

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Теплоенергетичний факультет**

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено:

Завідувач кафедри

_____ Наталія АУШЕВА

«___» _____ 2022 р.

Дипломна робота

на здобуття ступеня бакалавра

спеціальності 122 «Комп'ютерні науки»

освітня програма «Комп'ютерний моніторинг та геометричне
моделювання процесів і систем»

на тему: «3D екскурсія в корпусі №5 КПІ ім. Ігоря Сікорського»

Виконала:

Студентка ІV курсу, групи ТР-81

Іванюк Анна Олегівна _____

Керівник:

Доктор технічних наук, професор,

Бадаєв Юрій Іванович _____

Рецензент:

Кандидат технічних наук, доцент,

Рачинський Артур Юрійович _____

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студентка _____

Київ – 2022

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший

Спеціальність 122 “Комп’ютерні науки”

Освітня програма “Комп’ютерний моніторинг та геометричне моделювання процесів та систем”

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Наталія АУШЕВА

(підпис)

” ___ ” _____ 2022 р.

ЗАВДАННЯ

на дипломну роботу студенту

Іванюк Анні Олегівні

(прізвище, ім’я, по батькові)

1. Тема роботи “3D екскурсія в корпусі №5 КПІ ім. Ігоря Сікорського”

керівник роботи Бадаєв Юрій Іванович, професор, доктор технічних наук

(прізвище, ім’я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від ” 8 ” червня 2022 р.

№ 965-с

2. Строк подання студентом роботи 10 червня 2022 р.

3. Вихідні дані до роботи персональний комп’ютер під керуванням операційної системи Microsoft Windows.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) проаналізувати існуючі програми та доцільність їх використання, обґрунтувати обрані програмні застосунки та шляхи розробки моделей, розробка проекту, зробити висновки за результатами роботи

5. Перелік ілюстраційного матеріалу:

1. Моделювання об’єктів для сцени 2. Subdivision surface моделювання.Булеве

моделювання 3. Solid та Shell моделі 4. Створення матеріалу для об'єкту. 5. Імпорт об'єктів до ігрового рушія
6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

Дата видачі завдання ” 10 ” вересня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Затвердження теми роботи	02.05.22	
2.	Вивчення та аналіз задачі	03.05.22	
3.	Розробка загальної структури проекту	03.05.22-05.05.22	
4.	Моделювання об'єктів	06.05.22-15.05.22	
5.	Текстурування об'єктів та інтегрування їх в ігровий рушій. Тестування системи.	16.05.22-22.05.22	
6.	Оформлення пояснювальної записки	23.05.22-28.05.22	
7.	Захист програмного продукту	29.05.22-30.05.22	
8.	Передзахист	07.06.22	
9.	Захист	20.06.22-30.06.22	

Студентка

_____ (підпис)

Іванюк А. О.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Бадаєв Ю. І.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Метою даної роботи було відтворення навчального корпусу №5 НТУУ “КПІ ім. Ігоря Сікорського” за допомогою 3D технологій, для абітурієнтів, студентів та викладачів. Актуальність даної роботи є високою, адже після тривалого дистанційного навчання та надзвичайних ситуацій в країні, важко знову адаптуватись до нових умов навчального процесу. Робота допоможе зорієнтуватись та допомогти знайти місце проведення занять.

Для моделювання усіх об'єктів використовувався застосунок Maya від компанії Autodesk. Робота зроблена на базі ігрового рушія Unreal Engine від компанії Epic Games. Для текстурування об'єктів в сцені та створення UV-розгортки для них використовувалась програма 3D-Coat.

Записка містить: 46 сторінки, 27 рисунків, 2 таблиці, 1 додаток і 15 посилань.

Ключові слова: 3D технології, моделювання об'єктів, ігровий рушій, текстури.

ANNOTATION

The purpose of this work was the creation of the building No. 5 of NTUU “KPI im. Igor Sikorsky” with a help of 3D technologies, for applicants, students and candidates. The urgency of this work is high, because after the prolonged remote and due to extraordinary situations in the country, it is important to re-adapt to the new conditions of the educational process. This work will help you to get to the right place in the building.

Maya services from Autodesk were used to model all objects. The work is based on the game engine Unreal Engine from Epic Games. 3D-Coat was used to texture objects in the scene and create UV scans for them.

The note contains: 46 pages, 27 pictures, 2 tables, 1 appendix and 15 links.

Keywords: 3D technologies, object modeling, game engine, texture.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	7
ВСТУП.....	11
1 ПОСТАНОВКА ЗАДАЧІ СТВОРЕННЯ 3D ЕКСКУРСІЇ.....	12
2 ОПИС ІСНУЮЧИХ РІШЕНЬ ДЛЯ 3D ЕКСКУРСІЙ.....	13
3 ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	15
3.1 Програмне забезпечення для 3D моделювання об'єктів	18
3.2 Програмне забезпечення для UV-mapping, та текстурування об'єктів	19
3.3 Вибір ігрового рушія	19
4 МОДЕЛЮВАННЯ ОБ'ЄКТІВ	21
4.1 Техніки моделювання	21
4.2 Організація робочого процесу	25
4.3 Робота в редакторі для моделювання об'єктів Autodesk Maya	27
4.4 Створення UV розгортки об'єктів та текстурування	33
5 СТВОРЕННЯ ТА НАЛАШТУВАННЯ ПРОЄКТУ	36
5.1 Імпорт об'єктів	36
5.2 Розміщення об'єктів.....	40
6 ІНФОРМАЦІЯ КОРИСТУВАЧЕВІ 3D ЕКСКУРСІЄЮ	43
ВИСНОВКИ.....	44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	45
ДОДАТОК А.....	47

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Блокінг – створення силуету моделі, крупних форм без деталізації. На цьому етапі візуалізатор лише починає розробку композиції.

Пайплайн – цикл розробки 3D моделі.

API (Application Programming Interface) – опис способів за допомогою яких, програми можуть взаємодіяти між собою. API спрощує процес програмування, надаючи розробнику лише об'єкти або дії, котрі йому необхідні.

UV mapping [9] – UV-перетворення, яку ще також називають розгорткою, у моделюванні – це коли координати на тривимірному об'єкті (X, Y, Z) відповідають координатам, що на текстурі (U, V). Значення U та V можуть змінюватись від 0 до 1. Розгортка будується вручну або автоматично – наприклад, у 3Ds Studio MAX є кілька алгоритмів автоматичного розгортання моделі.

Ретопологія – буквально означає зробити топологію ще раз (знову, з самого початку). Зазвичай ретопологію роблять на основі високополігональної (hi-poly) 3D моделі, й саме цю модель вже будуть використовувати у проєкті.

Авторетопологія – автоматична ретопологія для об'єкту.

Автомапінг – автоматичне створення розгортки.

Ігровий рушій – програмне забезпечення, котре створене для повторного розробки відеоігор, і є основою для розробки та запуску різних форматів ігор.

Рендер [11] — це процес отримання зображення. Тут модель — це опис тривимірних об'єктів за допомогою мови програмування, котра подається у вигляді структури даних. Рендер містить в собі геометричні дані, положення точки спостерігача, освітлення. Зображення є цифровим та растровим зображенням.

Трасування усіх променів світла є досить великою витратою ресурсів. Інколи трасування навіть малої кількості променів займає надто багато часу, разом з цим чіткість зображення буде недостатньою. Крім того, щоб отримати більш якісне зображення, необхідно використовувати більшу кількість променів.

Загалом є такі методи:

Растрезація (англ. rasterization) візуалізація усієї сцени проводиться проєкціюванням об'єктів на екран без розгляду ефекту перспективи відносно спостерігача.

Ray casting (метод кидання променів) сцена спостерігається з певної точки, звідки направляються промені на об'єкти, та визначається колір пікселя на двовимірному екрані. Промені припиняють своє поширення одразу, як тільки досягають будь-якого об'єкта сцени або її фону. В цьому методі можливе використання будь-яких дуже простих способів додавання оптичних ефектів. Ефект перспективи отримується природним чином, якщо промені надходять під кутом, що залежить від положення пікселя на екрані і максимального кута об'єктиву камери.

Трасування променів (англ. ray tracing) цей метод є схожим до попереднього. З точки спостереження на об'єкти сцени направляються промені, далі визначається колір пікселя на двовимірному екрані. Але промінь не припиняє своє поширення, як у попередньому методі, а розділяється на три променя, кожен з яких вносить свій внесок в колір пікселя на двовимірному екрані: відбитий, тінювий і заломлений. Кількість таких поділів на компоненти визначає глибину трасування та впливає на якість і фотореалістичність зображення. Завдяки своїм концептуальним особливостям, метод дозволяє отримати дуже фотореалістичні зображення, але при цьому він дуже ресурсозатратний, і процес візуалізації займає значні періоди часу.

Трасування шляху (англ. path tracing) цей метод має схожий принцип трасування променів, що поширюються, проте він є найбільш наближеним до фізичних законів поширення світла. Відповідно, він є найбільш ресурсозатратним.

Новітнє програмне забезпечення часто має декілька технік, для того щоб отримати реалістичне та якісне зображення при допустимих витратах ресурсів.

Скрипт [8] – сценарій, прописана послідовність певних дій.

Штучний інтелект — можливість обробляти, застосовувати, а також вдосконалювати здобуті знання та вміння.

Колізія - суперечність, зіткнення протилежних сил, зіштовхнення.

Subdivision моделювання - модифікатор Subdivision Surface (часто скорочують до Subdiv або SubD) використовується для підрозділу граней меша на більш дрібні, додаючи згладженості. Він дозволяє створювати складні згладжені поверхні, не ускладнюючи геометрію.

NURBS моделювання [10] - (англ. Non-uniform rational B-spline) — (неоднорідний раціональний базовий сплайн) це математична модель, що використовується в комп'ютерній графіці для створення й представлення кривих або поверхностей

Baking (запікання) - це назва процесу збереження інформації, що стосується 3D-сітки, у файл текстури (растрове зображення). Найчастіше у цьому процесі задіяна інша сітка. Інформація першого меша переноситься на розгортку іншого. Після цього вона зберігається у текстуру.

Меш - це сукупність вершин, ребер та граней, за допомогою яких, створюється форма багатогранного об'єкта в 3D-графіці та об'ємному моделюванні.

Z-fight – (planefighting) явище у 3D, коли два або більше примітивів мають схожі відстані до камери. Призводить до того, що вони матимуть майже подібні або ідентичні значення в z-буфері, котрий відстежує глибину.

Текстура [7] — це властивості поверхні 3D моделі, а саме: колір, фактура, блиск, металевість, прозорість і т. д. . Текстура є дуже важливим елементом 3D-моделювання,

оскільки дозволяє відтворити також малі, незначні особливості поверхні, котрі відтворювати за допомогою полігонів було б занадто ресурсомістким.

Якість поверхні текстури визначається текселями — це кількість пікселів на мінімальну одиницю текстури. Так як текстура по суті являє собою зображення, роздільність і формат цього зображення впливає на загальне враження 3D-моделі.

Створення матеріалу та його налаштувань здійснюється, використовуючи карти відповідних каналів, або через процедурні карти, також можливий змішаний метод.

ВСТУП

3D технології все більше розповсюджуються та постійно з'являються в нашому житті. На сьогоднішній день дешевше змоделювати простір та об'єкти, ніж майструвати декорації для кіноіндустрії чи навколишнє середовище для реклами продукту. Технології дійшли до рівня, коли відрізнити фотографію від змодельованого об'єкту надзвичайно важко. Крім того, це значно оптимізує витрати та час на розробку проєктів.

