його сприйняття, надати фото та відео чіткості та гармонійності. Ця програма дозволяє посилити ефект від картинки, створити складні та красиві конструкції із простих елементів.

SAI Paint Tool – високоякісне та легке програмне забезпечення, орієнтоване насамперед на цифрових художників. Цілком підтримує підключення графічних планшетів, забезпечує стабільну роботу. Дана програма дозволяє зберігати зображення у форматі *.psd, що дозволяє переносити роботу в Adobe Photoshop і назад зі збереженням шарів і режимів накладання.

Так як ця програма призначена для художників, то вона не містить складних інструментів обробки, але містить потужні інструменти, що імітують різні матеріали традиційного живопису (фарби, олівці, маркери, олівці тощо).

Приклад того, як виглядає це ПЗ можна побачити на рис. 2.9.

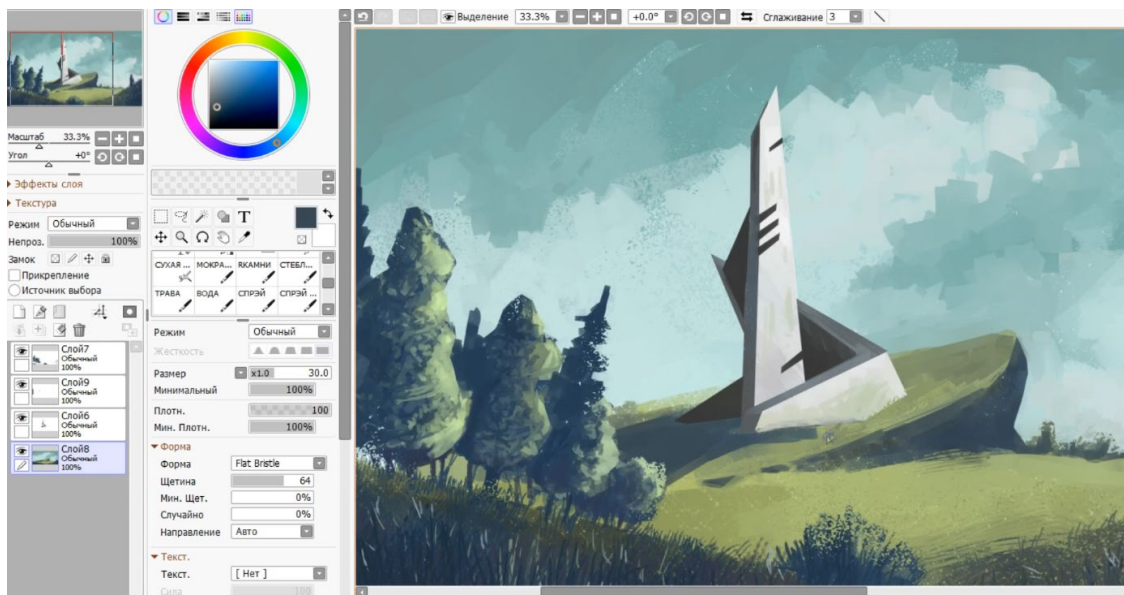


Рис. 2.9. Робоче середовище SAI Paint Tool

Photo Pos Pro – це ще одна програма для обробки фотографій. Представлена у двох варіантах – безкоштовній та професіональній. Інтерфейс доволі простий та зручний, так що обробити фото можна швидко та зручно.

Photo Pos Pro дозволяє:

- робити корекцію кольорів і використовувати «розумні» фільтри для обробки фото: видаляти фон або небажані об'єкти;
- редагувати у RAW форматі;
- створювати фотоколажі;
- малювати за допомогою растрових пензлів;
- додавати текст зі своїми шрифтами та застосовувати до нього ефекти.

Приклад того, як виглядає цей редактор є на рис. 2.10.

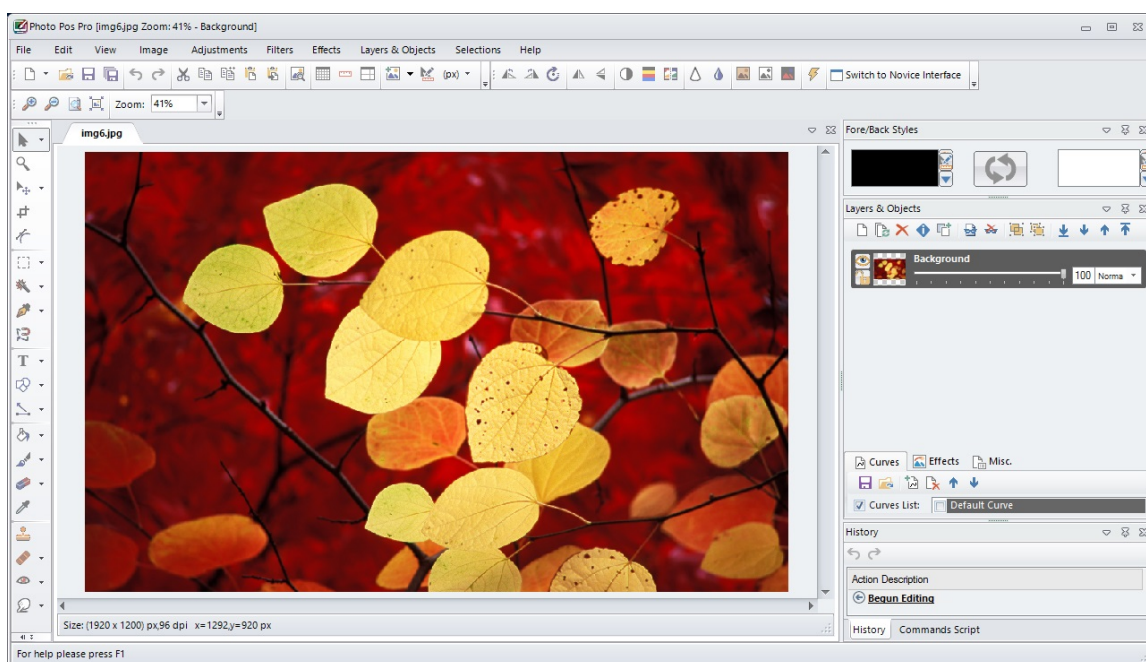


Рис. 2.10. Робоче середовище Photo Pos Pro

Однак у нього є достатньо великий недолік – безкоштовна версія не може надати достатню якість інструментарію для обробки та створення зображень.

GIMP (GNU Image Manipulation Program) – це безкоштовний редактор зображень з відкритим вихідним кодом. Ця програма підходить для графічних дизайнерів, фотографів та ілюстраторів, так як містить багато складних

інструментів. Крім того, завдяки відкритому коду існує безліч налаштувань та плагінів, які допомагають розширити можливості програми.

GIMP містить інструменти для ретушування та відновлення фотографій високої якості, підходить для створення колажів та фотоартінгу. Дизайнери можуть створювати в GIMP значки, елементи графічного дизайну та інтерфейсу користувача тощо. Для ілюстраторів у цій програмі також є безліч інструментів – пензлі, градієнти, кольорові палітри, що настроюються.

Приклад того, як виглядає цей редактор є на рис. 2.11.

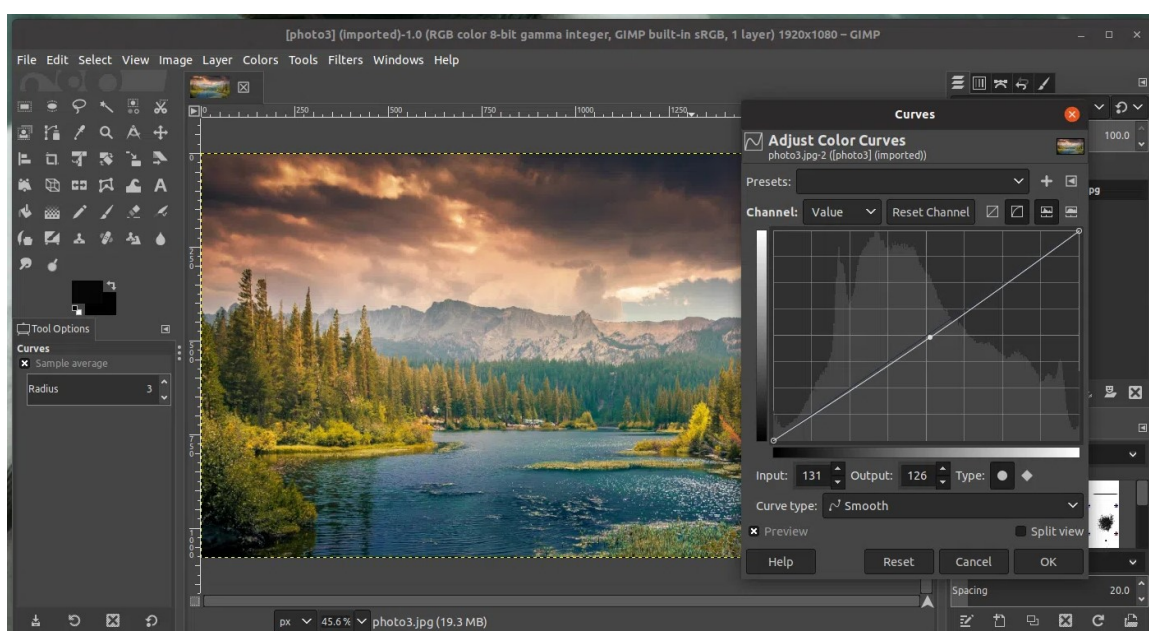


Рис. 2.11. Редактор зображень GIMP

Тепер можна створити порівняльну характеристику усіх названих до цього редакторів зображень, завдяки якій нарешті визначиться, яке саме середовище буде використано в даній кваліфікаційній роботі для створення текстури, яка буде накладена на створені 3d моделі.

Порівняльна характеристика буде наведена у вигляді таблиці 2.2.

Порівняльна характеристика редакторів зображень

Назва	Ліцензія	Наявність пробної версії	Специфіка	Складність у освоєнні	Платформа / ОС
Adobe Photoshop	300 USD	Пробна версія на 7 днів	Обробка фотографій	Середня	Windows, macOS, Android
SAI Paint Tool	50 USD	–	Цифровий живопис	Легка	Microsoft Windows
Photo Pos Pro	Безкоштовна	–	Обробка фотографій	Легка	Microsoft Windows
GIMP	Безкоштовна	–	Обробка фотографій	Середня	Linux, Mac OS, Windows

В результаті аналізу редакторів зображень вибір був зроблений на користь програм Adobe Photoshop, так як це програма професійного рівня, в якій містяться всі необхідні для створення текстур та обробки зображень інструменти. Крім того, ця програма користується популярністю серед професіоналів, а значить вона досить надійна. Окрім того на цей вибір вплинув той факт, що раніше вже був досвід ознайомлення і використання цього редактору.

2.3.3. Порівняння ігрових двигунів для розробки.

Нарешті з інструментарієм для моделювання та текстурної обробки покінчено. Наразі єдине, що залишилось – це обрати необхідний ігровий двигун, який, використовуючи отримані моделі та текстури, зможе створити безпосередньо сам застосунок. Для цього найліпшим варіантом буде здійснення порівняння цих самих двигунів, які допоможуть з реалізацією програмного модуля. З величезного списку ігрових двигунів було обрано 4 найбільш підходящі програмні модулі для реалізації поставлених цілей, а саме: Unity3D, CryEngine 3, Source та Unreal Engine 4. Розглянемо докладніше кожен з них.

Unity3D – це найпопулярніша у світі платформа для створення багатоплатформних 2D-і 3D-ігор та інтерактивного контенту” - впливає з опису двигуна на офіційному сайті програмного продукту. Мінімальні системні вимоги для двигуна такі - для розробки: операційна система Windows 7 SP1+, 8, 10, тільки 64-бітна версія або Mac OS X 10.11+; графічний процесор: відеокарта із підтримкою DX10 (версія шейдерів 4.0); серверні версії Windows & OS X не протестовані; центральний процесор: підтримка набору інструкцій SSE2; графічний процесор: відеокарта із підтримкою DX10 (версія шейдерів 4.0). Решта залежить, головним чином, від складності проектів, що створюються. Додаткові вимоги до платформи розробки – універсальна платформа Windows: Windows 10 (64-розрядна), Visual Studio 2015 з компонентами C++ або пізнішої версії та пакет SDK для Windows 10. Приклад того, як виглядає середовище цієї платформи є на рис. 2.12.

