

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)
Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки
(повна назва)

АТЕСТАЦІЙНА РОБОТА

Пояснювальна записка

другий (магістерський)

(рівень вищої освіти)

(тема)

Розробка віртуальної 3D моделі виробничого приміщення

Виконав: студент 2 курсу, гр. АУТПм-19-1
Тітов В.А.

(прізвище, ініціали)

Спеціальність 151 Автоматизація
та комп'ютерно-інтегровані технології

освітньої програми Автоматизація управління
технологічними процесами

(код і повна назва напрямку)

Тип програми освітньо-професійна

(повна назва освітньої програми)

Керівник проф. Цимбал О.М.

(посада, прізвище, ініціали)

Допускається до захисту
зав. кафедри

(підпис)

Невлюдов І.Ш.

(прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій

(Повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки

(Повна назва)

Рівень вищої освіти Другий (магістерський)

Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології

(Код і повна назва)

Тип програми освітньо-професійна

(Освітньо-професійна або освітньо-наукова)

Освітня програма Автоматизація управління технологічними процесами

(Повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(Підпис)

« _____ » _____ 2020 р.

ЗАВДАННЯ НА АТЕСТАЦІЙНУ РОБОТУ

студентові Тітову Віктору Артуровичу

(прізвище, ім'я, по батькові)

1. Тема роботи Розробка віртуальної 3D моделі виробничих приміщень

затверджена наказом по університету від 02.11.2020 р. № 1510 Ст.

2. Термін подання студентом роботи до екзаменаційної комісії 14.12.2020 р.

3. Вихідні дані до роботи Креслення виробничих приміщень, програмне забезпечення для 3D моделювання 3Ds Max Unreal Engine Archicad промислового робота, мова програмування HTML 5 Java Script.

4. Перелік питань, що потрібно опрацювати в роботі

4.1 Вступ

4.2 Аналіз програмно-технічних засобів 3D-моделювання виробничих приміщень

4.3. Формування віртуальної 3D-моделі виробничого приміщення

4.4. Реалізація 3D-моделі виробничих приміщень

4.5. Результати тестування розробленої 3D-моделі

4.6 Охорона праці

4.7 Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Демонстраційний матеріал у форматі презентації PowerPoint (*.ppt) – 11 с. формату А4.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз технічного завдання	04.11.2020 р.	
2	Аналіз програмно-технічних засобів 3D моделювання	11.11.2020 р.	
3	Розробка віртуальної 3D моделі виробничого приміщення	25.11.2020 р.	
4	Тестування розробленої 3D моделі	30.11.2020 р.	
5	Оформлення пояснювальної записки	10.12.2020 р.	
6	Подання роботи до ЕК	14.12.2020 р.	

Дата видачі завдання 02.11.2020 р.

Студент _____
(Підпис)

Тітов В.А.

Керівник роботи _____
(Підпис)

проф. Цимбал О.М.
(Посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка містить: 121 с., 81 рис., 2 дод., 46 джерел.

ВІРТУАЛЬНА МОДЕЛЬ ПРИМІЩЕННЯ, 3D МОДЕЛЬ, VR, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, СИСТЕМА ВІЗУАЛІЗАЦІЇ, 3D MAX, UNREAL ENGINE.

Об'єкт дослідження – формування моделі віртуальної реальності на прикладі проектування виробничого приміщення.

Предмет дослідження – підвищення швидкодії та ефективності методів формування віртуальної реальності.

Мета – формування віртуальної реальності та розробка 3D моделі приміщення виробничого типу.

Наукова новизна полягає у дослідженні та розробці моделей та засобів формування віртуальної реальності.

Методи дослідження та апаратура: для розв'язання задачі використовувався аналіз сучасних інформаційних технологій; тестування віртуальної моделі виробничого приміщення здійснювалося на персональній IBM сумісній ЕОМ (тактова частота процесора – 2 ГГц, обсяг оперативної пам'яті – 20 Гб, обсяг жорсткого диска – 500 Гб).

Результати: програмний – запропоновано рішення для створення віртуального приміщення для web додатку, що має переваги над існуючими рішеннями за функціональними показниками та швидкодією ; практичний – розроблено віртуальну 3D модель та web додаток виробничого приміщення.