Дана робота може бути розширена та доповнена, а також використовуватись в різних цілях, та мета дипломної роботи полягає в відтворенні навчального корпусу для легкого та швидкого знаходження потрібної аудиторії студентом чи викладачем, саме тому проєкт більше спрямований на те, щоб допомогти зорієнтуватись на місцевості. Робота була зроблена за допомогою наданих матеріалів, котрі були наявні на момент звернення до адміністрації корпусу № 5 НТУУ “КПІ ім. Ігоря Сікорського”.

Робота була зроблена за допомогою застосунку компанії Autodesk Maya, а саме з навчальним доступом до програмного забезпечення, саме тому, згідно з правилами компанії Autodesk, проєкт не може бути використаний у професійних чи комерційних цілях широкого призначення без придбання відповідної ліцензії. Навчальний доступ до використання програмного забезпечення був наданий завдяки НТУУ “КПІ ім. Ігоря Сікорського”.

1 ПОСТАНОВКА ЗАДАЧІ СТВОРЕННЯ 3D ЕКСКУРСІЇ

В інтернеті досить мала кількість фотографій теплоенергетичного корпусу, а функції прогулятися усіма поверхами будівлі немає, тому в абітурієнтів та студентів-першокурсників часто виникають проблеми з пошуком потрібних аудиторій. Через це, була поставлена задача створити 3D-простір корпусу №5. Модель корпусу може бути використана для різноманітних цілей, та в першу чергу вона буде використовуватись абітурієнтами чи студентами після тривалого дистанційного навчання.

Для цього необхідно проаналізувати декілька існуючих рішень, як вони зроблені, що можна покращити. Також потрібно дослідити доступне програмне забезпечення обрати найоптимальніше. Після цього необхідно змоделювати саму будівлі та її декор, затекстурувати, помістити в ігровий рушій.

В результаті виконання роботи, користувач може пройтись та побачити теплоенергетичний корпус №5 НТУУ ім. Ігоря Сікорського зсередини.

2 ОПИС ІСНУЮЧИХ РІШЕНЬ ДЛЯ 3D ЕКСКУРСІЙ

Поширене виконання 3D екскурсії полягає у тому, що користувач може переходити з місця на місце лише за допомогою спеціальних стрілок. Таке пересування виходить стрибками та може здійснюватись лише в окремих точках, тобто не по всій локації. Крім цього, багато реалізованих 3D екскурсій створені за допомогою фотографій, через це простір часто виглядає розмитим або з певною кількістю людей, які інколи заважають дослідити місцевість, та обличчя яких потрібно обов'язково маскувати.

Прикладом однієї з таких екскурсій є віртуальний 3D-тур по Київському театру оперети, котрий зображений на рисунку 2.1.

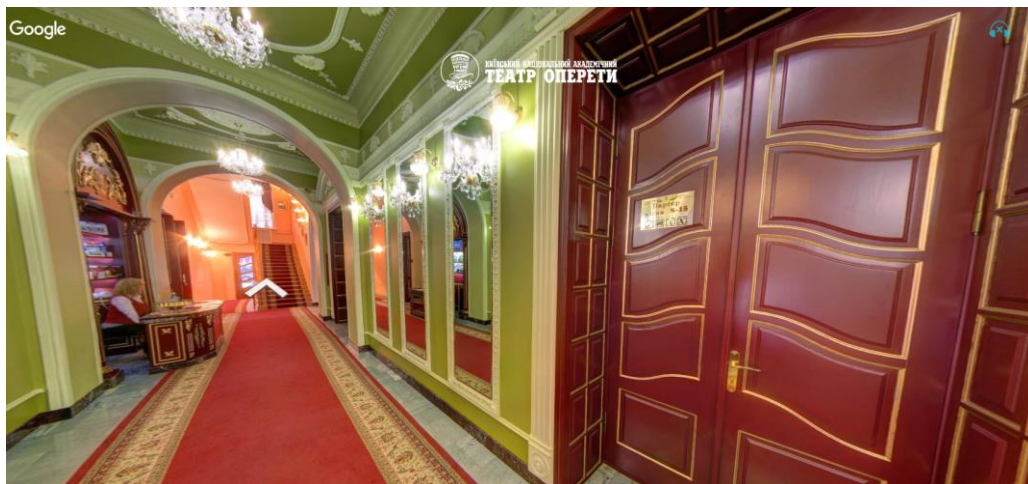


Рисунок 2.1 — 3D-тур по Київському театру оперети

Дане рішення було реалізоване саме завдяки фотографіям. Крім цього, швидкість такого 3D туру може здатись низькою, адже спочатку потрібно знайти стрілку (рухатись потрібне лише по визначеному маршруту), та, інколи, необхідно натискати

на багато стрілок в одному напрямку аби дійти до кінця коридору, а це займає час, бо кожного разу система завантажує нову фотографію.

Отже, основною перевагою даної роботи є вільне пересування по створеній локації. Також відсутнє скупчення людей, й робота створена завдяки 3D технологіям, тому у разі надзвичайного випадку може допомогти відновити корпус.

3 ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Вибір програмного забезпечення для розробки проєкту залежить від поставлених задач та подальших цілей. Більшість популярного програмного забезпечення підходить для моделювання даної роботи, та для зручності та економії часу, краще обрати програму з більш потрібними нам можливостями, що зображені на таблиці 3.1 та таблиці 3.2.

Таблиця 3.1 — Порівняння програмного забезпечення для моделювання об'єктів

Назва програми	Розробник	Платформи	Для чого використовуються	Ліцензія
Maya	Autodesk	Mac Os X, Microsoft Windows	Моделювання, моделювання NURBS Скульптура, такелаж, зняття шкіри, ретаргетинг, анімація, освітлення, візуалізація, Bifrost Fluids, Python Scripting, Bifrost Ocean Simulation System, UV Mapping, MASH процедурні ефекти, Редактор часу та графіків, засіб візуалізації Arnold, Візуальне моделювання	Власна

			полігонів з керуванням кольором, 3D ефекти	
--	--	--	--	--

Продовження таблиці 3.1

Blender	Blender Foundation	Mac Os X, Microsoft Windows	2D/3D мультфільм анімація, освітлення, моделювання, створення матеріалів на основі вузлів/текстурування/3D-малювання текстур/ультрафіолетове відображення, візуалізація (внутрішнє, зовнішнє, 3D-анагліф і VR), 3D-оснащення та анімація, ліплення, візуальні 3D-ефекти, основні пост- Виробниче редагування відео, відстеження руху, написання сценаріїв на Python, моделювання рідини, частинки, фізика, КОМПОЗИТИНГ	GPL-2.0-or-later
3dsMax	Autodesk	Mac Os X, Microsoft Windows	Моделювання, текстурування, освітлення, ліплення (деформація моделі), рендеринг	Власна

Кінець таблиці 3.1

Zbrush	Pixologic	Mac Os X, Microsoft Windows	Моделювання, текстурування, освітлення, ліплення (деформація моделі), рендеринг	Власна
--------	-----------	-----------------------------------	---	--------

Ліцензія типу GPL-2.0-or-later [12] – це серія загальних публічних ліцензій для безкоштовного користування програмним забезпеченням. Така ліцензія гарантує користувачеві запуск, вивчення, модифікацію та спільне використання програмного забезпечення.

Таблиця 3.2 — Порівняння програмного забезпечення для моделювання об’єктів

Назва програми	Hard Surface		Sculpt	Non-destructive modeling	
	polygon	subdivision		Boolean	Bevel
Maya	+	+	+	+	+
Blender	+	+	+	+	+
3dsMax	+	+	Paint deform	+	+
Zbrush	Zmodeler	Zmodeler	+	+	-

За допомогою порівняльного аналізу досліджуємо усі особливості програм та обираємо найбільш підходяжу.

3.1 Програмне забезпечення для 3D моделювання об'єктів

Для відбору необхідно спочатку зосередити увагу щодо вимог та поставлених цілей. Потрібно сконструювати досить багато об'єктів, тому використання специфічних інструментів може значно скоротити час. Оптимізувати процес також допоможе можливість експорту до сцени сторонніх об'єктів. Нам не потрібно робити великий обсяг анімації, всі об'єкти в сцені будуть статичними. Необхідно зазначити, що в нас відсутні точні розміри сцени, тому усі об'єкти будуть створюватись пропорційно, та остаточні розміри будуть визначені на етапі блокінгу. Також, для оптимізації робочого процесу, слід обирати програмне забезпечення, котре буде вмщати всі необхідні функції без додаткових плагінів. В проєкті не будуть використовуватись спеціальні ефекти.

Незважаючи на те, що інструментарій в усіх програмах подібного плану схожий, враховувався досвід попереднього використання програми.

Таким чином, стає зрозуміло, що використання програм, котрі мають розширені можливості для скульптурування, створення персонажів та анімації, немає потреби. Використання програм для створення інтер'єру теж не є доцільним через нестачу точних розмірів. Після проведення короткого дослідження популярних програм для полігонального моделювання, було вирішено зупинитись на програмі Maya Autodesk. Програма є відносно безкоштовною, має весь необхідний інструментарій, доступна для операційної системи Microsoft Windows 10, має точну документацію й багато відео-уроків у відкритому доступі. Серед переваг цієї програми також є:

- Підтримка всіх популярних мультимедійних форматів
- Має відкритий вихідний код;
- Наявність вбудованого редактора вузлів
- Створення динамічних ефектів на основі кривих
- Програми від Autodesk мають сумісність з усіма популярними операційними системами
- Гарна репутація надійного та професійного програмного забезпечення

- Архітектура програми зручна для побудування пайплайну, завдяки API.
- Серед недоліків програми:
- Наявність інтерфейсу лише на англійській мові
- Високі системні вимоги
- Складність програми для нових користувачів
- Незначна кількість сполучень клавіш, що трохи сповільнює роботу

Після дослідження було прийняте виважене рішення зупинитись саме на цій програмі.

3.2 Програмне забезпечення для UV-mapping, та текстурування об'єктів

Функція UV-mapping вбудована в Maya, але для оптимізації часу та зручності, було прийнято рішення текстурувати об'єкти в іншій програмі. Тому робити розгортку об'єкта в Maya не має сенсу.

Для текстурування та UV-mapping була обрана програма 3D-Coat. Вона підходить для швидкого процесу, в ній зручно та швидко робити розгортку, ретопологію. Програма постійно розширює свої можливості й відносно нещодавно розробники додали функцію скульптурування. Також вона має достатньо дружній інтерфейс, тому розібратись в її функціях досить легко. Серед переваг ще є авторетопологія та автомапінг.

3.3 Вибір ігрового рушія

Зазвичай ігровий рушій має такі функції як:

- Відсутність необхідності програмування

- Керування
- Фізика
- Необхідні скрипти
- Виявлення колізій
- Рендерінг
- Штучний інтелект

Ігровий рушій використовують розробники ігор для побудови основи ігор. Це допомагає оптимізувати робочий процес та надає більше часу зосередитись на інших деталях.

Для реалізації саме 3D-екскурсії, був обраний ігровий рушій Unreal Engine. Рушій є безкоштовним, постійно удосконалюється. Також навігація та налаштування системи є дуже схожими з іншим програмним забезпеченням. Крім цього Unreal Engine має систему Blueprint. Вона надає можливість швидко реалізувати свої ідеї без додаткових знань. Система Blueprint активно застосовується популярними іграми, адже значно прискорює роботу над проектами.