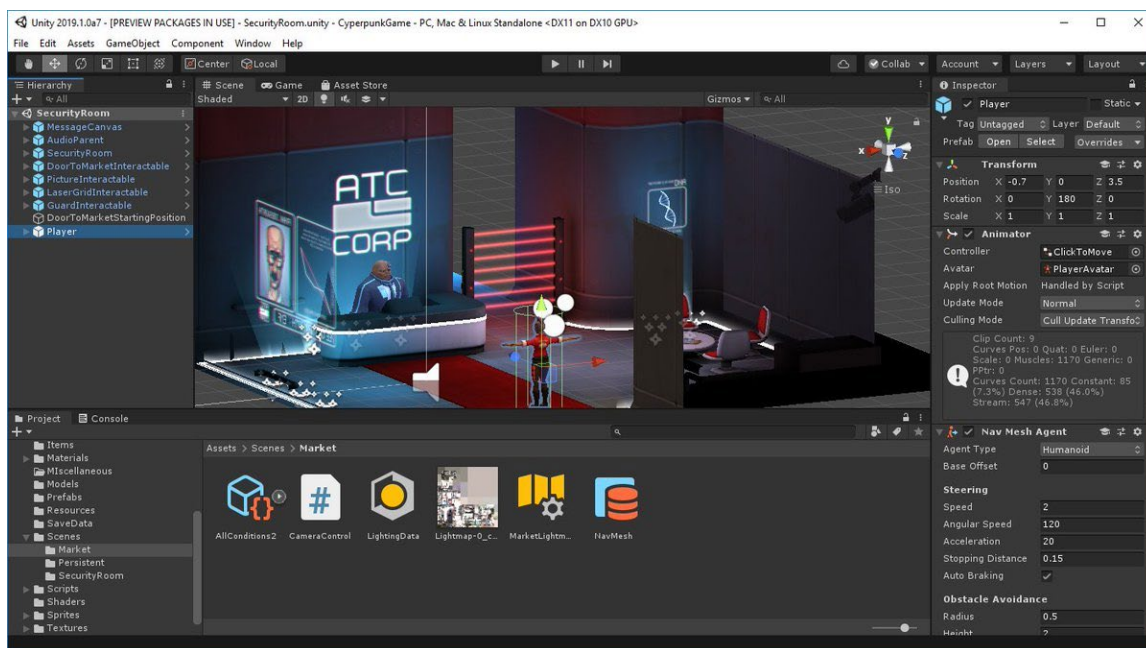


Рис. 2.12. Зображення вікна редактора Unity3D.

Безкоштовна версія Unity має мізерний функціонал, але не вимагає жодних відрахувань з прибутку: ні з проекту, ні за моделлю revenue share. Однак це правило працює доти, доки дохід з проекту не перевищує 100 тисяч доларів США на рік. Права на весь контент, створений за допомогою підписки Unity, належить автору, навіть при припиненні підписки. Підтримує 25+ платформ, включаючи мобільні пристрої, ПК, консолі, ТБ, VR, AR та Інтернет. Також існує можливість застосування сторонніх плагінів. Unity 3D має свій редактор тривимірної графіки та часто його можливостей достатньо, щоб зробити повноцінну ігрову модель.

Спільнота Unity дозволяє отримувати допомогу та обговорювати рішення з досвідченими користувачами Unity, а також ділитися своїми знаннями. Для цього на сайті Unity є спеціальні розділи. "Форуми" є центральним майданчиком суспільства для різноманітних обговорень. Відповіді є майданчиком для розміщення конкретних питань та відповідей щодо Unity. Вітаються повідомлення як експертів, так і тих, хто тут тільки нещодавно з'явився, які допомагають один одному з Unity. Можна використовувати "Баг-трекер" для отримання відомостей про статус успішно відтворених багів Unity.

Згідно з інформацією на офіційному сайті двигуна, CryEngine 3 – це найпотужніша платформа для розробки ігор. Розроблена німецькою компанією з обмеженою відповідальністю Crytek, є наступною покращеною версією двигуна CryEngine 2. CryEngine 3 є кросплатформовим двигуном – він спочатку орієнтований на IBM PC-сумісні комп'ютери та ігрові консолі Microsoft Xbox 360 та Sony PlayStation 3, а також на їх наступні версії. Приклад того, як виглядає середовище цієї платформи є на рис. 2.13.

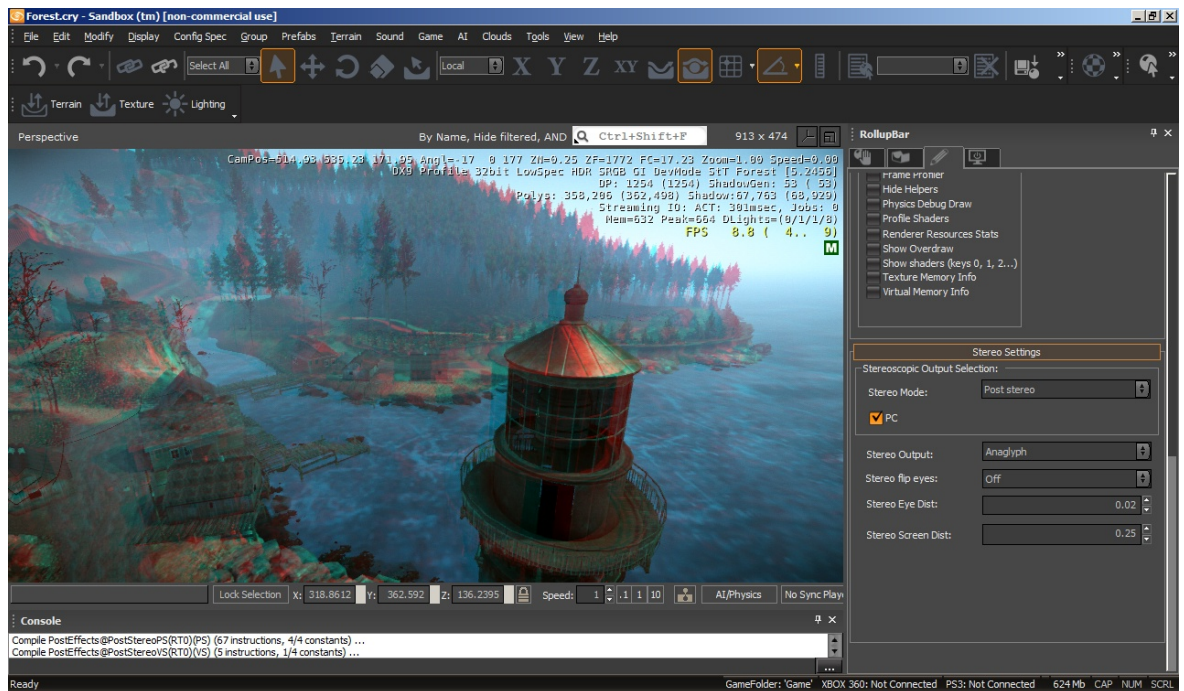


Рис. 2.13. Зображення вікна редактора CryEngine 3.

Необхідні мінімальні характеристики системи для нормальної роботи двигуна: операційна система Windows XP SP2 або Windows Vista або Windows 7, 64-бітний процесор (багатоядерність рекомендується), 2 гігабайти оперативної пам'яті (4 гігабайти рекомендується), відеокарта з підтримкою покоління GeForce 6600 і вище, також аналогічні ряду ATI/AMD). Як повідомляє нам Crytek, вимоги до продуктивності CPU багато в чому залежать від навантаження на підсистеми AI та фізики двигуна. Ціна використання двигуна складає 10 доларів США на місяць.

Мінуси використання двигуна CryEngine 3:

- безліч багів на клієнті, сервері та в редакторі. Потрібно доопрацьовувати двигун щонайменше 50% часу розробки, а не просто робити на ньому гру;
- відсутність своєчасної підтримки (навіть платної). Багато труднощів доведеться вирішувати самостійно і поодиночі. При цьому їхня кількість дуже велика. Навіть у разі свідчення про баги їх можуть виправляти роками;
- документація для багатьох модулів застаріла чи відсутня;

- спільнота дуже мала та неактивна;
- невеликий асортимент магазину асетів;
- користувач SDK повинен мати стабільне підключення до Інтернету.

У 2017 році Crytek офіційно заявив про відмову від подальшого розвитку GameSDK. І зараз двигун перебуває у проміжній стадії розвитку. Багато старих інструментів видаляються, код тотально переписується. Тепер готовий новий Shooter Game, але він все ще на ранній стадії розвитку.

Двигун Source, розроблений компанією Valve Corporation, використовується для створення власних ігор. Завдяки гнучкості движка, звичайні юзери створювали модифікації до вже випущених ігор, які переросли в самостійні проекти на цьому ж движку.

До складу двигуна входять інші двигуни, які дозволяють обробляти графіку та фізику. Завдяки кросплатформенності користується популярністю в інших розробників.

Вартість двигуна є конфіденційною інформацією компанії, для отримання повного доступу до двигуна видавцю гри потрібно дотримуватися умов, які також не розголошуються компанією Valve.

Для звичайних користувачів є безкоштовна, але в той же час урізана версія двигуна під назвою SourceSDK, що містить у собі набір утиліт, який дозволяє створювати модифікації до вже створених ігор або створювати нові ігри.

Для створення графічної складової для цього двигуна були створені утиліти:

- Faceposer – це утиліта дозволяє створювати реалістичні лицьові анімації персонажів без використання технології Motion Capture;
- Valve Hammer Editor – утиліта що дозволяє створювати оточення та ландшафт;
- Model Viewer – завдяки цій утиліті користувач зможе створювати моделі персонажів без використання сторонніх графічних редакторів.

Приклад того, як виглядає середовище платформи є на рис. 2.14.

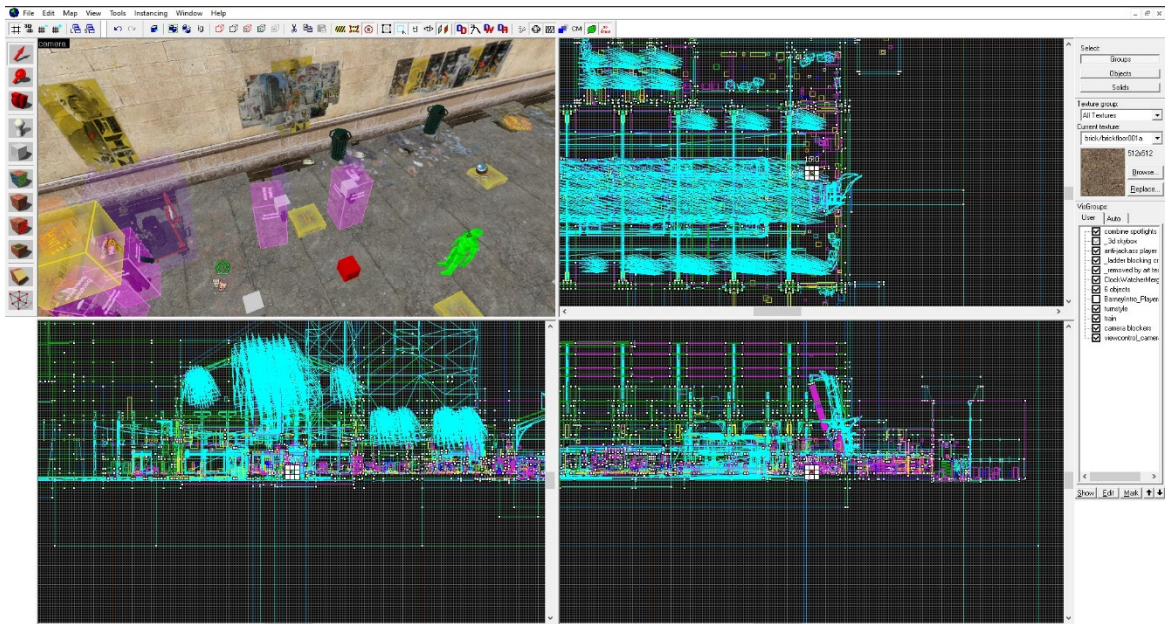


Рис. 2.14. Зображення вікна редактора Source

Unreal Engine – це середовище розробки для створення модулів за допомогою програмування мовою C++. Середовище знаходиться на вільному доступі на сайті розробника, монетизація середовища розробки починається тільки після подолання певного порогу доходності з продажу продукту на його основі. За умови заробітку на реалізації програми більше \$3,000 на квартал, потрібно заплатити за використання двигуна 5% прибутку від реалізації гри. Приклад того, як виглядає середовище цієї платформи є на рис. 2.15.