Область застосування – розміщення віртуальних моделі виробничого приміщення на web додатках.

ABSTRACT

The explanatory note contains: 121 pp., 81 figs., 2 appendices, 46 sources.

VIRTUAL PREMISES MODEL, 3D MODEL, VR, SOFTWARE, VISUALIZATION SYSTEM, 3D MAX, UNREAL ENGINE.

The object of research - the formation of a model of virtual reality on the example of designing a production facility.

The subject of research is to increase the speed and efficiency of methods of virtual reality formation.

The goal is to form an augmented reality and develop a 3D model of a production-type room.

The scientific novelty lies in the research and development of new models and means of shaping virtual reality.

Research methods and equipment: analysis of modern information technologies was used to solve the problem; testing of the virtual model of the production room was carried out on a personal IBM-compatible computer (processor clock speed - 2 GHz, RAM - 20 GB, hard disk capacity - 500 GB).

Results: software - a solution for creating a virtual room for a web application, which has advantages over existing solutions in terms of functionality and performance; practical - developed a virtual 3D model and web application of the production room.

Scope – placement of virtual models of industrial premises on web applications

ЗМІСТ

Скорочення та умовні позначки	7
Вступ	8
1 Аналіз програмно-технічних засобів 3D-моделювання виробничих приміщень	10
1.1 Аналіз методів формування віртуальної та доповненої реальності	10
1.2 Програмно-технічні засоби 3D моделювання	12
1.3 Приклади реалізації віртуальних 3D-моделей виробничих приміщень	27
1.4 Висновки до розділу та постановка мети	32
2 Формування віртуальної 3D-моделі виробничого приміщення	34
2.1 Системи координат засобів 3D-моделювання та основні операції	34
2.2 Формування моделей стандартних об'єктів	36
2.3 Рендерінг сформованих моделей	53
2.4 Висновки до розділу	64
3 Реалізація 3D-моделі виробничих приміщень	65
3.1 Формування вхідної інформації про виробничі приміщення	65
3.2 Моделювання приміщень та об'єктів в приміщеннях	67
3.3 Конвертування отриманих моделей у засоби віртуальної реальності	82
3.4 Тестування розробленої 3D моделі	89
3.4.1 Тестування на локальному сервері	89
3.4.2 Тестування на віддаленому сервері	93
4 Охорона праці	97
4.1 Вимоги до організації робочих місць	97
4.2 Вимоги до електробезпеки	98
4.3 Фактори, що впливають на функціональний стан розробника	100
Загальні висновки	103
Перелік джерел посилання	105

Додаток А Лістинг коду програми	110
Додаток Б Демонстраційний матеріал	114

ПЕРЕЛІК СКОРОЧЕНЬ

AR – доповнена реальність;

CAD – система автоматизованого проектування;

VR – віртуальна реальність;

UE – Unreal Engine;

ЧПУ – числове програмне управління;

ВСТУП

В сучасному світі комп'ютерна графіка набуває велике значення, швидкі темпи розвитку технічного прогресу надає комп'ютерній графіці розвиток в напрямках архітектури, дизайну, ігрової індустрії. Потужність комп'ютерних обчислень дозволяє обробляти та відтворювати складні сцени в режимі реального часу без втрати швидкості і якості відображення. Розвиток 3D графіки тісно перетинається з доповненою реальністю та віртуальною реальністю. В архітектурі і містобудуванні віртуальні будівлі з прогулянками по приміщеннях і віртуальні міста знаходять все більш широке застосування.

Моделювання фотореалістичної реконструкції об'єктів надає можливість на етапі проектування працювати із замовником в режимі реального часу ,також використати 3D моделі в процесі навчання, що є сучасним і перспективним напрямом. Проте, на сьогодні важливою є проблема якісного відображення об'єктів у доповненій реальності. Важливою складовою у відтворенні віртуальної є генерація освітлення, тому що від освітлення та згенерованих тіней залежить як користувач буде сприймати віртуальну або доповнену реальність. Існуючі програмні рішення даної проблеми на сьогодні не дозволяють отримати достатньо реалістичний результат візуалізації об'єктів доповненої реальності. Для даної роботи використовується програмне забезпечення: Autodesk 3D Max та Unreal Engine 4. Використання Web-технологій в поєднанні з можливостями віртуального світу дозволяє створювати інтерактивний освітній контент.