Серед переваг також наявна можливість швидкого створення прототипу гри та постобробки, велика дружня спільнота та безліч навчальних відеоматеріалів у відкритому доступі, можливість створювати ігри для різних платформ.

Версія ігрового рушія Unreal Engine 4.26.4. Це не остання найновіша версія Unreal Engine 5, адже в останній версії зараз присутня дуже велика кількість помилок.

4 МОДЕЛЮВАННЯ ОБ'ЄКТІВ

Моделювання об'єктів – важливий етап в розробці проекту. Аби досягнути реалістичності й не витратити на це більше часу ніж потрібно, необхідно чітко слідувати алгоритму розробки певного об'єкту чи цілої сцени.

3D-моделювання об'єктів відбувається за допомоги спеціального програмного забезпечення. Необхідно розуміти, що кожна модель – це набір точок у просторі, котрі між собою поєднуються полігонами. Моделі можуть створюватись різними способами: вручну, кодом або автоматично спеціальним 3D сканером. В залежності від вибору способу моделювання, наша сцена чи об'єкт будуть представлені кодом або полігонами у в'юпорті.

4.1 Техніки моделювання

Майже всі 3D моделі можуть бути поділені на дві великі групи[5]:

- Solid (твердотільна модель) – коли створюється сітка деталі, програма створює один з наступних типів елементів на основі активних параметрів сітки для дослідження:
 - Draft quality mesh (програма генерує лінійні чотириохгранні тверді елементи, як на рисунку 4.1)
 - High quality mesh (програма генерує параболічні чотириохгранні тверді елементи, як на рисунку 4.2)

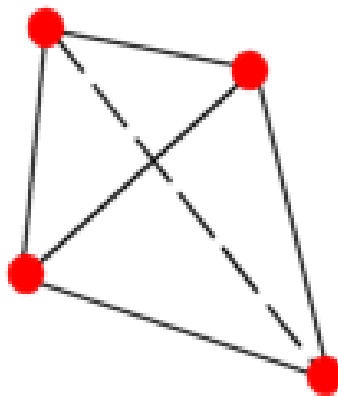


Рисунок 4.1 — Лінійний чотирьохгранний елемент

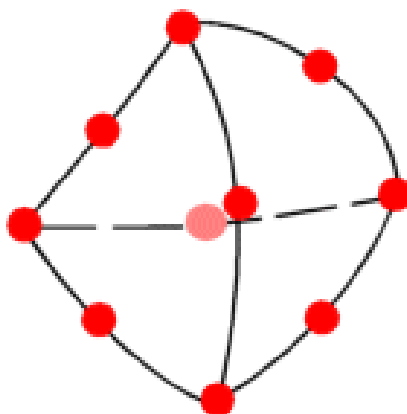


Рисунок 4.2 — Параболічний чотирьохгранний елемент

- Shell, boundary (модель-оболонка) – при використанні елементів оболонки програма створює такі типи елементів в залежності від активних параметрів створення сітки для дослідження:
 - Draft quality mesh (програма генерує трикутні елементи оболонки, як на рисунку 4.3)
 - High quality mesh (програма генерує параболічні трикутні елементи оболонки, як на рисунку 4.4)

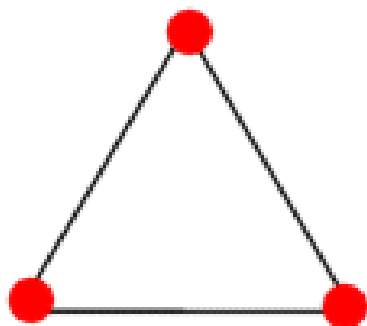


Рисунок 4.2 — Трикутні елементи оболонки

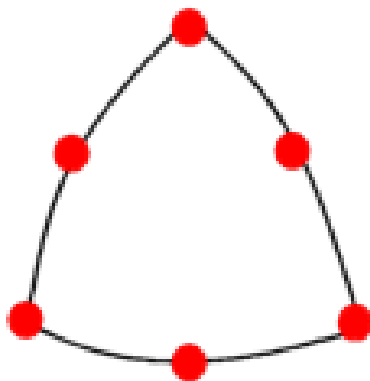


Рисунок 4.3 — Параболічні трикутні елементи оболонки

Можна створити сітки будь-якої твердотільної моделі з чотиригранними елементами, але це може спричинити велику кількість елементів невеликого розміру. Використання більшого розміру сітки елемента призведе до гіршої якості та неточного результату роботи, тому важливо в таких випадках обирати модель типу Shell.

Існує безліч технік моделювання[6]. Всі вони використовуються в різних умовах, тому інколи варто перемикатись з однієї техніки на іншу.

- Цифрове скульптування
- Булеве моделювання
- Лазерне сканування

- Vox-модельювання
- Subdivision модельювання
- NURBS модельювання

В даному випадку використовувалось дві техніки модельювання, а саме: subdivision та булеве модельювання.

Subdivision surface модельювання збільшує кількість полігонів, але разом з цим згладжує криві й робить їх більш круглими. Цей вид модельювання має декілька рівнів, кожний з яких збільшує кількість вершин у 4 рази. Часто після сильного згладжування високополігональну модель запікають в карти нормалей на такій самій за розміром низькополігональній моделі. Це значно оптимізує процес. Для того, щоб залишити потрібні краї гострими, їм можна призначити значення “Hard”. Такий вид модельювання є одним з найрозповсюдженіших, зображено на рисунку 4.5.

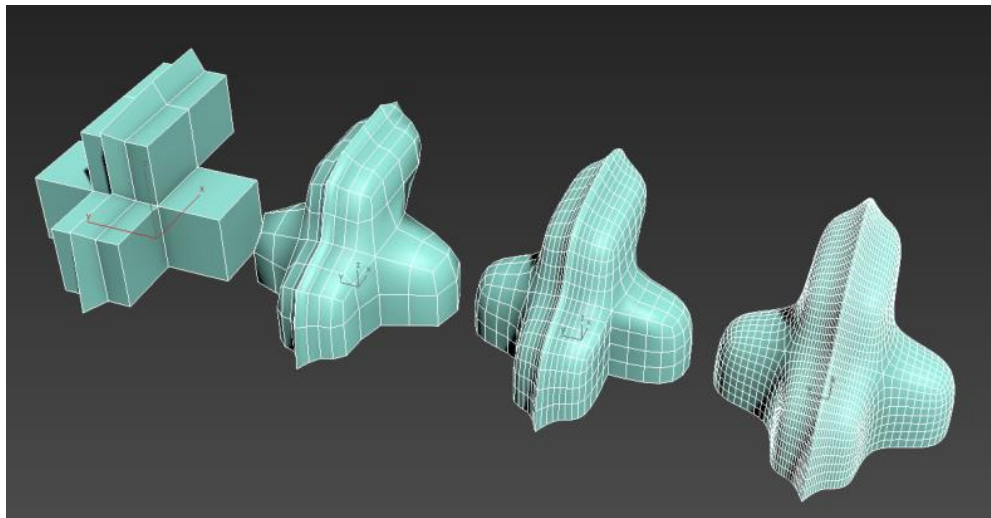


Рисунок 4.4 — Модельювання Subdivision surface

Булеве модельювання засновано на булевих операціях. В такий об’єкт входить одразу два меша (також називаються примітивами). В модельюванні є три можливі операції, як на рисунку 4.6:

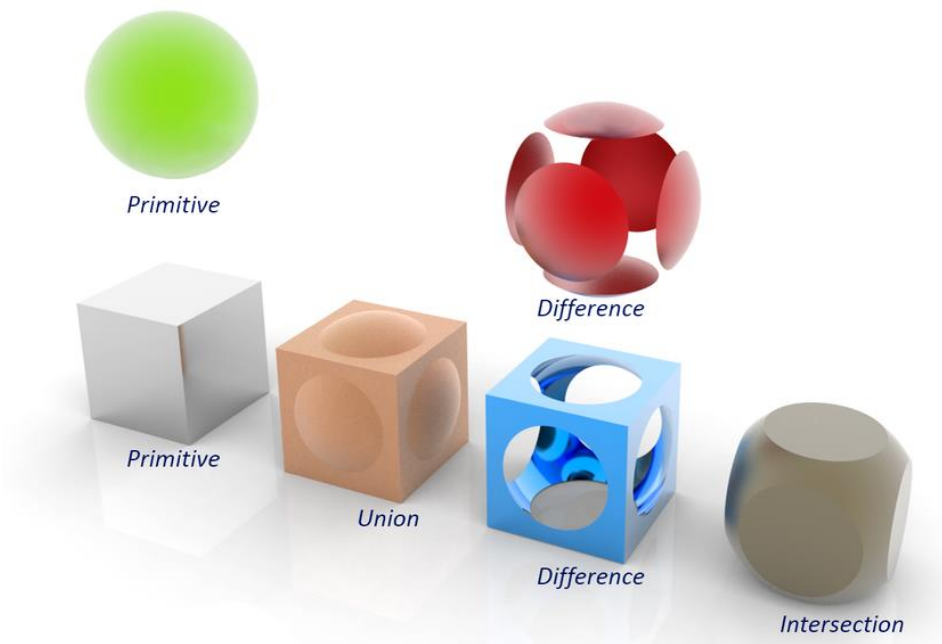


Рисунок 4.5 — Булеві операції

- Union (об'єднання) – коли два об'єкти зливаються в один.
- Difference (різниця) – коли залишається один із варіантів різниці об'єктів
- Intersection (перетин) – коли в готовий булевий об'єкт входять спільні частини вихідного об'єкту.

4.2 Організація робочого процесу

В моделюванні надзвичайно важливим є дотримання правильного порядку усіх етапів. Цикл розробки 3D моделі або ж сцени називають пайплайном. Цикл зображено на рисунку 4.7. Ця схема значно спрощує процес створення проєкту, бо розбиває великі завдання на менші. Пайплайн може відрізнятись, та в основному виділяють такі етапи розробки:

- Пошук референсів

- Блокінг
- Драфт
- Highpoly
- Lowpoly
- UV-mapping
- Текстури
- Рендер

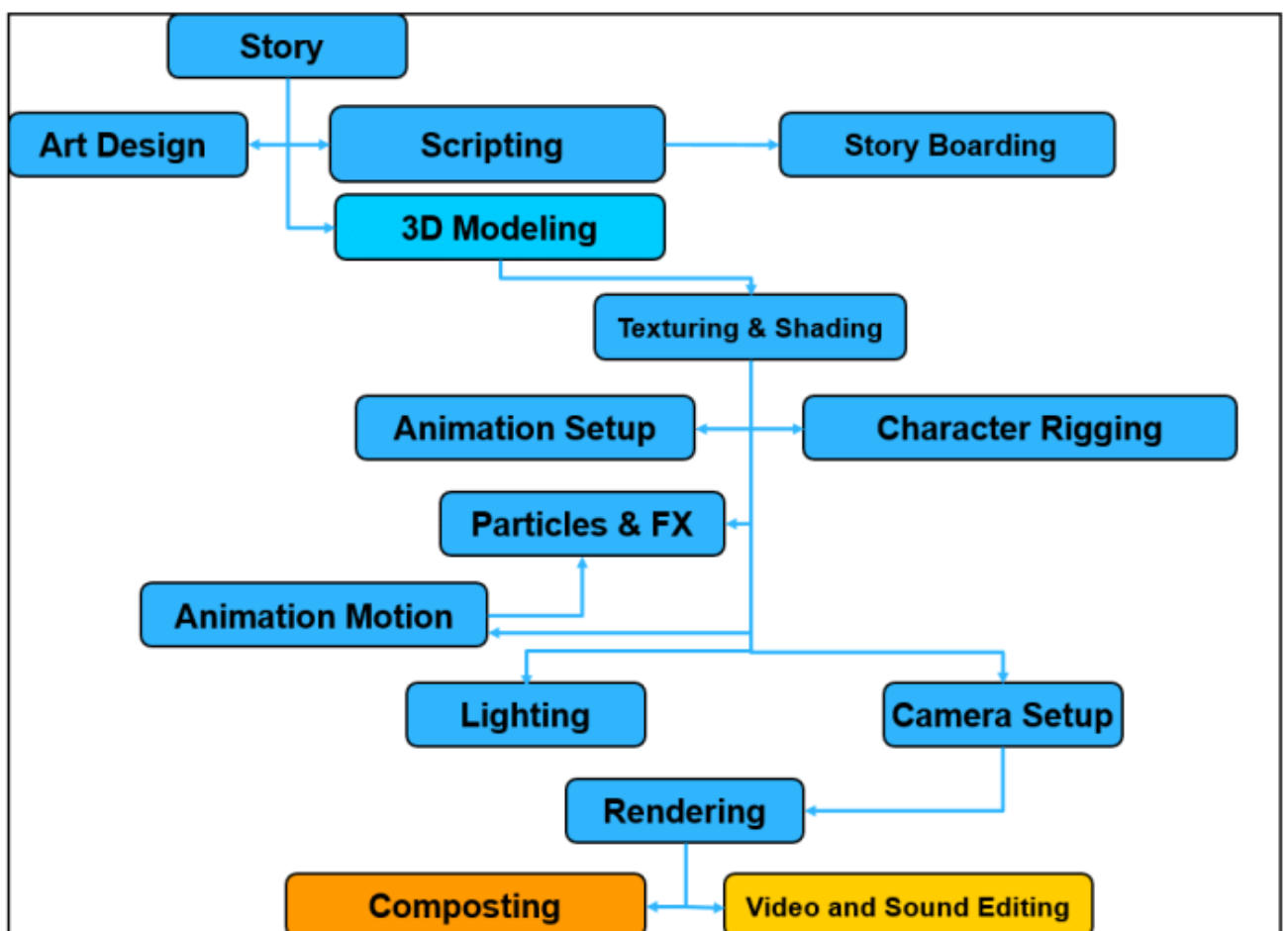


Рисунок 4.6 — Приклад пайплайну

Через великий обсяг роботи та обмежений час, дуже важливо поділити роботу на рівномірні частини та приділити приблизно однакову кількість часу кожному етапу.

4.3 Робота в редакторі для моделювання об'єктів Autodesk Maya

В даному проєкті було важливим віднайти правильні пропорції, щоб досягнути максимально можливої точності й схожості зі справжньою будівлею. Так як розмір сцени дуже великий, було прийнято рішення зробити спочатку малу версію усієї будівлі, як на рисунку 4.8, а потім збільшити її за допомогою розширених налаштувань, щоб всі об'єкти збільшувались не відносно себе, а відносно усієї сцени.

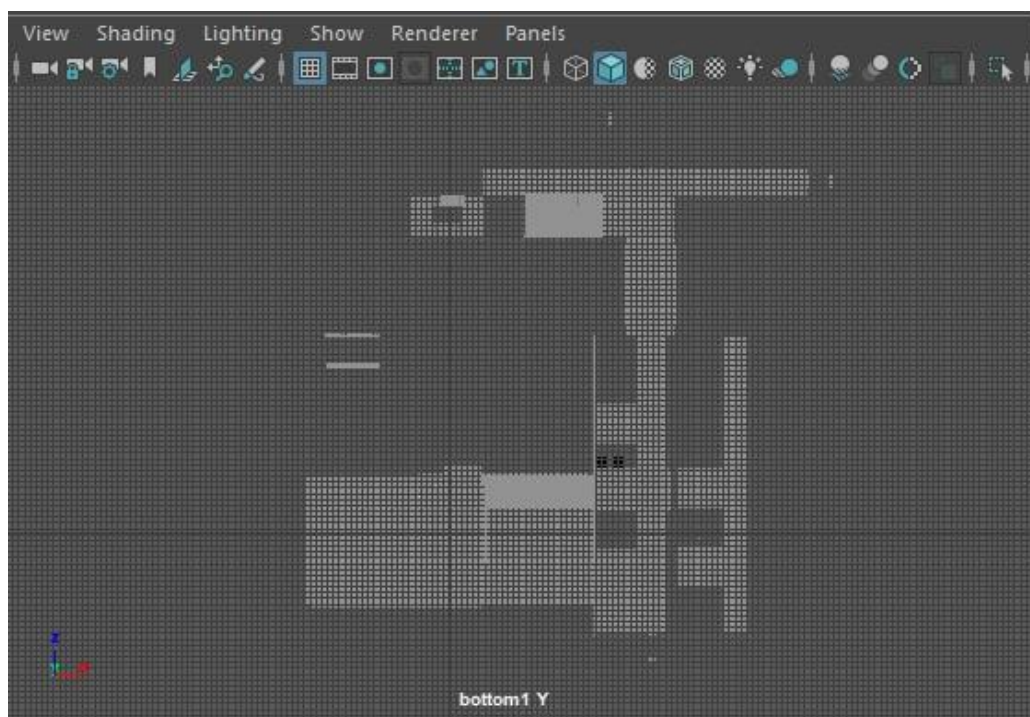


Рисунок 4.7 — Мала версія будівлі

Саме тому, було вирішено почати зі стін та кімнат. Визначившись з розмірами головного входу та холу, зображено на рисунку 4.9, необхідно було зробити перший та другий поверхи (через їхню відмінність), далі поверх в іншій частині будівлі, зображено на рисунку 4.10. Для сполучення поверхів потрібно зробити сходи, як на рисунку 4.11 та, відповідно, кімнати, де вони знаходяться.