Рис. 2.15. Зображення вікна редактора Unreal Engine 4

Двигун має велику, активну і постійно зростаючу спільноту, що дозволяє отримати інформацію про проблему. Також є постійна підтримка з боку розробника, є можливість звернутися на підтримку та отримати швидку відповідь. У середині самої програми є відео уроки та графічні моделі для найкращого освоєння програми. Є можливість використовувати приклади з навчальних модулів для створення власних проектів, а також розбирати, змінювати та покращувати їх. Понад те, дозволяється використовувати цей контент у комерційних цілях. Це ж правило поширюється і на безкоштовний контент з Marketplace.

Інтерфейс є гнучким і достатньо зручним у налаштуванні. Для користувача він простий у освоєнні та зрозумілий. Основною мовою програмування є C++, а також можливість використання графічного модуля – Blueprints – для опису логіки програми. Синхронізація з іншими програмними продуктами (MS Visual Studio, Xcode) дозволяє не відволікатися на перенесення даних з однієї програми до іншої.

Розробники Unreal Engine вирішили не закінчувати на створенні тільки двигуна для розробки ігор, а й піти далі, і тому анонсували і створили Epic Games Store - платформу дистрибуції ігор, аналогічній відомому майданчику Steam.

Цей майданчик дозволить розробникам ігор розповсюджувати свої продукти на вигідніших умовах, ніж інші магазини, а також дає різного роду переваги, на кшталт відсутності платної реклами та інструментів для комунікації з гравцями. Більш того, на майданчик можуть потрапити ігри не тільки на базі Unreal Engine, а й на будь-якому другому движку. Цей майданчик поширюватиме ігри на PC та Mac, а найближчим часом обіцяє підтримку й іншим відкритим платформам, у тому числі й Android.

Unreal Engine дає можливість створення проектів для Windows PC, Mac, Linux, iOS та Android, HTML5. Також є інтегрована підтримка віртуальної дійсності для Oculus Rift. Крім цього UE4 підтримує Xbox One та PlayStation 4.

Завдяки відкритому вихідному коду, можна самостійно додати підтримку додаткових пристроїв, або оптимізувати/доповнити вже існуючі.

Unreal Engine має інструменти для створення абсолютно будь-якого проекту. Ігри (2D-3D; RTS, Action-RPG, Shooter, Racing, MMO-ігри та будь-який інший жанр чи напрямок), симулятори і навіть програмне забезпечення. Можна використовувати UE4 для архітектурної візуалізації та багато іншого.

Для більш детального аналізу та остаточного рішення отриманні до цього дані можна занести у таблицю 2.3.

Порівняльна характеристика графічних двигунів.

	Unity 3D	Unreal Engine	Source	CryEngine 3
Ціна	Безкоштовна версія має скудний функціонал	Безкоштовна, % з продажу береться лише з певної суми	Доступ відсутній для для приватного використання	10\$ на місяць
кросплатформність	+	+	+	+
Можливість працювати із C++	+	+	+	–
Інструментарій для 3D графіки	–	+	+	+
використання сторонніх плагінів	+	+	+	–

В результаті аналізу вищенаведених двигунів та їх характеристик, обраний найбільш підходящим для даної роботи двигун – Unreal Engine.

2.3.4. Створення візуальної частини застосунку

Створювати 3D–моделі відбувалося у редакторі 3Ds Max.

Для того, щоб створити тривимірну модель необхідно:

- відкрити програмний інструментарій для 3d моделювання 3Ds Max;
- розпочати новий проект (або створити інший). Приклад наведено на рис. 2.16;
- створити заготовку моделі (можна використати базові заготовки редактору);
- провести обробку, щоб заготовка отримала необхідну моделі форму;

- накласти текстуру, обрану для 3d моделі (будуть використані текстури, створені);
- отримати готову тривимірну модель.

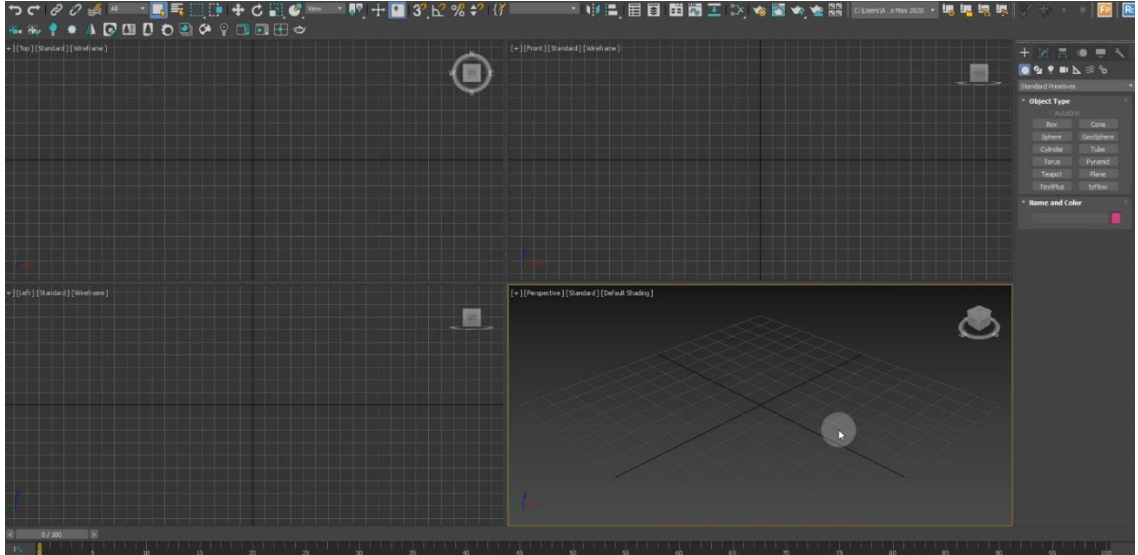


Рис. 2.16. Новий проект у редакторі 3Ds Max

Спочатку змодельємо об'єкт «платформа». Для цього:

- створимо заготівку моделі, яку будемо в процесі обробляти для отримання необхідного результату; (рис. 2.17)

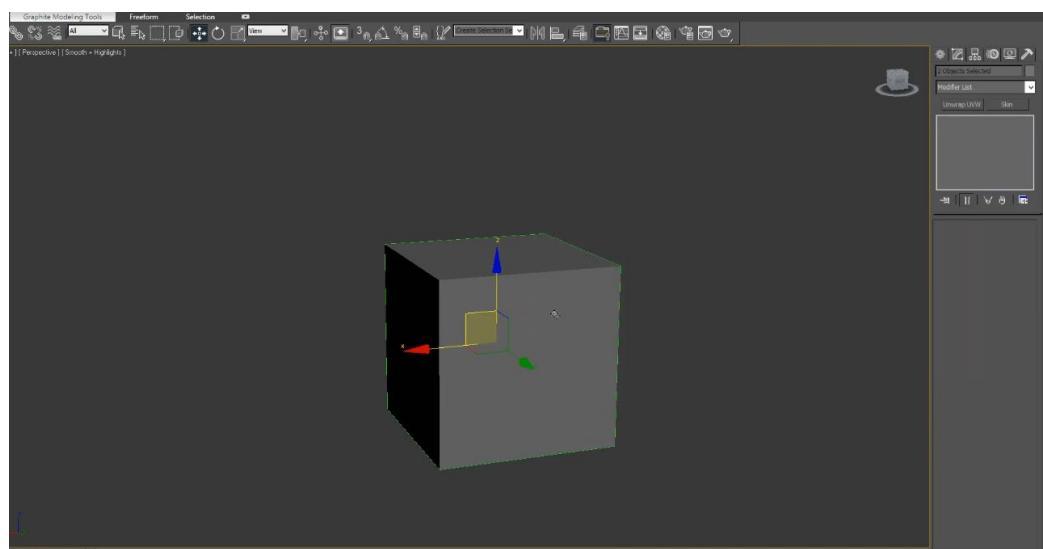


Рис. 2.17. Процес створення моделі «платформа»

- накладемо необхідну для моделі текстуру; (рис. 2.26)
- отримаємо необхідну 3d модель, яку зможемо використати у грі. (рис. 2.18)

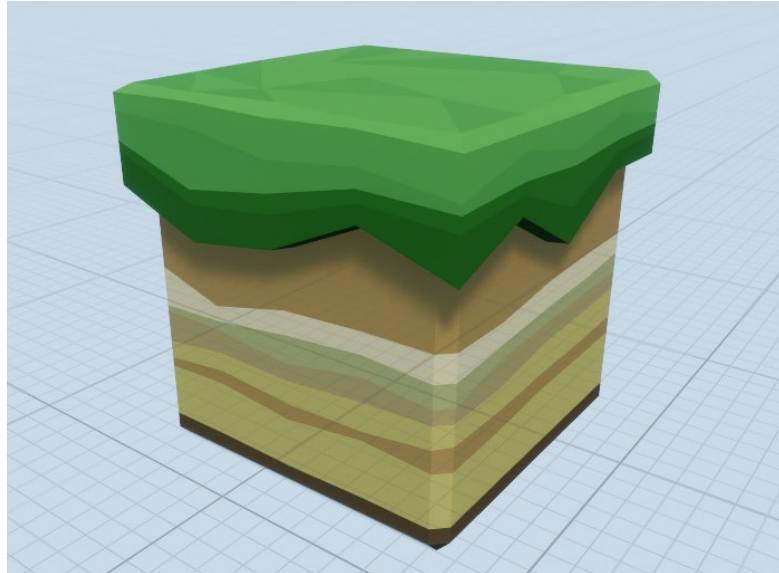


Рис. 2.18. Кінцевий вигляд об'єкту «платформа»

Тепер змодельюємо об'єкт «призова монета». Для цього:

- створимо іншу заготовку моделі, яку будемо в процесі обробляти для отримання необхідного результату. (рис. 2.19)

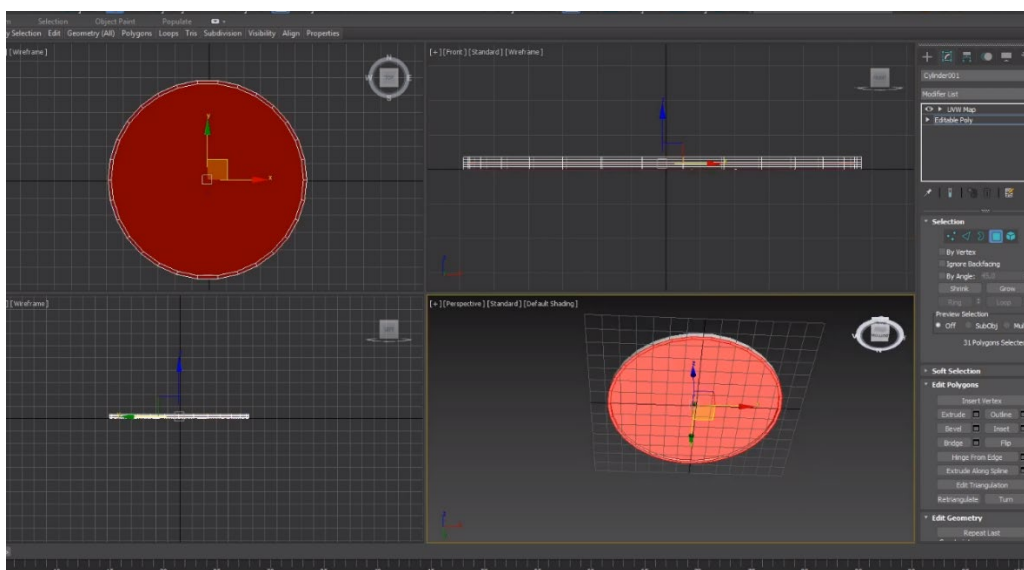


Рис. 2.19. Процес створення моделі «призова монета»