Актуальність даної проблеми постає у необхідності покращення та модернізації існуючих віртуальних екскурсій для відповідності новим вимогам користувачів та прискорення розвитку проектування приміщень на ринку. Одним із найбільш очевидних способів покращення якості розробки є комбінування існуючих систем моделювання та розробки віртуальної реальності.

Об'єкт дослідження – формування моделі віртуальної реальності на прикладі проектування виробничого приміщення.

Предмет дослідження – підвищення швидкодії та ефективності методів формування віртуальної реальності.

Мета – формування доповненої реальності та розробка 3D моделі приміщення виробничого типу.

Наукова новизна полягає у дослідженні та розробці нових моделей та засобів формування віртуальної реальності.

Для досягнення мети необхідно виконати наступні завдання:

- дослідити сучасні методи формування та розробки доповненої реальності;
- розробити віртуальну модель приміщення за допомогою 3D редакторів;
- провести оптимізацію моделей та текстур для збільшення швидкодії
- сформувати доповнену реальність з одночасним відстеженням місцеположення та взаємодією;
- розробити Web додаток для формування доповненої реальності на сайті з використанням розробленої моделі.
- оформити атестаційну роботу відповідно ДСТУ 3008-2015 [1], навчальному посібнику з дипломного проектування [2], методичних вказівок до випускної кваліфікаційної роботи рівня «Магістр» [3] та положенню про протидію академічному плагіату [4].

В ході виконання магістерської атестаційної роботи опубліковано тези доповіді [5].

1 АНАЛІЗ ПРОГРАМНО-ТЕХНІЧНИХ ЗАСОБІВ 3D-МОДЕЛЮВАННЯ ВИРОБНИЧИХ ПРИМІЩЕНЬ

1.1 Аналіз методів формування віртуальної та доповненої реальності

Перший прототип системи «віртуальної реальності» – «Sensorama», який з'явився на початку 1960-х років, розроблений кінематографістом Мортонем Хейлігом і надавав користувачам можливість отримання мультисенсорного віртуального досвіду «Sensorama». В 1966 році інформатик А. Сазерленд замінив інфрачервону камеру комп'ютерними зображеннями. Джарон Ланьє вчений в галузі інформатики ввів термін «віртуальна реальність» в 1989 році [6]. З 1994 року передбачалося, що б клієнтам проходити через потенційні об'єкти на своєму комп'ютері. Ідея полягала в тому, що клієнт входить до офісу, і клієнту надається декілька варіантів розробленого приміщення. Значним прискоренням розвитку у галузі віртуальної реальності стали смартфони, завдяки повсюдному використанню яких віртуальна реальність набула широкого застосування не лише в освіті, а й у таких сферах, як розваги, в тому числі комп'ютерні ігри, охорона здоров'я, управління роботами, музеї, автомобілебудування [7].

На сьогоднішній день під визначенням візуалізація представляється зображення зовнішньої або внутрішньої частини приміщення. Візуалізація може бути статичним кадром та може бути анімована за допомогою камери та анімаційних ключів. Розробник визначає шлях камери, ракурс та швидкість руху камери. Однак глядач бачить лише анімоване відео, та не має можливості інтерактивно прогулятися по будівлі. З розвитком засобів розробки ігор дана інтерактивна візуалізація почала застосовуватись до архітектурної візуалізації. Даний тип візуалізації для замовника є більш інформативним ніж статичний кадр, однак для розробника є більш важким оскільки потрібно відтворити

більше кадрів, визначити та анімувати шлях камери [8]. Досвід віртуального середовища можна додатково покращити, використовуючи VR-шоломи, шоломи із вбудованими екранами та датчиками руху, або середовища VR-Save, кімнати з проєкціями на кожній стіні для візуалізації. Завдяки цим засобам клієнт розвиває власний досвід перегляду, може змінювати в інтерактивному режимі окремі деталі або зміну приміщення в цілому. Слід виділити 3 основні види реальності (рис. 1.1):

- віртуальна реальність (VR) занурює користувачів в повністю штучну цифрову середу;
- доповнена реальність (AR) накладає віртуальні об'єкти на реальне середовище;
- змішана реальність (MR) не тільки накладає, але і прив'язує віртуальні об'єкти до реального світу.