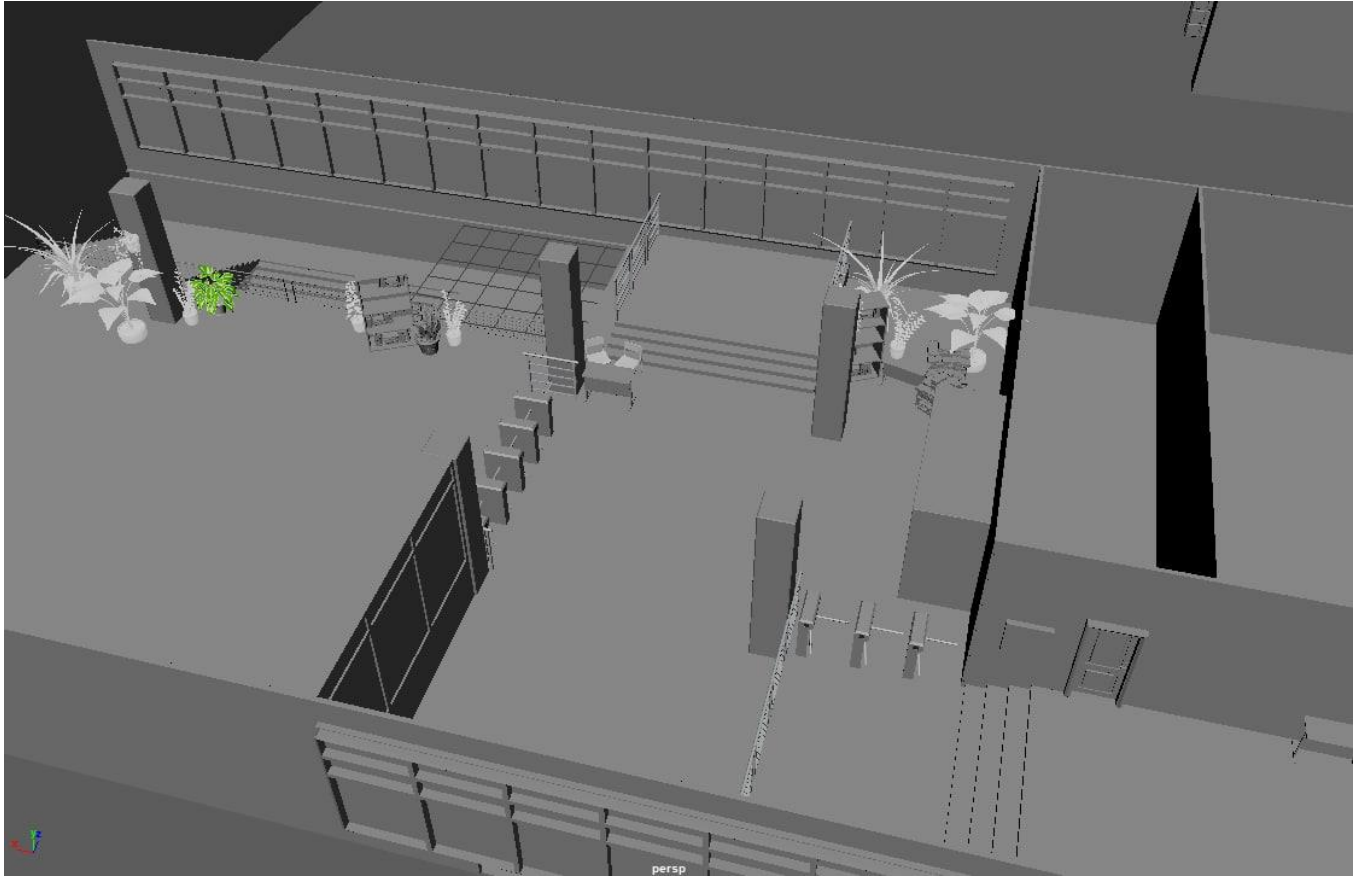


Рисунок 4.8 — Головний вхід

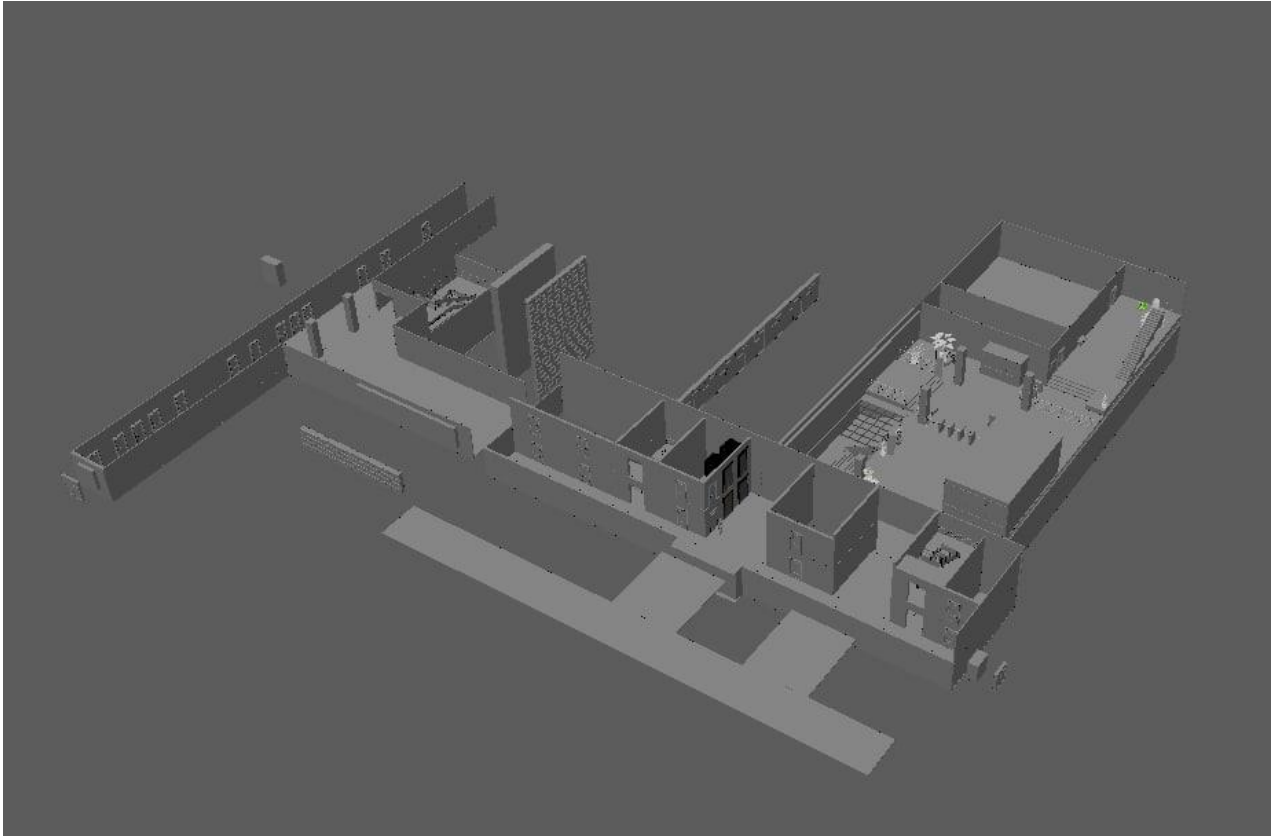


Рисунок 4.9 — 1 та 2 поверхи будівлі

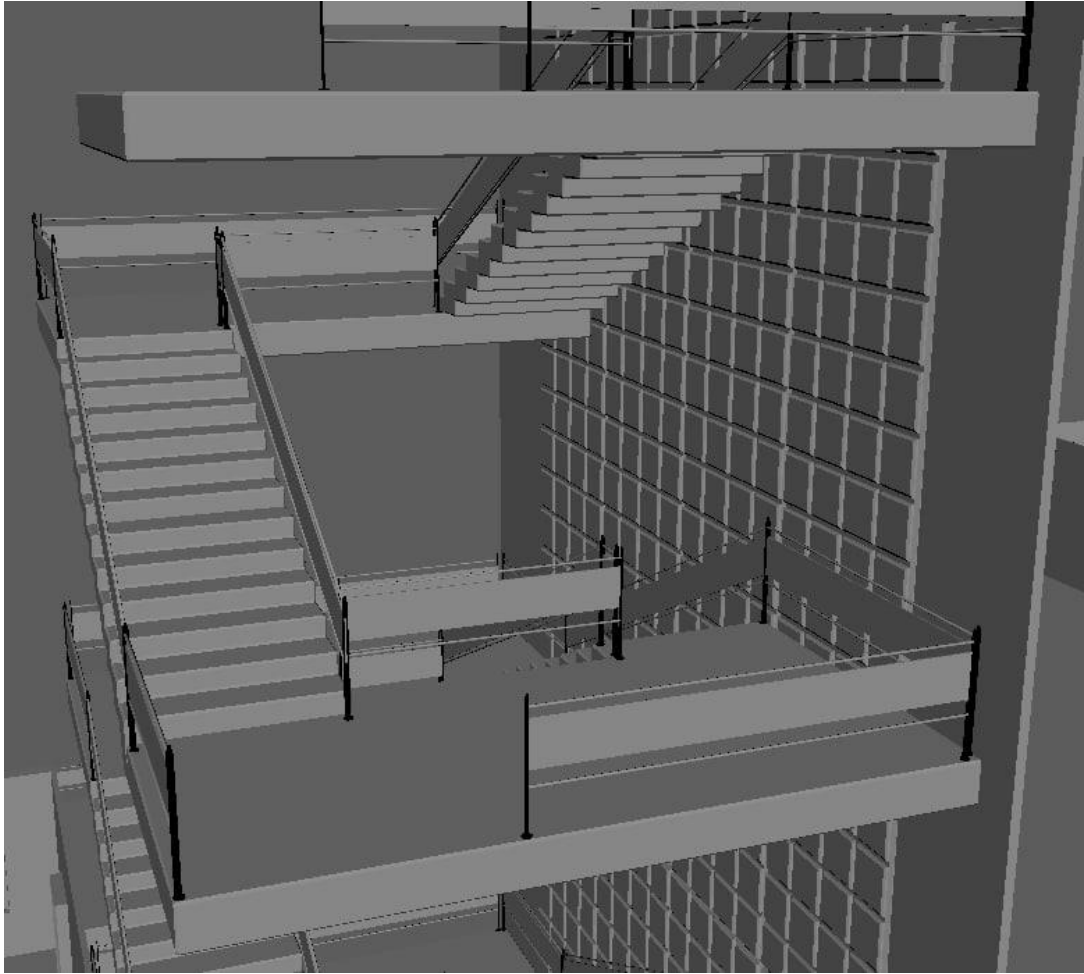


Рисунок 4.10 — Сходи для сполучення поверхів будівлі

Таким чином, в майбутньому ми зможемо здублювати поверхи та зекономимо великий проміжок часу в нашому проєкті. Крім того, важливо розуміти, що проєкт несе саме навігаційну функцію по корпусу університету, й важливим є саме маршрут студента чи викладача до аудиторії, через це досить раціональним рішенням є не моделювати кімнати та лекційні. Це значно оптимізує проєкт та економить ресурси, адже одне з правил оптимізації в моделюванні говорить про те, що не потрібно моделювати те, чого ти не бачиш.

Наступним пунктом було освітлення, а саме вікна та лампи. Для моделювання вікон ми скористались технікою моделювання за допомогою булевих операцій, й вирізали відповідні до вікон отвори у стінах. Також важливо примітити, що скло у вікні

в нас має бути прозорим, тому вікно важливо змодельовати правильно, з двох частин (рама та скло). Після цього, необхідно змодельовати, підготувати усі необхідні декорації. Для того щоб бути впевненим в тому, що в ігровому рушії частини будівлі зійдуться як треба, ми здублювали пару поверхів та перевірили як це буде виглядати.

Після цього настав час збільшити сцену відповідно до її реальних розмірів, як на рисунку 4.12. Можна помітити, що якість зображення сцени значно погіршилась. Це так званий ефект “z-fight”. На якість проєкту він не вплине, та працювати разом з ним раніше було б незручно.

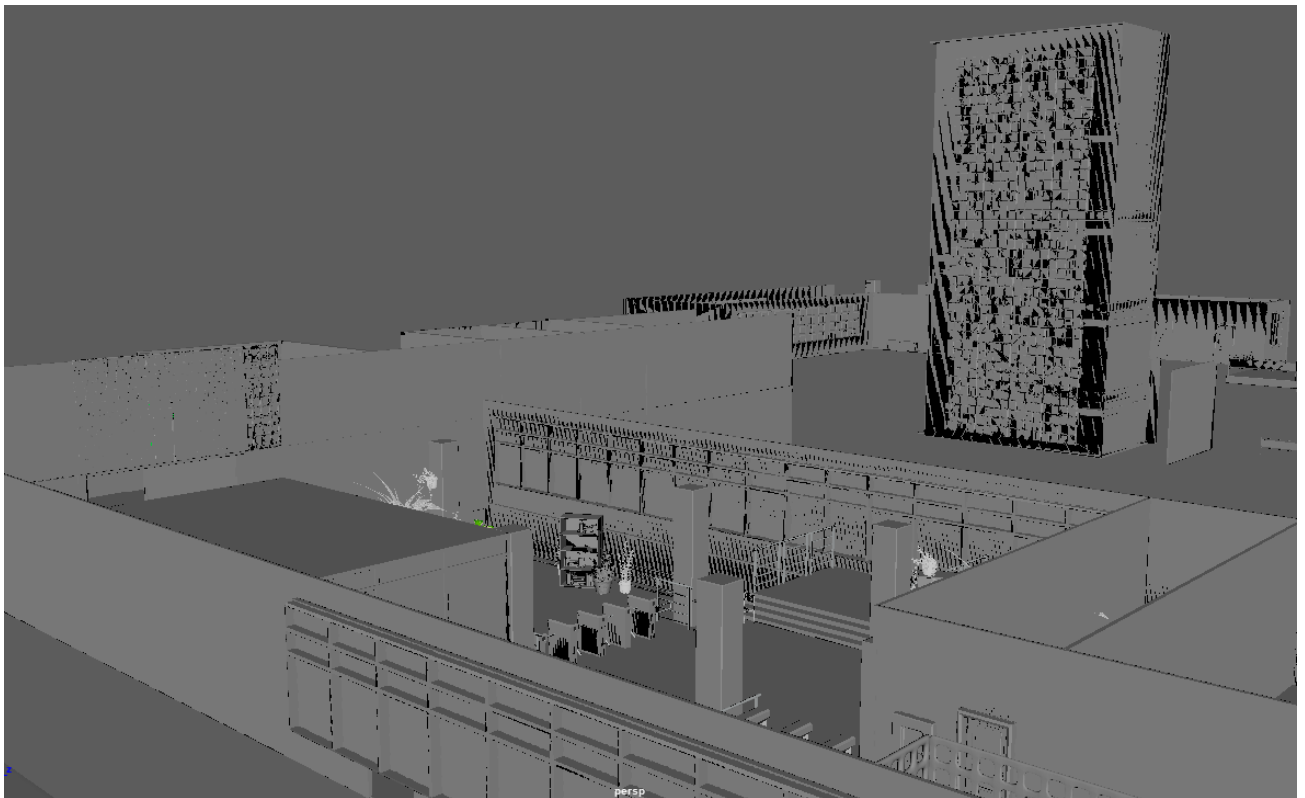


Рисунок 4.11 — Збільшена сцена до реальних розмірів

Далі необхідно було експортувати всі моделі у відповідні папки, зображено на рисунках 4.13-4.14 для подальшого текстуровання чи імпорту в ігровий рушій.

1partFloor1	19.05.2022 18:35	Папка с файлами
1partFloor3-7	19.05.2022 18:35	Папка с файлами
1PartSmallWall	19.05.2022 18:35	Папка с файлами
1partwall1f11	20.05.2022 14:38	Папка с файлами
1partWall1f12	19.05.2022 18:35	Папка с файлами
1partWall2f11	19.05.2022 18:35	Папка с файлами
1partWall2f12	19.05.2022 18:35	Папка с файлами
2 part Hallway	19.05.2022 18:35	Папка с файлами
2 part wall 1	19.05.2022 18:35	Папка с файлами
2 part Wall2 f12-3	19.05.2022 18:35	Папка с файлами
2partplaceforstairs	20.05.2022 15:06	Папка с файлами
benchlowpoly	19.05.2022 18:36	Папка с файлами
BigStairs	20.05.2022 12:39	Папка с файлами
bookshelf	20.05.2022 12:41	Папка с файлами
cagedoor	20.05.2022 12:40	Папка с файлами
CageWithStairs	20.05.2022 12:39	Папка с файлами
Column	20.05.2022 12:39	Папка с файлами
doubledoor	20.05.2022 12:41	Папка с файлами
EmergencyStairs	20.05.2022 12:42	Папка с файлами
expo-books	20.05.2022 12:43	Папка с файлами
Fence Hall	20.05.2022 12:43	Папка с файлами
HallCeiling	20.05.2022 12:44	Папка с файлами
HallFloor	20.05.2022 12:44	Папка с файлами

Рисунок 4.13 — Экспорт усіх моделей

Имя	Дата изменения	Тип	Размер
material	19.05.2022 18:34	Папка с файлами	
texture	20.05.2022 14:07	Папка с файлами	
1partFloor1	19.05.2022 17:43	3D Object	45 КБ

Рисунок 4.14 — Экспорт усіх моделей

Усі наявні об'єкти були експортовані у форматі .fbx. Такий тип файлу легший за файл у форматі .obj, й він достатньо популярний, тому відкривається у багатьох програмах.