- накладемо необхідну для моделі текстуру; (рис. 2.25)
- отримаємо другу необхідну 3d модель для використання у грі. (рис. 2.20)

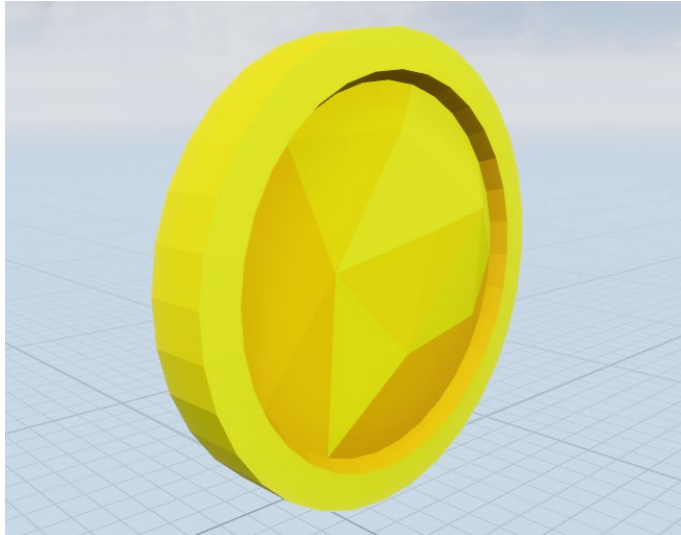


Рис. 2.20. Кінцевий вигляд об'єкту «призова монета»

У кінці змодельюємо об'єкт «сфера». Для цього:

- створимо останню заготовку моделі, яку будемо в процесі обробляти для отримання необхідного результату; (рис. 2.21)

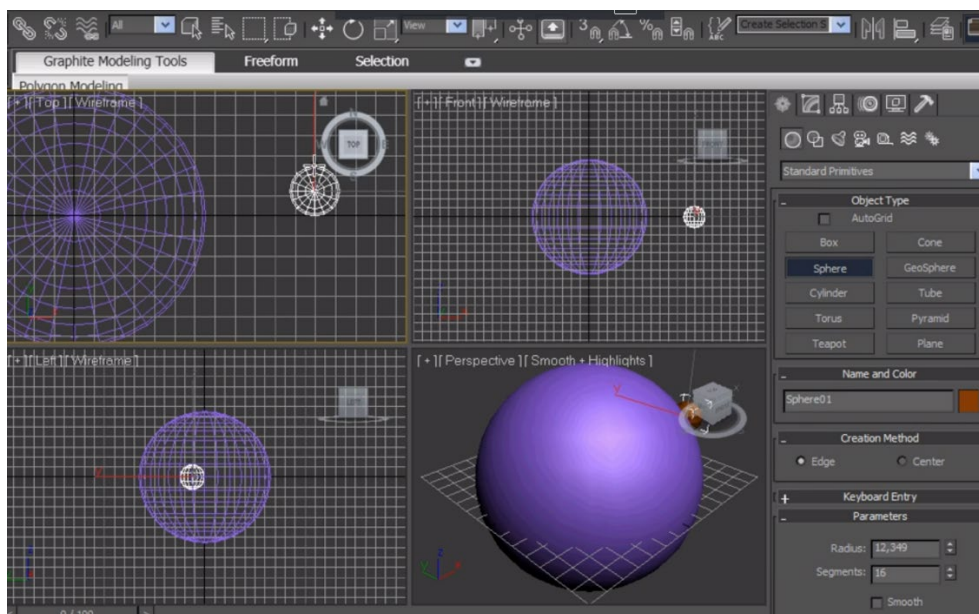


Рис. 2.21. Процес створення моделі «сфера»

- накладемо необхідну для моделі текстуру; (рис. 2.24)
- отримаємо третю необхідну 3d модель для використання у грі. (рис. 2.22)



Рис. 2.22. Кінцевий вигляд об'єкту «сфера»

3D-моделі – це, добре, однак використовувати їх у тому вигляді, який вони мають за базою не є доцільним. Тому для них будуть створенні текстури.

Для створення текстур скористаємося інструментарієм редактора зображень Adobe Photoshop.

Перш за все створимо палітру кольорів, за допомогою якої ігровий двигун зможе розмальовувати чи доповнювати моделі кольорами. (рисунок 2.23.)

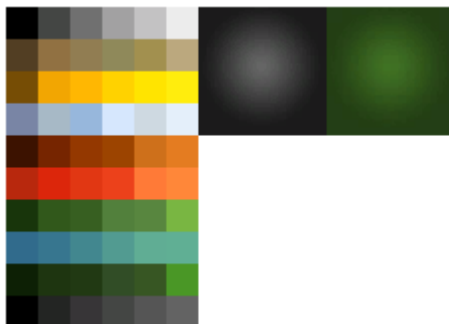


Рис. 2.23. Палітра кольорів

Далі необхідно створити безпосередньо текстури для тривимірних моделей. Вони будуть наведені нижче:

- текстура об'єкту «сфера»;(рис. 2.24)



Рис. 2.24. Текстура об'єкту «сфера»

- текстура об'єкту «призова монета»; (рис. 2.25)



Рис. 2.25. Текстура об'єкту «призова монета»

- текстура об'єкту «платформа»; (рис. 2.26)



Рис. 2.26. Текстура об'єкту «платформа»

Після того, як були створені тривимірні моделі та їм надали текстур, прийшов час для проектування рівня та додавання до нього всіх об'єктів, створених до цього.

Створювати рівень будемо вже на ігровому двигуні Unreal Engine. Для цього прикріпимо до двигуна створені раніше тривимірні моделі, надаємо їм необхідних текстур(приклад цього є на рис. 2.27), підправимо ці моделі, якщо це необхідно, і у нас вже маються готові об'єкти.

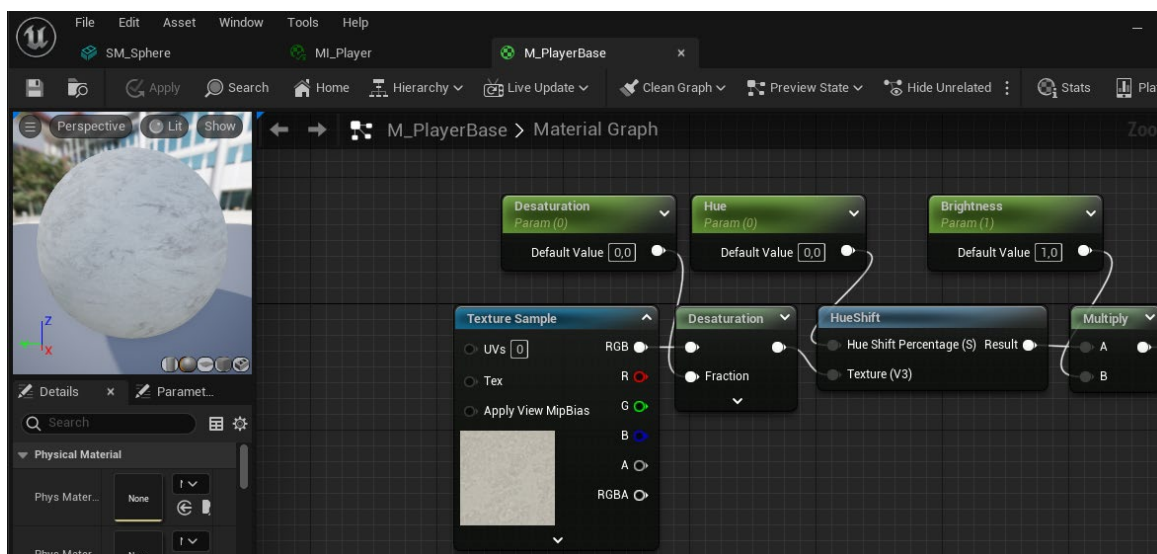


Рис. 2.27. надаємо текстури об'єкту «сфера»

Тепер можна почати створення. Для цього необхідно розташувати першу об'єкт – платформу. (рис. 2.28)

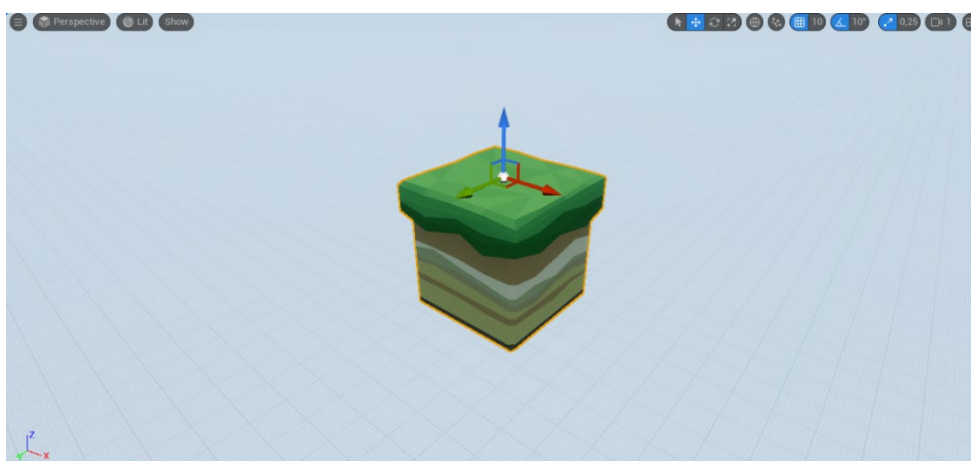


Рис. 2.28. Перша «платформа»

Далі з цього об'єкту можна зробити дублікат. Якщо натиснути та держати клавішу ALT, та в той же час натиснути на одну зі стрілочок, які відображають координатну вісь, то можна побачити, як при пересуванні миші буде створена та переміщена на інше місце копія цього об'єкту. (рис. 2.29)

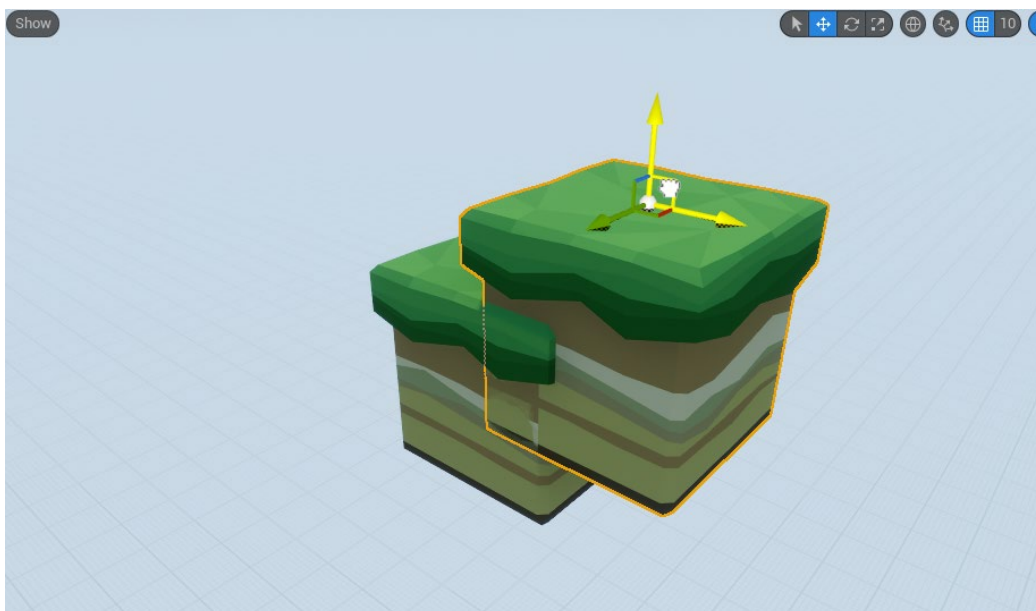


Рис. 2.29. дублікат об'єкту «платформа»

Даний спосіб підходить для будь-яких об'єктів, тому він може дуже сильно полегшати працю.