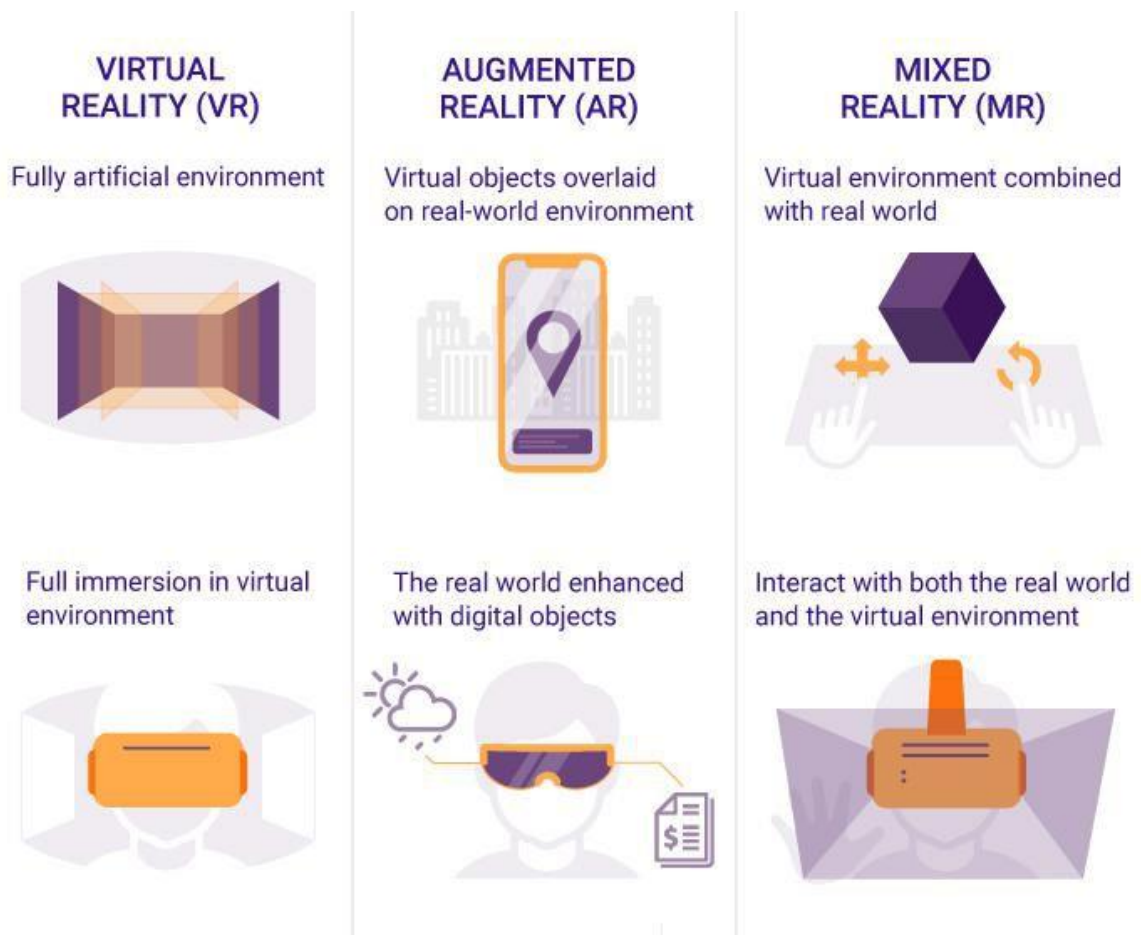


Рисунок 1.1 – Види реальностей (VR)

Слід зазначити, що кожна із типів реальності набула широкого застосування в багатьох галузях життєдіяльності людини [9].