4.4 Створення UV розгортки об'єктів та текстурування

Наступним важливим етапом у роботі є підбір матеріалів. Деякі матеріали були взяті з публічної бібліотеки ресурсів для 3D художників. Частину матеріалів необхідно було створити власноруч, тому після роботи з Maya, треба перейти в редактор 3DCoat.

Перш ніж наносити готову текстуру чи фарбувати модель власноруч потрібно зробити розгортку моделі, як на рисунку 4.15. UV-розгортка переносить нашу тривимірну модель на площиність для зручності нанесення текстур. Для того, щоб розгортка, а в подальшому текстури вийшли гарними, необхідно робити розгортку на lowpoly моделі. Це важливо, інакше текстури будуть нанесені спотворено.

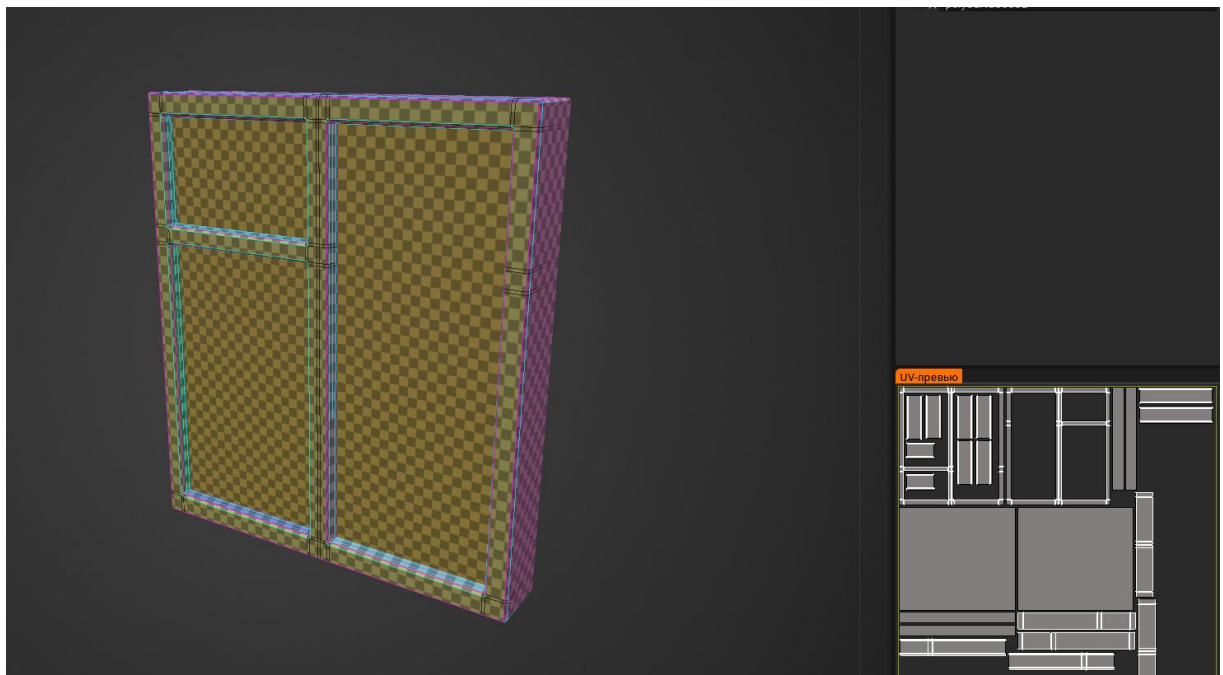


Рисунок 4.15 — Готова UV-розгортка в 3DCoat

Після створення розгортки її необхідно застосувати до нашого об'єкту. Тоді, можна починати текстурувати модель, як на рисунку 4.16 та рисунку 4.17.

Для текстуровання використовувались вже вбудовані смарт матеріали й смарт матеріали створені на базі завантажених текстур каменю, плитки тощо. Щоб створити власний матеріал, ми знайшли потрібну нам текстуру та налаштували новий смарт матеріал, завантаживши до нього його майбутні характеристики та налаштували їх.

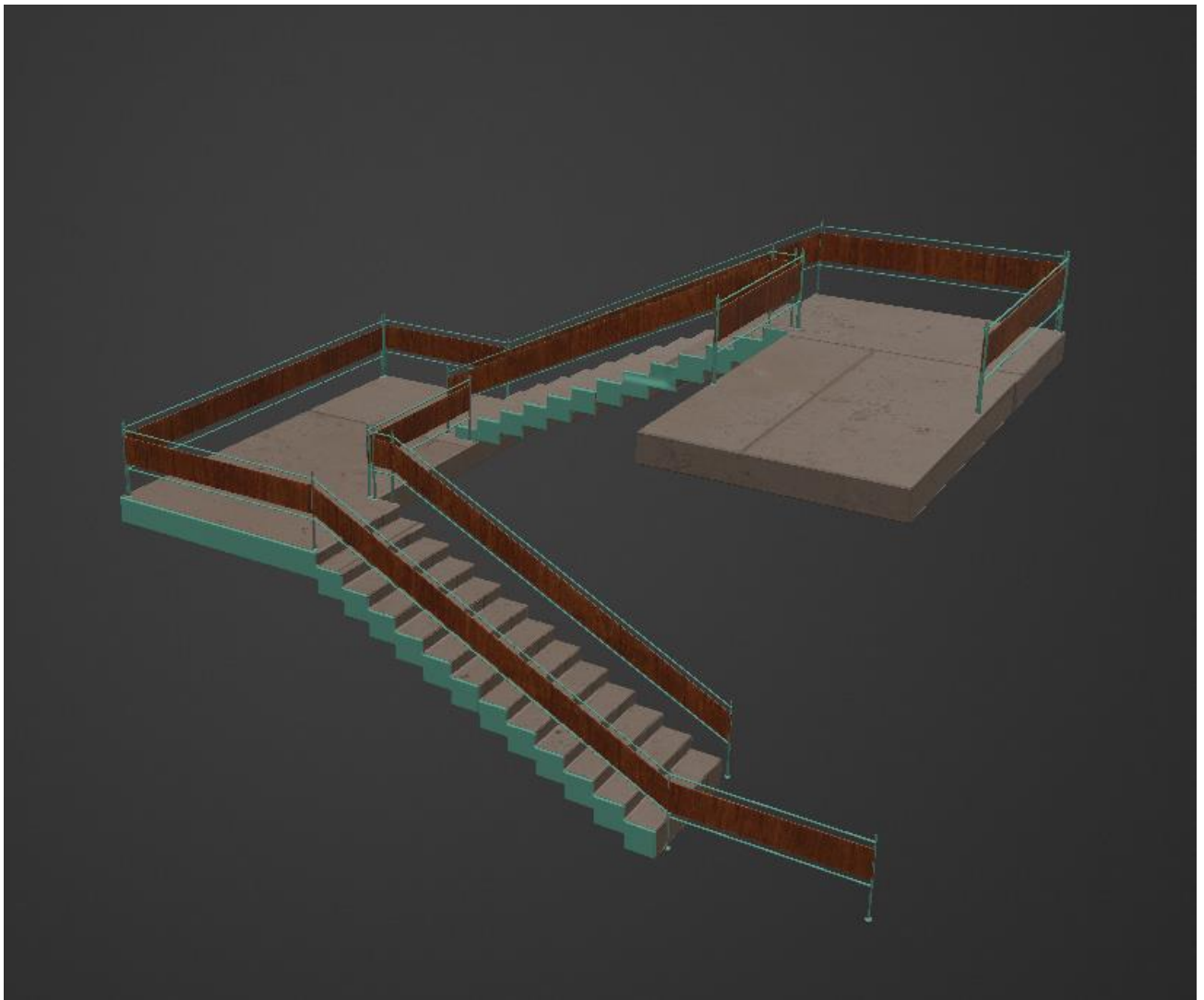


Рисунок 4.16 — Текстуровання моделі

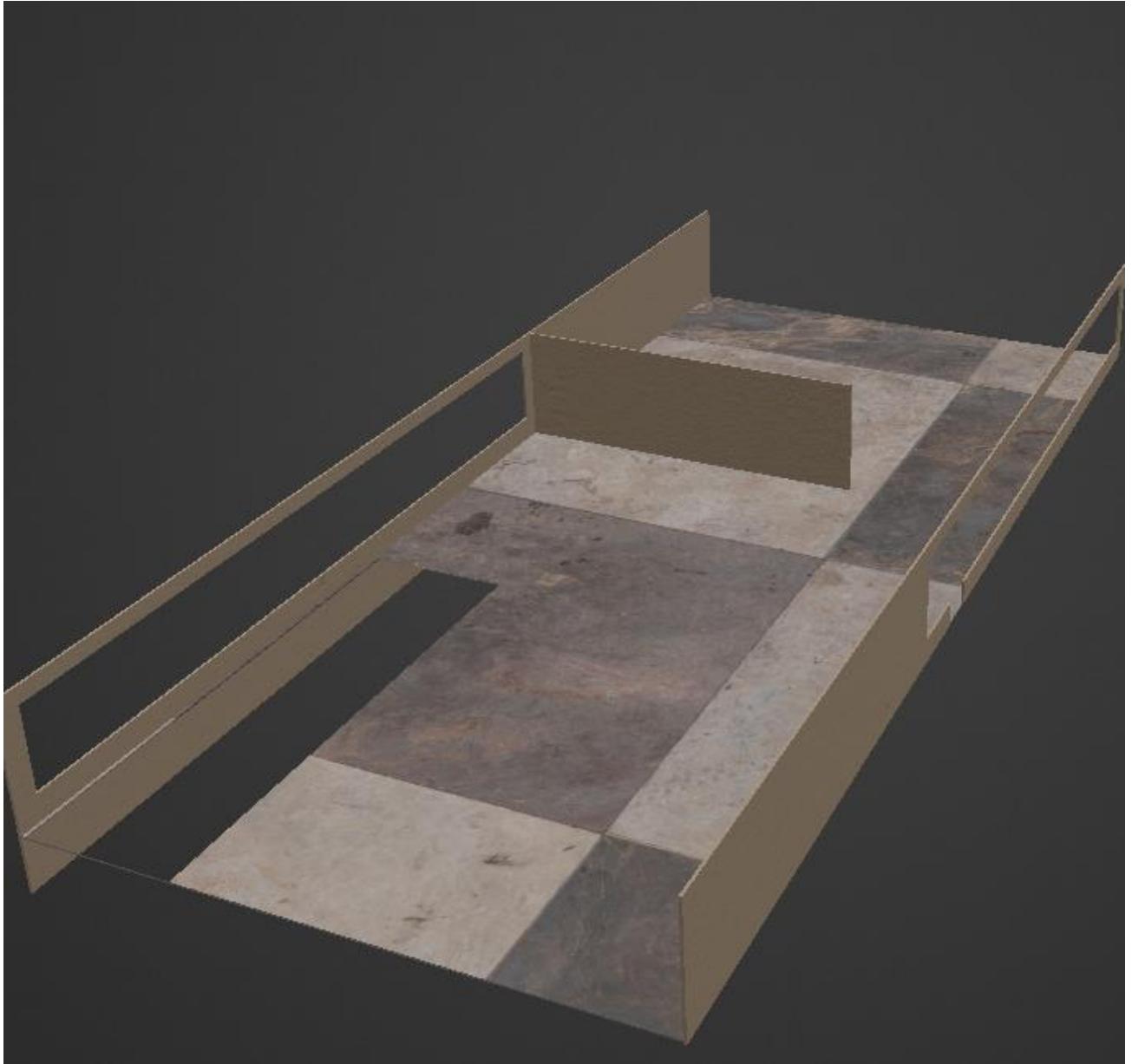


Рисунок 4.17 —Текстурування моделі

Після накладанні текстур, експортуємо об'єкт з готовою розгорткою та текстури під цю розгортку в відповідну папку.