Тепер створюємо безпосередньо сам ландшафт рівня. (рис. 2.30)

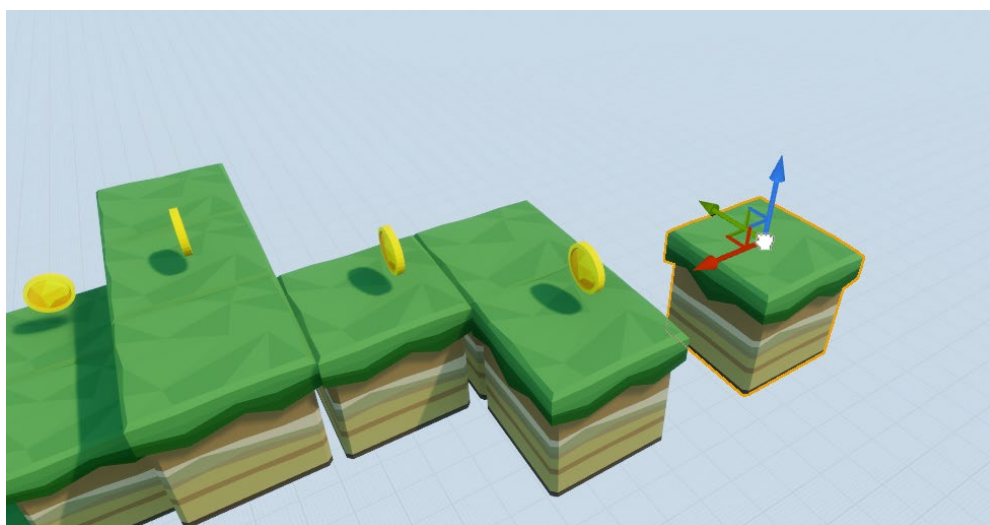


Рис. 2.30. Процес створення ландшафту рівня

Для того, щоб гравець не міг стрибнути із ігрового майданчику було прийняте рішення скористатися невидимою стіною. Невидима стіна – це певний фізичний об’єкт, який не дає можливості персонажу рухатися в її бік(як і звичайна стіна). Щоб додати її необхідно у вкладці Place Actors скористатися пошуком, в якому знайти елемент Blocking Volume. Далі його можна перенести до робочого екрану де й використовувати.(рис. 2.31)

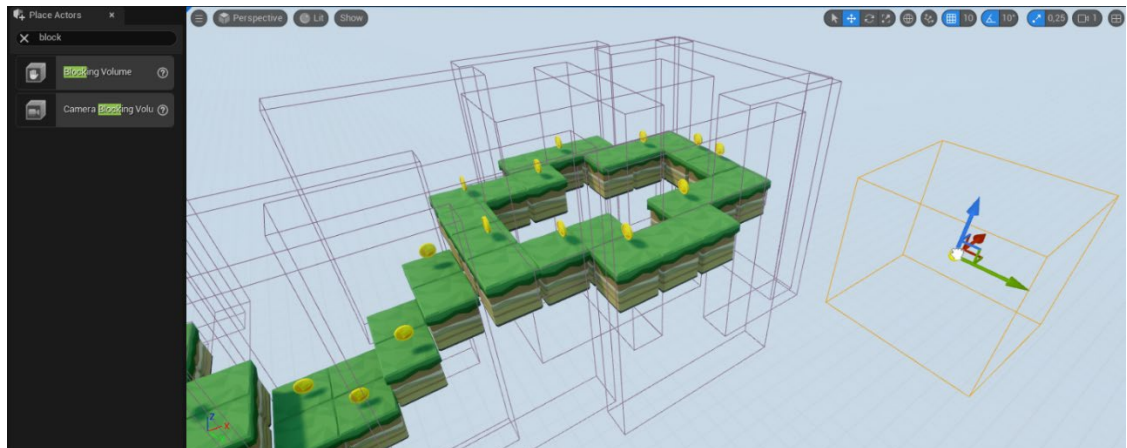


Рис. 2.31. Перенесення невидимої стіни на рівень гри

Тепер єдине, що залишилося додати – це точку, де гравець починає свою гру, освітлення, та початковий вид камери.(рис. 2.32)

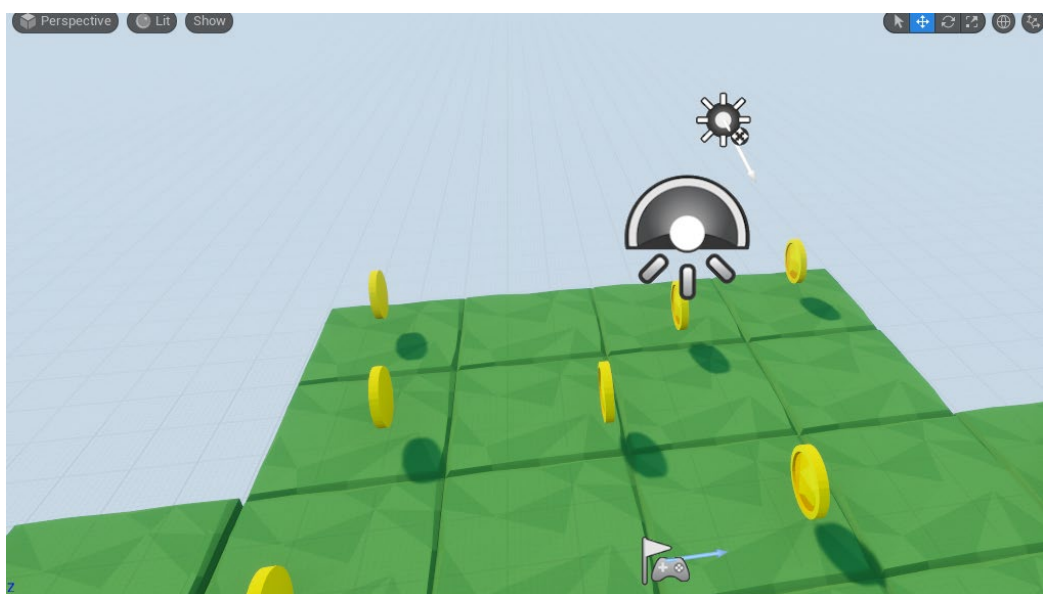


Рис. 2.32. Точка початку, камера та освітлення

Кінцевий же вигляд можна розглянути на рис. 2.33.

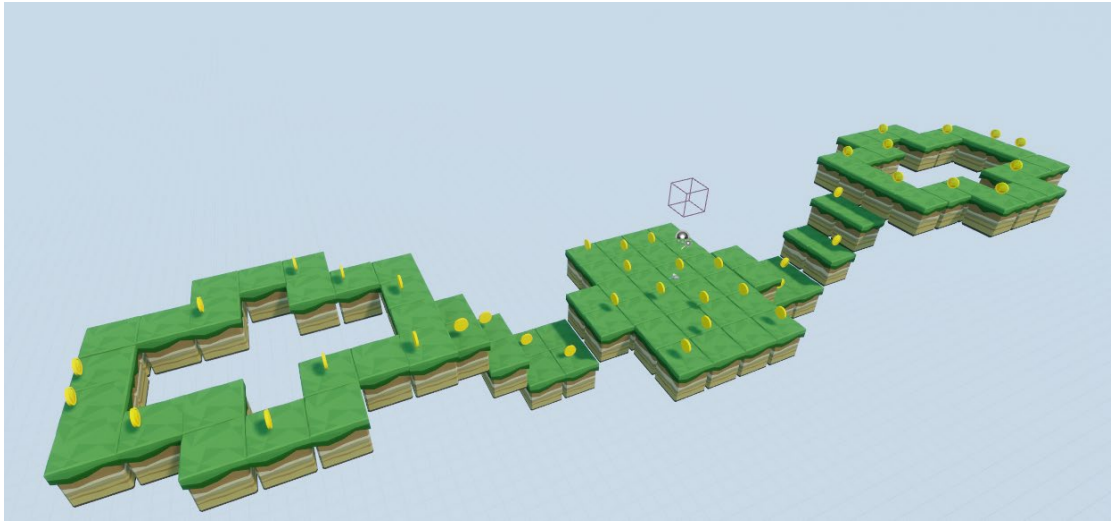


Рис. 2.33. Повністю створений рівень гри

2.4. Опис структури системи та алгоритмів її функціонування

На рис. 2.34 була неведена ілюстрація того, як саме працює створений застосунок.

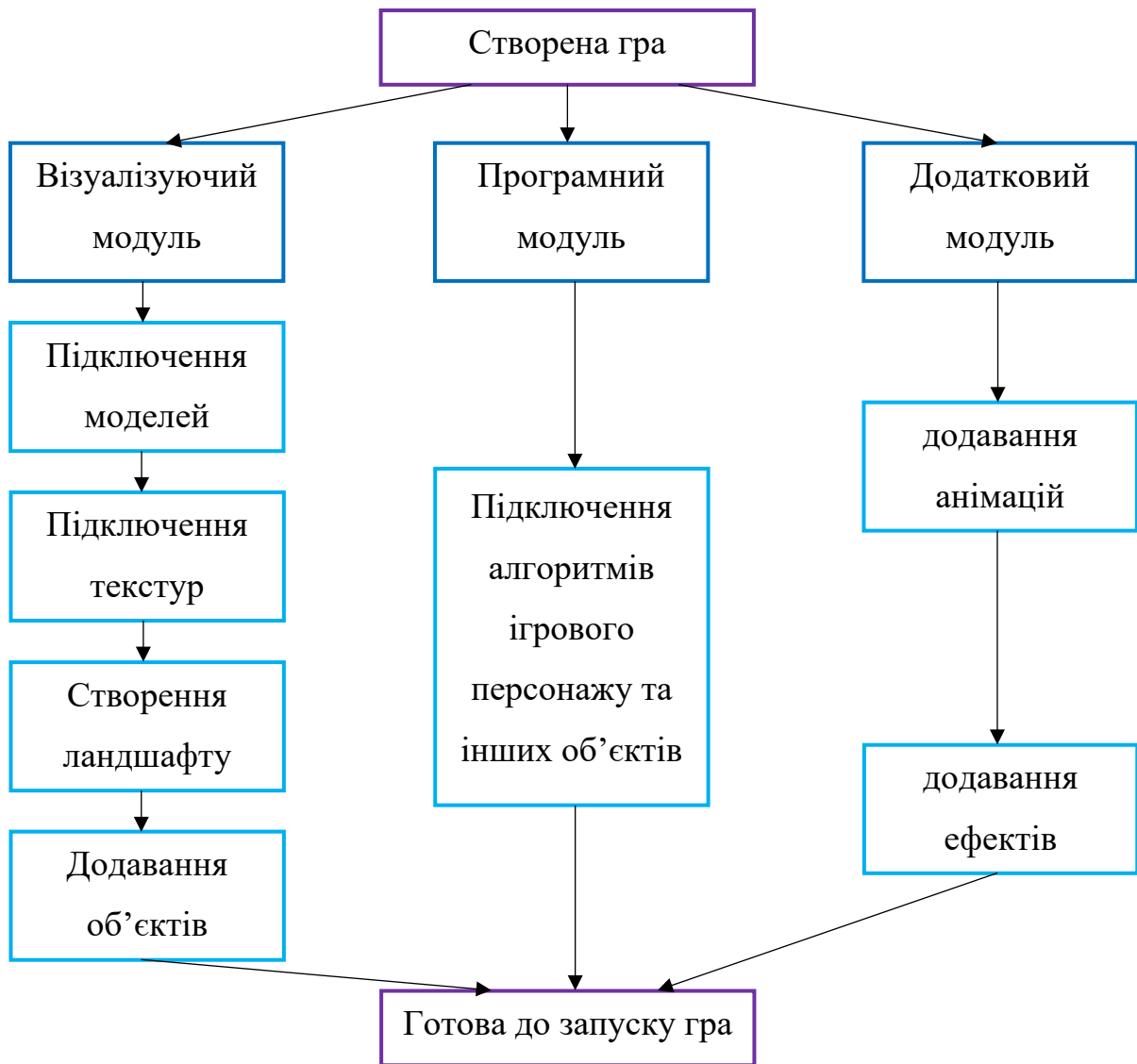


Рис. 2.34. Структура праці застосунку – гри

В даній кваліфікаційній роботі для надання алгоритмів функціонування використовувалися як Blueprints так і програмування на C++.

Blueprints – це система візуального скриптування, яка представляє собою певні блоки вузлів, котрі можна міняти, активувати, переміщати, видозмінювати, утворюючи при цьому цілі мініпрограми.

У вигляді Blueprints була створена можливість стрибку. Приклад цього можна побачити на рис. 2.35.

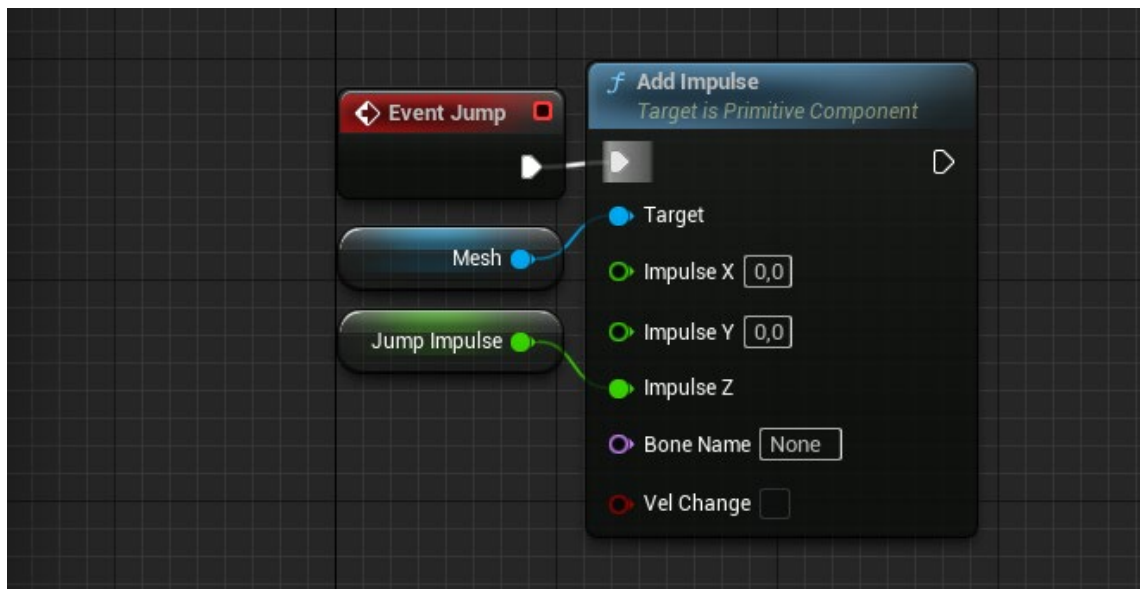


Рис. 2.35. Blueprint, що відповідає за стрибок об'єкту «сфера»

Також була створена анімація зникнення монети при натисканні на неї ігровим персонажем. Те, як воно виглядає, було наведено на рис. 2.36.

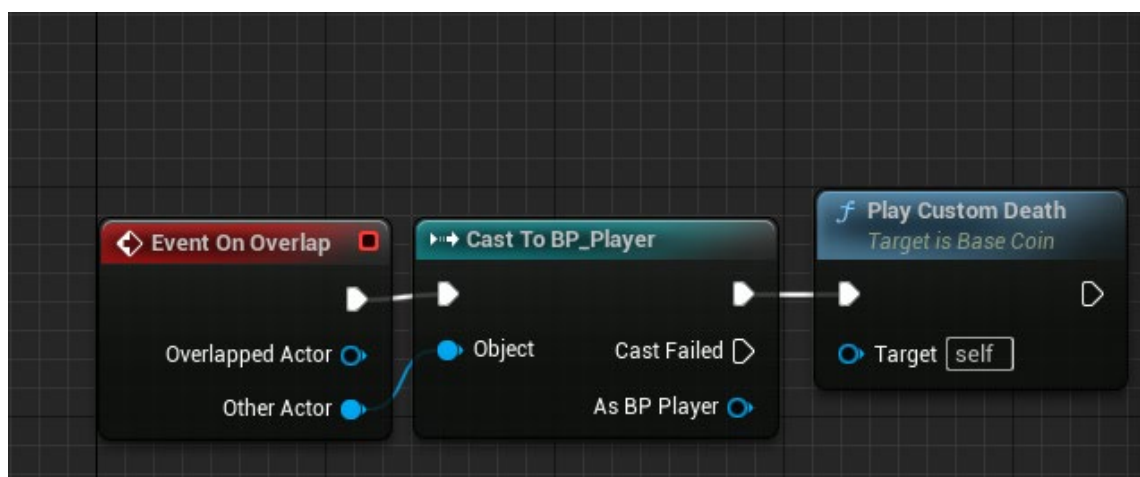


Рис. 2.36. Blueprint, що відповідає за анімацію зникнення монети

Усі інші алгоритми функціонування(як, наприклад, той же рух ігрового персонажа) були наведені у лістингу програми у вигляді коду C++.

2.5. Обґрунтування та організація вхідних та вихідних даних програми

Організація вхідних та вихідних даних влаштована за принципом запитування у користувача певних ключів, котрі програма зчитує та обробляє, після чого надає свою відповідь у вигляді відтворення на екрані якоїсь події(стрибок об'єкту «сфера») чи явища(зникнення об'єкту «призова монета» при натисканні на неї персонажем). Ключами виступають клавіші на клавіатурі.

Для того, щоб застосунок виконав певну конкретну команду можна скористатися довідкою, в котрій зібрані усі ключі, що використовуються програмою.

2.6. Опис розробленої системи

2.6.1. Використані технічні засоби

КР виконана на персональному комп'ютері власної збірки. Характеристики цього ПК наведені у таблиці 2.4.

Таблиця 2.4.

Характеристики персонального комп'ютера

Характеристика	Опис
Процесор	Intel Core i7-7740X
Частота процесора	4.3 GHz
Кількість ядер процесора	4 ядра
Операційна система	Windows 10 Enterprise
Розрядність системи	64-розрядна операційна система
Розмір оперативної пам'яті	16 GB
Модель відеокарти	NVIDIA GeForce GTX 970
Об'єм відеопам'яті	4 GB
Тип пам'яті	GDDR5

Характеристик даного ПК було більш ніж достатньо створення застосунку.

2.6.2. Використані програмні засоби

При розробці програмного забезпечення були використано наступні програмні засоби:

- Visual Studio 2019;
- Autodesk 3ds MAX;
- Adobe Photoshop 2021;
- Unreal Engine 5.

2.6.3. Виклик та завантаження програми

Для того, щоб виконати запуск застосунку достатньо у редакторі ігрового двигуна натиснути на клавішу «Play» на панелі зверху, як на рис. 2.37.

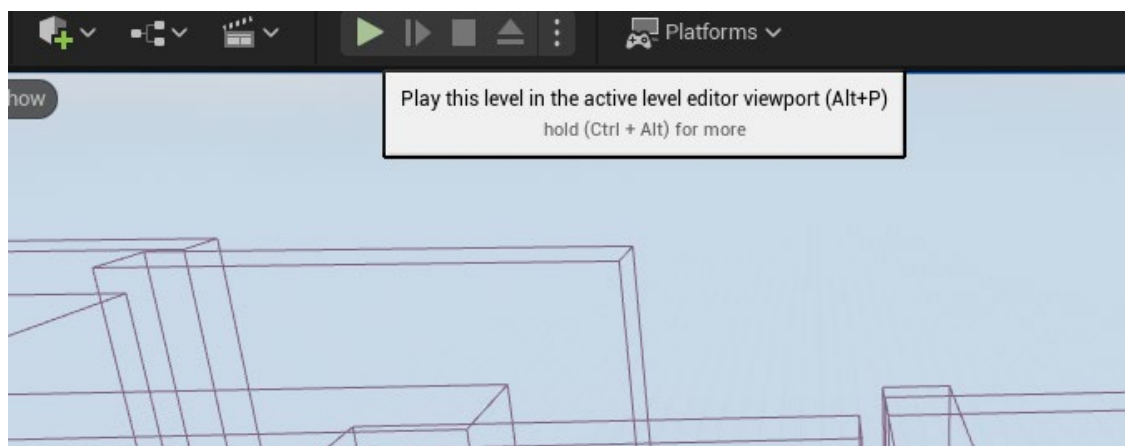


Рис. 2.37. клавіша «Play»

Після цього двигун запустить прототип гри у вікні всередині, в якому розробник може провести тестування свого продукту на рівні звичайного користувача. Приклад наведено на рис. 2.38.

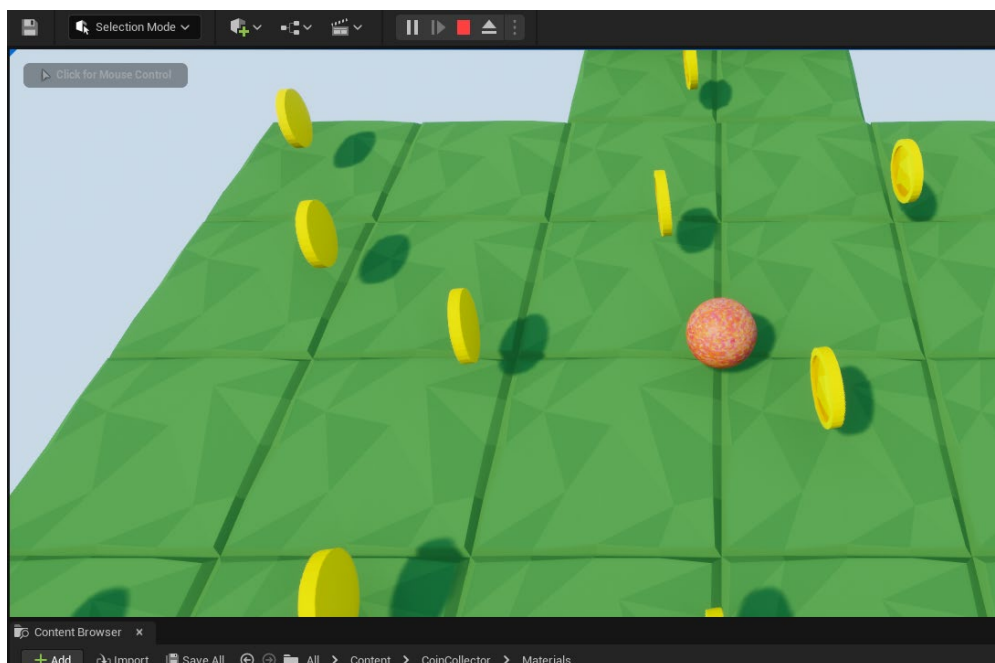


Рис. 2.38. Прототип гри у редакторі Unreal Engine

Вході цієї роботи також були проведені тестування, в яких було перевірено:

- переміщення персонажем;
- стрибок вгору об'єктом «сфера»;
- поведінка при зіткненні з ландшафтом ігрового рівня;
- анімація зникнення об'єкту «призова монета»;
- надійність захисту від падіння за рівень гри.

Всі пункти тестування були виконанні та визначені, як вдалі, тому подальшого тестування не потребують.