5 СТВОРЕННЯ ТА НАЛАШТУВАННЯ ПРОЄКТУ

Перш ніж почати імпорт наших моделей необхідно створити проєкт та правильно налаштувати його. Спочатку нам пропонується обрати категорію нашого проєкту. Ми обрали категорію ігор. Далі пропонується обрати шаблон. Ми обираємо шаблон “Blank” й переходимо безпосередньо до налаштувань самого проєкту. Відключаємо функцію “ray-traced rendering” для оптимізації проєкту та прибираємо з початкової сцени “starter pack”. Перед імпортування всіх об’єктів та текстур потрібно створити папки для кожної моделі в сцені, як на рисунку 5.1.

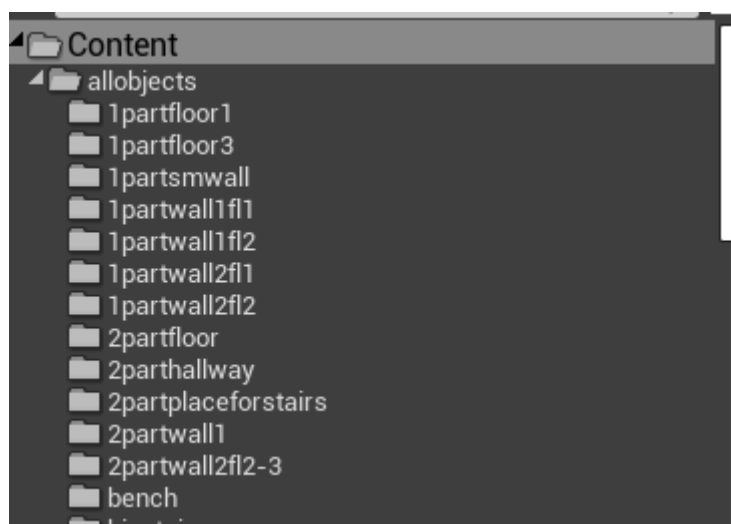


Рисунок 5.1 — Створення відповідних папок в контекстному меню

Важливо створити саме нові папки, програма дуже чутлива й може не дозволити зберегти сцену з текстурованими об’єктами в подальшому.

5.1 Імпорт об’єктів

В попередньо створенні папки, в програму скидаємо спочатку об’єкт, а потім текстури. Важливо при імпорті об’єкту повернути його на 90 градусів, як на

рисунку 5.2. Осі в різних програмах відрізняються, тому при імпорті деяких об'єктів, їх потрібно повертати.

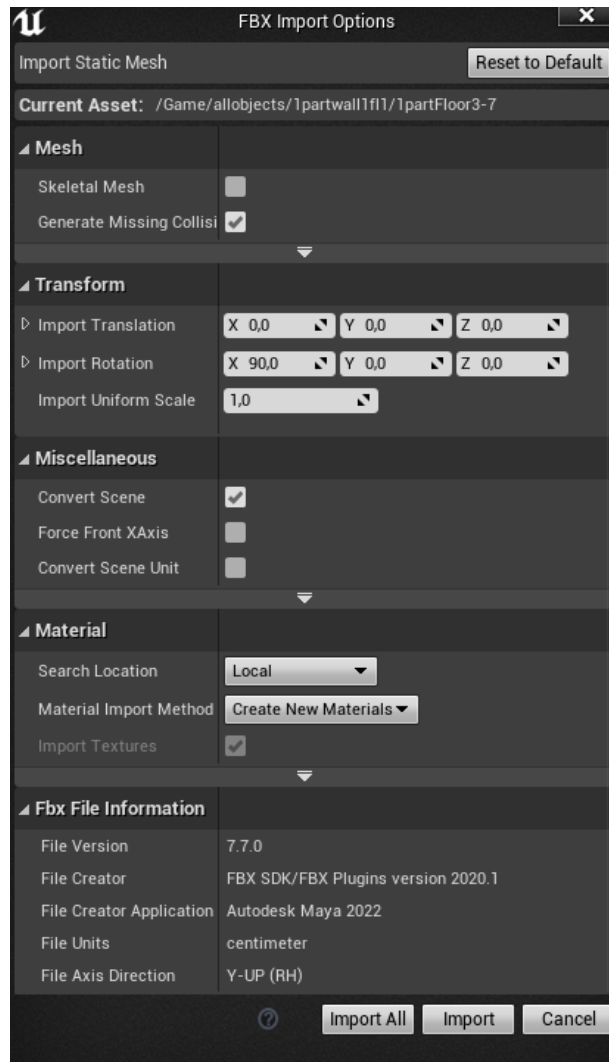


Рисунок 5.2 — Імпорт об'єкту та налаштування

При імпорті об'єкта в нас автоматично створюється його матеріал або декілька матеріалів та в матеріалі вже підключена текстура до каналу “Base color”. Тому ми вже можемо побачити колір об'єкту, проте для досягнення кращого ефекту необхідно підключити й інші текстури до каналів.

Вузол під назвою “Lambert 5” – це наш матеріал. Для того щоб програма створила матеріал з інших текстур, необхідно їх під’єднати у відповідні канали, як на рисунку 5.3. Таким чином ми отримаємо потрібний нам рельєф, металевість, тіні, об’єм та інше. Після зберігання матеріалу можна перевірити як він застосувався на нашому об’єкті за допомогою перегляду матеріалу.

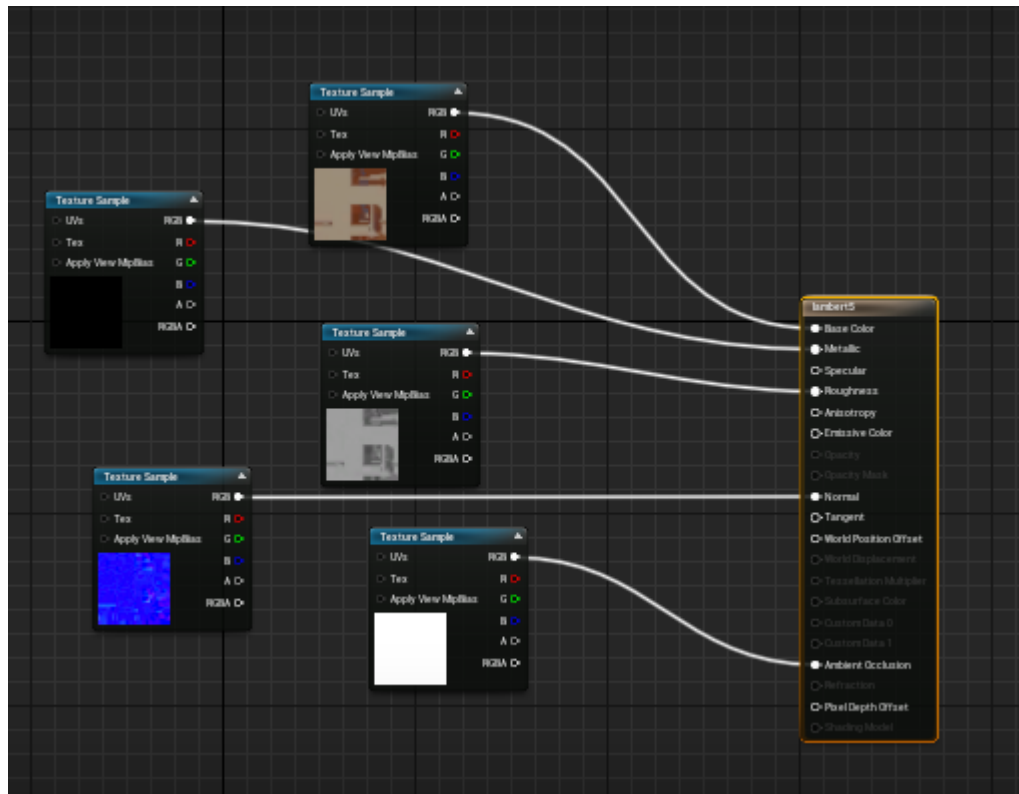


Рисунок 5.3 — Створення матеріалу за допомогою підключення текстур

Для створення матеріалу використовуємо різні текстури. Кожний канал відповідає за певні властивості:

Карта roughness (шорсткості) описує нерівності поверхні, які викликають дифузію світла.

Normal-карти працюють з RGB-інформацією, яка точно відповідає X, Y та Z значенням у 3D-сцені. Ця RGB-інформація повідомляє 3D-редактору точний напрямок нормалей кожного полігону поверхні. Карти normal існують двох типів:

- object-space — для об'єктів, котрі не піддаються деформації (стіл, стіни, підлога);
- tangent-space — для об'єктів, де повинна бути можливість деформації об'єктів (звірі, персонажі);

Зазвичай для створення карт нормалей використовується highpoly та lowpoly моделі, за їхнім порівнянням встановлюють потрібне значення відхилення нормалей.

Карти metallic (металевості). Ця властивість визначає, які частини поверхні є металевими, а які неметалевими. Карта металевості — це зображення у відтінках сірого, де чорні пікселі відповідають неметалам, а білі — металам. Метали відбивають світло того ж кольору, що й метал, і вони відбивають набагато більше світла, ніж неметали.

Ambient occlusion (навколишня оклюзія) — це скалярне значення, записане в кожній точці поверхні, що вказує на середню кількість самооклюзії, що виникає в точці на поверхні. Він вимірює ступінь захищеності місця на поверхні від навколишніх джерел світла.

Дифузна карта є найпоширенішим типом текстурної карти. Вона визначає колір і малюнок об'єкта. Відображення розсіяного кольору схоже на малювання зображення на поверхні об'єкта. Наприклад, якщо ви хочете, щоб стіна була з цегли, ви можете вибрати файл зображення з фотографією цегли.

Це ті карти, що були використані в матеріалах об'єктів в даному проєкті.

Крім цього існує можливість підключити й інші карти:

Specular (зеркальні) карти — це чорно-білі зображення в матеріалах, які не містять PBR, які визначають блиск або відбивну здатність об'єкта в 3D. Він використовується в поєднанні з дифузною картою та картою нормалей для підвищення реалістичності текстури об'єкта.

Карти bump використовують чорно-білі кольори (на відміну від normal map), щоб утискати або видавлювати поверхню геометрії.

Anisotropy (анізотропна карта) — це текстура, яка керує формою дзеркальних відблисків від світла в реальному часі, часто використовується для затінення волосся.

Opacity карти (непрозорості) створюють ефект непрозорості та прозорості. Ви можете вказати області прозорості та непрозорості за допомогою каналу карти прозорості. Ефект прозорості найкраще переглядати на фоні візерунка.

Після створення матеріалу, необхідно перевірити чи правильно лягли текстури на об'єкт, зображено на рисунку 5.4.



Рисунок 5.4 — Перевірка матеріалів

Після того як ми побачили, що усі текстури лягли на об'єкт належним чином, можна переходити до наступної моделі.

5.2 Розміщення об'єктів

Розміщення об'єктів відбувається за допомогою модифікаторів програми, приклад на рисунку 5.5. Обираємо об'єкт з контекстного меню та переміщуємо його в сцену. Якщо нам не підходить розмір об'єкту, ми можемо змінити його розміри.



Рисунок 5.5 — Розміщення об'єктів

Для оптимізації робочого часу, деякі ділянки можна скопіювати, як зображено на рисунку 5.6 та рисунку 5.7.



Рисунок 5.6 — Сцена з об'єктами, що повторюються



Рисунок 5.7 — Сцена з об'єктами, що повторюються

Після розміщення усіх наявних об'єктів, тестуємо простір.

6 ІНФОРМАЦІЯ КОРИСТУВАЧЕВІ 3D ЕКСКУРСІЄЮ

Для того, щоб скористатись програмною системою необхідно запустити файл “MyProject3”. Приклад запуску програми наведений нижче на рисунку 6.1.

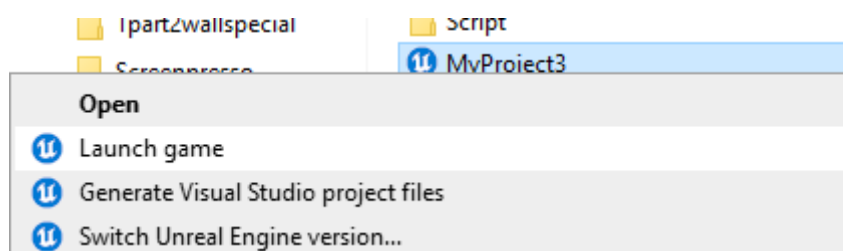


Рисунок 6.1 — Запуск програми

Далі завантажиться проєкт й користувач може вільно пересуватися по сцені за допомогою стрілок на клавіатурі, як на рисунку 6.2.



Рисунок 6.2 — Пересування користувача між кімнатами

В подальшому даний проєкт можна інтегрувати на сайт факультету.

ВИСНОВКИ

Метою дипломної роботи було створити 3D модель корпусу №5, НТУУ “КПІ ім. І. Сікорського”.

Перед виконанням завдання було вивчено та досліджено усі необхідні матеріали, інструменти та програмне забезпечення для реалізації 3D проєктів. У ході робочого процесу було порівняно різне програмне забезпечення. Порядок робочого процесу був попередньо організований та спланований.

Протягом роботи над проєктом засвоєні такі навички: моделювання під subdivision, булеві функції, створення правильної розгортки для об’єктів, текстурування, робота з ігровим рушієм.

Було проаналізовано аналогічні 3D екскурсії, знайдено та враховано їх недоліки, а саме: обмежене пересування та час завантаження екскурсії. Обрані методи моделювання є найоптимальнішими. Для всіх змодельованих об’єктів були створені UV розгортки та текстури.

У результаті маємо готовий продукт, який у подальшому можна додати до сайту факультету або використовувати самостійно.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Unreal Engine 5 Documentation веб-сайт. URL: <https://docs.unrealengine.com/5.0/en-US/>.
2. Autodesk Maya Documentation веб-сайт. URL: <https://help.autodesk.com/view/MAYAUL/2023/ENU/>.
3. Unreal Engine 4 Tutorial - How to Create Materials in UE4 веб-сайт. URL: <https://bit.ly/3GckK13>.
4. Уроки Unreal Engine 4 / 5 Начало веб-сайт. URL: <https://bit.ly/3Pve02R>.
5. SOLIDWORKS Online Help веб-сайт. URL: https://help.solidworks.com/2019/english/SolidWorks/sldworks/r_welcome_sw_online_help.htm?id=0a0394933a7e4bdba85e9bca78564738#Pg0.
6. What is 3D modeling in animation and how should we start the workflow? веб-сайт. URL: <https://dreamfarmstudios.com/blog/a-quick-guide-to-3d-modeling/>.
7. Reallusion веб-сайт. URL: <http://www.reallusion.com/>
8. Mactech Scripting Languages веб-сайт. URL: <http://preserve.mactech.com/articles/mactech/Vol.15/15.09/ScriptingLanguages/index.html>.
9. Koloro NURBS веб-сайт. URL: <https://koloro.ua/blog/3d-tekhnologii/vidy-3d-modelirovaniya-poligonalnoe-splajnovoe-i-nurbs-modelirovanie.html>.
10. Akenine-Möller Tomas. Non-photorealistic rendering. Natick, Mass.: AK Peters, 2001, 145p.
11. Gooch Bruce. Real-time rendering. Natick, Mass.: AK Peters, 2004, 80p.
12. GNU general public license веб-сайт. URL: <https://www.gnu.org/licenses/gpl-3.0.html>.
13. Jensen, Henrik Wann. Realistic image synthesis using photon mapping. Natick, Mass.: AK Peters, 2001, 237p.

14. Blinn Jim. Jim Blinn's corner : a trip down the graphics pipeline. San Francisco, Calif.: Morgan Kaufmann Publishers, 1996, 39p.
15. The RADIANCE Lighting Simulation and Rendering System веб-сайт. URL: <https://floyd.lbl.gov/radiance/papers/sg94.1/>.

ДОДАТОК А

3D екскурсія в корпусі №5 КПІ ім. Ігоря Сікорського

Текст програми

УКР.НТУУ “КПІ ім. Ігоря Сікорського” ТЕФ_АПЕПС_ТР81329_22Б 12-1

Аркушів 5

Київ – 2022

Реалізація 3D екскурсії на ігровому рушії Unreal Engine 4.

```
#pragma once
```

```
#include "CoreMinimal.h"
```

```
#include "Components/ActorComponent.h"
```

```
#include "STUCoreTypes.h"
```

```
#include "STUHealthComponent.generated.h"
```

```
class UCameraShakeBase;
```

```
class UPhysicalMaterial;
```

```
UCLASS(ClassGroup = (Custom), meta = (BlueprintSpawnableComponent))
```

```
class SHOOTTHEMUP_API USTUHealthComponent : public UActorComponent
```

```
{
```

```
    GENERATED_BODY()
```

```
public:
```

```
    USTUHealthComponent();
```

```
    FOnDeathSignature OnDeath;
```

```
    FOnHealthChangedSignature OnHealthChanged;
```

```
    UFUNCTION(BlueprintCallable, Category = "Health")
```

```
    bool IsDead() const { return FMath::IsNearlyZero(Health); }
```

```
    UFUNCTION(BlueprintCallable, Category = "Health")
```

```
    float GetHealthPercent() const { return Health / MaxHealth; }
```

```
    float GetHealth() const { return Health; }
```

```
bool TryToAddHealth(float HealthAmount);
```

```
bool IsHealthFull() const;
```

```
protected:
```

```
UPROPERTY(EditDefaultsOnly, BlueprintReadWrite, Category = "Health", meta =  
= (ClampMin = "0.0", ClampMax = "1000.0"))
```

```
float MaxHealth = 100.0f;
```

```
UPROPERTY(EditDefaultsOnly, BlueprintReadWrite, Category = "Heal")
```

```
bool AutoHeal = true;
```

```
UPROPERTY(EditDefaultsOnly, BlueprintReadWrite, Category = "Heal", meta =  
(EditCondition = "AutoHeal"))
```

```
float HealUpdateTime = 1.0f;
```

```
UPROPERTY(EditDefaultsOnly, BlueprintReadWrite, Category = "Heal", meta =  
(EditCondition = "AutoHeal"))
```

```
float HealDelay = 3.0f;
```

```
UPROPERTY(EditDefaultsOnly, BlueprintReadWrite, Category = "Heal", meta =  
(EditCondition = "AutoHeal"))
```

```
float HealModifier = 5.0f;
```

```
UPROPERTY(EditDefaultsOnly, BlueprintReadWrite, Category = "VFX")
```

```
TSubclassOf<UCameraShakeBase> CameraShake;
```

```
UPROPERTY(EditDefaultsOnly, BlueprintReadWrite, Category = "Health")
```

```
TMap<UPhysicalMaterial*, float> DamageModifiers;
```

```
virtual void BeginPlay() override;
```

```
private:
```

```
float Health = 0.0f;
```

```
FTimerHandle HealTimerHandle;
```

```
UFUNCTION()
```

```
void OnTakeAnyDamage(
```

```
    AActor* DamagedActor, float Damage, const class UDamageType*  
DamageType, class AController* InstigatedBy, AActor* DamageCauser);
```

```
UFUNCTION()
```

```
void OnTakePointDamage(AActor* DamagedActor, float Damage, class  
AController* InstigatedBy, FVector HitLocation,  
    class UPrimitiveComponent* FHitComponent, FName BoneName, FVector  
ShotFromDirection, const class UDamageType* DamageType,  
    AActor* DamageCauser);
```

```
UFUNCTION()
```

```
void OnTakeRadialDamage(AActor* DamagedActor, float Damage, const class  
UDamageType* DamageType, FVector Origin, FHitResult HitInfo,  
    class AController* InstigatedBy, AActor* DamageCauser);
```

```
void HealUpdate();
```

```
void SetHealth(float NewHealth);
```

```
void PlayCameraShake();
```

```
void Killed(AController* KillerController);
```

```

void ApplyDamage(float Damage, AController* InstigatedBy);
float GetPointDamageModifier(AActor* DamagedActor, const FName&
BoneName);

```

```

void ReportDamageEvent(float Damage, AController* InstigatedBy);
};
const auto Player = Cast<APawn>(GetOwner());
const auto VictimController = Player ? Player->Controller : nullptr;

GameMode->Killed(KillerController, VictimController);
}

```

```

float USTUHealthComponent::GetPointDamageModifier(AActor* DamagedActor,
const FName& BoneName)
{
const auto Character = Cast<ACharacter>(DamagedActor);
if (!Character //
!Character->GetMesh() //
!Character->GetMesh()->GetBodyInstance(BoneName))
return 1.0f;

const auto PhysMaterial = Character->GetMesh()->GetBodyInstance(BoneName)-
>GetSimplePhysicalMaterial();
if (!PhysMaterial || !DamageModifiers.Contains(PhysMaterial)) return 1.0f;

return DamageModifiers[PhysMaterial];
}

```

```

void USTUHealthComponent::ReportDamageEvent(float Damage, AController*
InstigatedBy)
{
    if (!InstigatedBy || !InstigatedBy->GetPawn() || !GetOwner()) return;

    UAISense_Damage::ReportDamageEvent(GetWorld(), //
    GetOwner(), //
    InstigatedBy->GetPawn(), //
    Damage, //
    InstigatedBy->GetPawn()->GetActorLocation(), //
    GetOwner()->GetActorLocation());
}

```