2.6.4. Опис інтерфейсу користувача

В якості інтерфейсу користувача був обраний тип Realistic(реалістичний), тобто коли на екрані майже немає чи взагалі відсутній інтерфейс. На рис. 2.39 наведено приклад вигляду цього інтерфейсу.

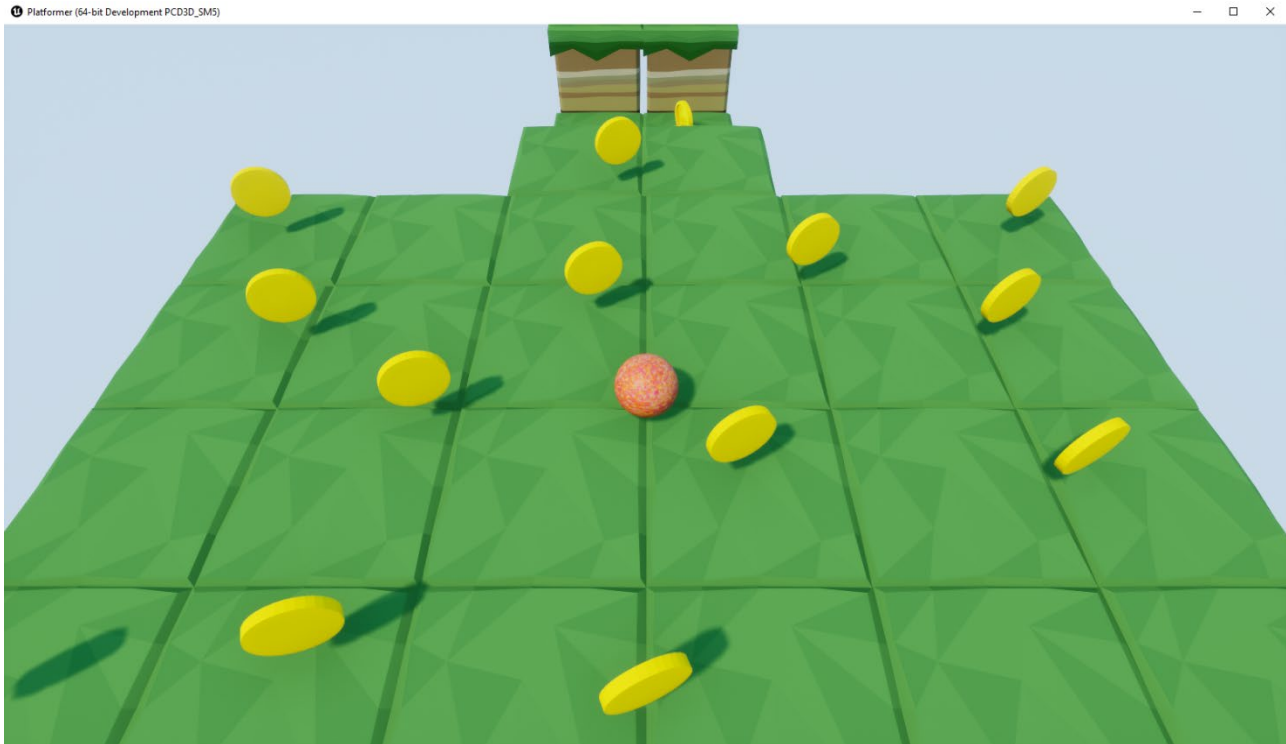


Рис. 2.39. Інтерфейс користувача

Причин обрати саме цей тип було декілька:

- простота – цей інтерфейс доволі зрозумілий;
- у нього відсутня «хвороба» багатьох комп'ютерних ігор з високим бюджетом – перенасичене інформацією зображення, в якому користувач не має змоги зрозуміти що-небудь;
- відчуття – він надає змогу поринути у віртуальний світ гри на більш глибокому рівні;
- дешевизна – вартість створення цього інтерфейсу є найменшою з усіх інших.

РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ

3.1. Розрахунок трудомісткості та вартості розробки програмного продукту

Початкові дані, які потрібні для подальшого розрахунку:

- передбачуване число операторів програми – 138;
- коефіцієнт складності програми – 1,4 ;
- коефіцієнт корекції програми в ході її розробки – 0,3;
- сер. год. зарплатня програміста с++ – 350 грн/год;
- коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі – 1,2;
- коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності –1,1;
- вартість машино-години ЕОМ – 25 грн/год.

Заробітна плата представників ІТ-департаменту залежить від багатьох факторів: складність постановленої задачі, попит на мови програмування, на спеціалізацію, час на виконання проекту, степінь довіри, яку надає фахівцеві замовник. У деяких випадках початківець рівня Junior може заробити більше, ніж кваліфіковані розробники С++ рівня Middle, маючи певний досвід і знання. Такі мови, як С, С++ та С# нараховують за собою вже десятки років свого функціонування(мова С ще з 1973р. існує), і при цьому вони не втрачають своєї актуальності. Працівник рівня Middle отримує в середньому 2000 доларів., що при переведені у гривні буде дорівнювати майже 74 тисячам гривень за сьогоднішнім курсом»

Весь процес розробки застосунку можна описати, як: розробка Beta версії моделі, виявлення всіх помилок, які не були виправлені до цього, шліфування продукту, щоб він відповідав усім необхідним вимогам, котрі накопичилися за

весь час створення застосунку та безпосередньо кінцева фаза, в якій замовнику надають вже повністю готовий застосунок.

Далі будуть розглянуті витрати, котрі з'явилися в ході розробки ігри – застосунку з використанням 3D–моделей.

Обладнання. Можна використовувати як і новий так і б/у комп'ютер, як мати своє, так і орендувати на певний час у компанії. За знайденою інформацією, якщо припустити, що комп'ютер використовувався протягом чотирьох років, а комплектуючі не оновлювалися, то щомісячна амортизація обладнання буде становити майже 300 грн.

Інтернет. Можна обрати одного з безлічі провайдерів, підключити різноманітні тарифи, кожен з яких надає різну швидкість інтернету. При пошуку середньої вартості інтернету, можна знайти, що в Україні ціна на інтернет становить 160 грн на місяць.

Електроенергія. Можна економити електроенергію різними способами, регулюючи яскравість екрану монітору та перемикаючи ПК у режим спокою, коли він не працює. При пошуку того, скільки потребляє енергії ПК, можна отримати таку інформацію: якщо комп'ютер споживає приблизно 250 Вт/год., і при цьому він працює 8 годин на добу, то споживання цього ПК за місяць буде становити 158 грн.

Таким чином, мінімальні щомісячні витрати на обладнання, електроенергію та інтернет для створення програмного забезпечення становлять 618 грн.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{deb} + t_d, \text{ людино-годин,} \quad (3.1)$$

де t_o - витрати праці на підготовку й опис поставленої задачі (приймається 40 людино-годин);

t_u – витрати праці на дослідження алгоритму рішення задачі;

t_a – витрати праці на розробку блок-схеми алгоритму;

t_n – витрати праці на програмування по готовій блок-схемі;

$t_{отл}$ – витрати праці на налагодження програми на ЕОМ;

t_d – витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у програмному забезпеченні, яке розробляється.

Умовне число операторів (підпрограм):

$$Q = q * C * (1 + p) ,$$

де q – передбачуване число операторів;

C – коефіцієнт складності програми;

p – коефіцієнт кореляції програми в ході її розробки.

Отримаємо:

$$Q = 138 * 1.4 * (1 + 0.3) = 256.2$$

Витрати праці на вивчення опису задачі t_u визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q * B}{(75..85) * k} , \text{ ЛЮД-ГОД,}$$

де B – коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі;

k – коефіцієнт кваліфікації програміста, обумовлений від стажу роботи зданої спеціальності.

Отримаємо:

$$t_u = \frac{256.2 * 1.2}{76 * 1.1} = 3.7 \text{ ЛЮД-ГОД.}$$

Витрати праці на розробку алгоритму рішення задачі:

$$t_a = \frac{Q}{(20..25) * k} , \text{ ЛЮД-ГОД.}$$

Отримаємо:

$$t_a = \frac{256.2}{21 * 1.1} = 11.1 \text{ ЛЮД-ГОД.}$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20..25) * k} , \text{ ЛЮД-ГОД.}$$

Отримаємо:

$$t_n = \frac{256.2}{24 \cdot 1.1} = 9.7 \text{ люд-год.}$$

Витрати праці на налагодження програми на ЕОМ:

- за умови автономного налагодження одного завдання:

$$t_{отл} = \frac{Q}{(4..5) \cdot k}, \text{ люд-год.}$$

Отримаємо:

$$t_{отл} = \frac{256.2}{4.6 \cdot 1.1} = 50.6 \text{ люд-год}$$

- за умови комплексного налагодження завдання

$$t_{отл}^k = 1.5 * t_{отл}, \text{ люд-год.}$$

Отримаємо:

$$t_{отл}^k = 1.5 * 50.6 = 75.9 \text{ люд-год.}$$

Витрати праці на підготовку документації:

$$t_d = t_{др} + t_{до}, \text{ люд-год,}$$

де $t_{др}$ - трудомісткість підготовки матеріалів і рукопису.

$$t_{др} = \frac{Q}{(15..20) \cdot k}, \text{ люд-год,}$$

Підрахуємо:

$$t_{др} = \frac{256.2}{17 \cdot 1.1} = 18.7 \text{ люд-год.}$$

$t_{до}$ - трудомісткість редагування, печатки й оформлення документації

$$t_{до} = 0.75 * t_{др}, \text{ люд-год.}$$

Підрахуємо:

$$t_{до} = 0.75 * 18.7 = 14.0 \text{ люд-год.}$$

Отже витрати праці на підготовку документації:

$$t_d = 18.7 + 14.0 = 32.7 \text{ люд-год.}$$

Повертаючись до формули (3.1) ми отримуємо:

$$t = 40 + 3.7 + 11.1 + 9.7 + 50.6 + 32.7 = 147.8 \text{ люд-год.}$$

3.2. Рахунок витрат на створення програми

Витрати на створення ПЗ ($K_{\text{ПО}}$) включають витрати на заробітну плату виконавця програми ($z_{\text{зп}}$) і витрат машинного часу ($z_{\text{мв}}$), необхідного на налагодження програми на ЕОМ

$$K_{\text{ПО}} = z_{\text{зп}} + z_{\text{мв}}, \text{ грн} \quad (3.2.1)$$

Заробітна плата виконавців визначається за формулою:

$$z_{\text{зп}} = t * C_{\text{пр}}, \text{ грн},$$

де t - загальна трудомісткість, людино-годин;

$C_{\text{пр}}$ – середня годинна заробітна плата програміста, грн/год.

Підрахуємо:

$$z_{\text{зп}} = 147.8 * 350 = 51730 \text{ грн}$$

Вартість машинного часу, необхідного для налагодження програми на ЕОМ:

$$z_{\text{мв}} = t_{\text{отл}} * V_{\text{мг}}, \text{ грн},$$

де $t_{\text{отл}}$ - трудомісткість налагодження програми на ЕОМ, год;

$V_{\text{мг}}$ - вартість машино-години ЕОМ, грн/год.

Підрахуємо:

$$z_{\text{мв}} = 50.6 * 25 = 1265 \text{ грн}$$

Отже повертаючись до формули (3.2.1) ми отримаємо :

$$K_{\text{ПО}} = 51730 + 1265 = 52995 \text{ грн}$$

Визначені в такий спосіб витрати на створення програмного забезпечення є частиною одноразових капітальних витрат на створення АСУП.

$$T = \frac{t}{V_{\text{к}} * F_{\text{р}}}, \text{ міс},$$

де $V_{\text{к}}$ - число виконавців ($V_{\text{к}} = 1$);

$F_{\text{р}}$ - місячний фонд робочого часу (при 40 годинному робочому тижні $F_{\text{р}} = 171$ год за місяць).

Отримаємо:

$$T = \frac{147.8}{1 * 171} = 0.86 \text{ міс}$$

Округлимо число T у більшу сторону:

$$T = 0.9 \text{ міс}$$

Таким чином було розраховано наступні показники:

- трудомісткість розробки ПЗ – 147,8 люд-год;
- витрати на створення 52 995 – грн;
- очікуваний період створення ПЗ – 0,9 міс.

ВИСНОВОК

З кожним роком сфера 3D-індустрії дедалі більше зростає, що говорить про дуже сильну зацікавленість масового користувача в ній, а це вже надає актуальності розробленому проекту. Зараз популярні різні проекти, в яких використовують 3D технології: кіноіндустрія, індустрія анімації, індустрія відеоігор, телебачення, різноманітні виробництва об'ємних деталей, 3D-друк.

Метою кваліфікаційної роботи є створення 3D-моделі та її впровадження у Unreal Engine за допомогою мови програмування C++.

Розроблений програмний додаток вміє:

- впроваджувати 3D-моделі до двигуна Unreal Engine;
- накладати на ці моделі текстури;
- відтворювати карту рівня;
- надавати об'єктам анімацій та ефектів;
- надавати ігровому персонажеві елементів управління.

Для досягнення поставлених цілей вивчена література щодо створення повноцінних 3D-моделей та текстур, розглянуті різноманітні сайти для впровадження цих елементів до Unreal Engine, їх взаємодія з ігровим двигуном та можливості, котрі він надає.

В результаті виконання роботи, створено застосунок, який надає користувачеві під керування ігрового персонажа, котрим можливо переміщатися на рівні гри, стрибати вгору і вверх, збирати літаючі монети, кожна з яких вміє ефектно зникати при зіткненні з нею.

Працездатність даного програмного продукту підтверджується вдалими експлуатаційними випробуваннями.

В економічному розділі визначено трудомісткість розробленої інформаційної системи (147,8 люд-год), проведений підрахунок вартості роботи по створенню програми (52995 грн.) та розраховано час на його створення (0,9 міс).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. 3D моделювання в промисловій сфері / навч.журн./ К. Ю. Маслів, М. Ю. Похорукова; 2016. - 232 с.
2. Practical Autodesk AutoCAD 2021 and AutoCAD LT 2021: A nonsense, beginner's guide to drafting and 3D modeling with Autodesk AutoCAD / science book / Yasser Shoukry, Jaiprakash Pandey; 2020. – 826 p.
3. Autodesk 3ds Max 2020: A Detailed Guide to Modeling, Texturing, Lighting, and Rendering, 2nd Edition / science book / Pradeep Mamgain; 2019. – 648 p.
4. Adobe Photoshop Elements 2023: A Guide For Beginners And Seniors To Master Photoshop Element Techniques, Features, And Tools, With Tutorials To Professionally Organize, Edit, And Enhance Images / science book / George V. Todd; 2023. – 141 p.
5. 3D-ландшафти у реальному часі на C++ і DirectX 9. - 2-ге вид / навч.книг. / Г. Снук; 2007. - 368 с.
6. Unreal Engine 5 RPG Development with C++ and Blueprint: Volume I: Basics, Combat, and VFX / science book / Dr. Chihming Chiu; 2022. – 546 p.
7. Unreal Engine 5 Character Creation, Animation, and Cinematics: Create custom 3D assets and bring them to life in Unreal Engine 5 using MetaHuman, Lumen, and Nanite / science book / Henk Venter, Wilhelm Ogterop; 2022, – 608 p.
8. Game Development with Unreal Engine 5: Learn the Basics of Game Development in Unreal Engine 5 / science book / Mitchell Lynn, Cliff Sharif; 2022. – 421 p.
9. 3D Game Textures: Create Professional Game Art Using Photoshop / science book / Luke Ahearn; 2016. – 412 p.
10. Іванов, А. С. Батраков. Тривимірна комп'ютерна графіка / Под ред. Поліщука, 1995. - 224 с.
11. A Practical Guide to Level Design: From Theory to Practice, Diplomacy and Production / science book / Benjamin Bauer; 2023. – 374 p.
12. What is 3D Modeling & What's It Used For? / URL: <https://conceptartempire.com/what-is-3d-modeling/>. дата звернення: 12.04.2023.
13. 3D Max Models / URL: <https://www.educba.com/3d-max-models/>. дата звернення: 12.04.2023.
14. How to Use Photoshop: The Bookmarkable Photoshop Tutorial for Beginners / URL: <https://blog.hubspot.com/marketing/how-to-use-photoshop>. дата звернення: 13.04.2023.
15. How to Use Adobe Photoshop / URL: <https://www.wikihow.com/Use-Adobe-Photoshop>. дата звернення: 13.04.2023.
16. How to Create Textures in Photoshop CC / URL: <https://www.makeuseof.com/tag/create-textures-photoshop/>. дата звернення: 14.04.2023.

17.10 Different types of 3D modeling techniques / URL: <https://artisticrender.com/10-different-types-of-3d-modeling-techniques/>. дата звернення: 14.04.2023.

18. Comparison of 3D computer graphics software / URL: https://en.wikipedia.org/wiki/Comparison_of_3D_computer_graphics_software. дата звернення: 15.04.2023.

19. Unreal Engine 5 Tutorial for Beginners: Getting Started / URL: <https://www.kodeco.com/31800833-unreal-engine-5-tutorial-for-beginners-getting-started>. дата звернення: 15.04.2023.

20. Understanding the Basics / URL: <https://docs.unrealengine.com/5.0/en-US/understanding-the-basics-of-unreal-engine/>. дата звернення: 16.04.2023.

21. The Best Gaming Engines You Should Consider for 2023 / URL: <https://www.incredibuild.com/blog/top-gaming-engines-you-should-consider>. дата звернення: 16.04.2023

ЛІСТИНГ ПРОГРАМИ

BaseCoin.cpp

```
#include "BaseCoin.h"
#include "BasePlayer.h"

ABaseCoin::ABaseCoin()
{
    PrimaryActorTick.bCanEverTick = true;

    Root = CreateDefaultSubobject<USceneComponent>("Root");
    RootComponent = Root;
    CoinMesh = CreateDefaultSubobject<UStaticMeshComponent>("CoinMesh");
    CoinMesh->SetupAttachment(Root);
    CoinMesh->SetCollisionResponseToAllChannels(ECollisionResponse::ECR_Overlap);
    RotationRate = 100;

    OnActorBeginOverlap.AddDynamic(this, &ABaseCoin::OnOverlap);
}

void ABaseCoin::BeginPlay()
{
    Super::BeginPlay();
}

void ABaseCoin::Tick(float DeltaTime)
{
    AddActorLocalRotation(FRotator(0, RotationRate * DeltaTime, 0));
}

void ABaseCoin::DeathTimerComplete()
{
    Destroy();
}

void ABaseCoin::PlayCustomDeath()
{
    RotationRate = 1500;
    GetWorldTimerManager().SetTimer(DeathTimerHandle, this,
&ABaseCoin::DeathTimerComplete, 0.5f, false);
}

void ABaseCoin::OnOverlap_Implementation(AActor* OverlappedActor, AActor* OtherActor)
{
    if (Cast<ABasePlayer>(OtherActor) != nullptr)
    {
        Destroy();
    }
}
```

BaseCoin.h

```
#pragma once

#include "CoreMinimal.h"
#include "GameFramework/Actor.h"
#include "Components/StaticMeshComponent.h"
#include "BaseCoin.generated.h"

UCLASS()
class PLATFORMER_API ABaseCoin : public AActor
{
    GENERATED_BODY()

public:
    ABaseCoin();

protected:

    virtual void BeginPlay() override;

    FTimerHandle DeathTimerHandle;

    void DeathTimerComplete();

public:

    virtual void Tick(float DeltaTime) override;

    UPROPERTY(VisibleAnywhere, BlueprintReadOnly)
    USceneComponent* Root;

    UPROPERTY(VisibleAnywhere, BlueprintReadOnly)
    UStaticMeshComponent* CoinMesh;

    UPROPERTY(EditAnywhere, BlueprintReadWrite)
    float RotationRate;

    UFUNCTION(BlueprintCallable)
    void PlayCustomDeath();

    UFUNCTION(BlueprintNativeEvent)
    void OnOverlap(AActor* OverlappedActor, AActor* OtherActor);
};
```


BasePlayer.cpp

```
#include "BasePlayer.h"

ABasePlayer::ABasePlayer()
{
    PrimaryActorTick.bCanEverTick = true;

    Mesh = CreateDefaultSubobject<UStaticMeshComponent>("Mesh");
    SpringArm = CreateDefaultSubobject<USpringArmComponent>("SpringArm");
    Camera = CreateDefaultSubobject<UCameraComponent>("Camera");

    RootComponent = Mesh;
    SpringArm->SetupAttachment(Mesh);
    Camera->SetupAttachment(SpringArm);

    Mesh->SetSimulatePhysics(true);
    MovementForce = 100000;
}

void ABasePlayer::BeginPlay()
{
    Super::BeginPlay();
}

void ABasePlayer::Tick(float DeltaTime)
{
    Super::Tick(DeltaTime);
}

void ABasePlayer::SetupPlayerInputComponent(UInputComponent* PlayerInputComponent)
{
    Super::SetupPlayerInputComponent(PlayerInputComponent);
    InputComponent->BindAxis("MoveUp", this, &ABasePlayer::MoveUp);
    InputComponent->BindAxis("MoveRight", this, &ABasePlayer::MoveRight);
    InputComponent->BindAction("Jump", IE_Pressed, this, &ABasePlayer::Jump);
}

void ABasePlayer::MoveUp(float Value)
{
    FVector ForceToAdd = FVector(1, 0, 0) * MovementForce * Value;
    Mesh->AddForce(ForceToAdd);
}

void ABasePlayer::MoveRight(float Value)
{
    FVector ForceToAdd = FVector(0, 1, 0) * MovementForce * Value;
    Mesh->AddForce(ForceToAdd);
}
```

BasePlayer.h

```
#pragma once

#include "CoreMinimal.h"
#include "GameFramework/Pawn.h"
#include "Components/StaticMeshComponent.h"
#include "GameFramework/SpringArmComponent.h"
#include "Camera/CameraComponent.h"
#include "BasePlayer.generated.h"

UCLASS()
class PLATFORMER_API ABasePlayer : public APawn
{
    GENERATED_BODY()

public:
    ABasePlayer();

protected:
    virtual void BeginPlay() override;

public:
    virtual void Tick(float DeltaTime) override;

    virtual void SetupPlayerInputComponent(class UInputComponent* PlayerInputComponent)
    override;

    UPROPERTY(VisibleAnywhere, BlueprintReadOnly)
    UStaticMeshComponent* Mesh;

    UPROPERTY(VisibleAnywhere, BlueprintReadOnly)
    USpringArmComponent* SpringArm;

    UPROPERTY(VisibleAnywhere, BlueprintReadOnly)
    UCameraComponent* Camera;

    UPROPERTY(EditAnywhere, BlueprintReadWrite)
    float MovementForce;

    void MoveUp(float Value);
    void MoveRight(float Value);

    UPROPERTY(EditAnywhere, BlueprintReadWrite)
    float JumpImpulse;

    UFUNCTION(BlueprintImplementableEvent)
    void Jump();
};
```

Platformer.cpp

```
#include "Platformer.h"  
#include "Modules/ModuleManager.h"  
  
IMPLEMENT_PRIMARY_GAME_MODULE( FDefaultGameModuleImpl, Platformer, "Platformer" );
```

Platformer.h

```
#pragma once  
  
#include "CoreMinimal.h"
```

ДОДАТОК Б.

ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ

ПЕРЕЛІК ДОКУМЕНТІВ НА ОПТИЧНОМУ НОСІЇ

Ім'я файлу	Опис
Пояснювальні документи	
ПЗ_Пелін.docx	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
ПЗ_Пелін.pdf	Пояснювальна записка до кваліфікаційної роботи в форматі PDF
Програма	
Program.rar	Архів. Містить коди програми і откомпільовану програму
Презентація	
Презентація_Пелін.ppt	Презентація кваліфікаційної роботи