Віртуальна реальність (Virtual Reality, VR) – це технологія, яка дозволяє користувачеві повністю зануритися у штучно створене за допомогою комп'ютера віртуальне середовище. VR забезпечує пересування користувача в цифровому середовищі, взаємодію з ним, та, за допомогою додатків, можливість чути звуки. Для більшого контролю VR використовуються контролери та маніпулятори. Для використання потрібно носити спеціальну гарнітуру VR. Більшість VR-навушників підключаються до комп'ютера (Oculus Rift) або до ігрової консолі (PlayStation VR), але існують і окремі пристрої, серед яких найпопулярнішим є Google Cardboard. Приклад віртуальної реальності наведено на рисунку 1.2.



Рисунок 1.2 – Приклад віртуальної реальності (VR)

Віртуальна реальність здатна не тільки надати відомості про саме явище, а й продемонструвати його з будь-якою мірою деталізації. Прикладами є моделювання небезпечних ситуацій (операції на серці, управління надшвидкісним поїздом, ліквідація пожежі), коли можна занурити глядача в

будь-яку з цих ситуацій без найменших загроз для життя. Інший аспект залучення користувача до модельованого процесу: віртуальна реальність дає змогу змінювати сценарії, впливати на хід експерименту або вирішувати завдання в ігровій і доступній для розуміння формі. Під час віртуального заняття можна побачити світ минулого очима історичного персонажа, відправитися в подорож по людському організму в мікрокапсулі або обрати правильний курс на якомусь кораблі. Користувач отримує функцію фокусування – формування віртуального світу, який оточить глядача з усіх боків на всі 360 градусів, дасть змогу цілком зосередитися на матеріалі і не відволікатися на зовнішні подразники. З'являється можливість проводити віртуальні заняття від першої особи і відчувати свою присутність в намальованому світі – одна з головних особливостей віртуальної реальності. Це дає змогу проводити заняття цілком у віртуальній реальності. Підвищення ефективності навчання з використанням технологій віртуальної реальності обумовлене також тим, що заняття з використанням сучасних технологій викликають великий інтерес, результатом чого стає підсилення навчальної мотивації та активності учнів. Як вказував один з відомих дослідників віртуальної реальності О. В. Юхвід, всі звіти про реалізацію різних навчальних програм на основі VR технологій повідомляють про підвищений інтерес учнів до подібної форми занять й ентузіазм, з яким вони готуються до кожного заняття, вивчаючи теоретичний матеріал, який потім зможуть пропрацювати у віртуальному середовищі [10]. При використанні віртуальної реальності (VR) клієнт може вільно пересуватися в приміщенні або сцені. Можливості візуалізації різко зросли за останні кілька років до того моменту, коли майже фото реалістичні візуалізації можна робити в режимі реального часу. Інтегрування технології ігрового движка в тривимірну анімацію забезпечує швидку попередню візуалізацію складної анімації. Використання віртуальної реальності відкриває багато нових можливостей в навчанні та освіті, які є доволі складними, затратними за часом або дорого коштують при традиційних підходах [11].

Доповнена реальність (Augmented Reality, AR) – це техніка візуалізації, яка комбінується з об'єктів реального світу і об'єктами створеними у за допомогою 3D редакторів. Основна задача доповненої реальності полягає в поєднанні живої реальності з віртуальною. Характерною рисою цієї технології є отримання динамічних і статичних даних в реальному часі за допомогою візуалізації об'єктів. Виділяють три основні функціональні характеристики доповненої реальності:

- комбінування об'єктів реального світу з віртуальними елементами;
- об'єкти що взаємодіють у реальному часі;
- об'єкти які відображені на будь-якому дисплеї з урахуванням положення в просторі (координати і кут нахилу).

Технологія доповненої реальності дозволяє «накладати» віртуальний контент на реальний світ [12]. Приклад наведено на рисунку 1.3.



Рисунок 1.3 – Приклад доповненої реальності (AR)

Змішана реальність (Mixed Reality, MR) – розробка в технологіях реальності, яка іноді викликає плутанину, в першу чергу через те, що так називають різний досвід. Змішана реальність починається з реального світу – віртуальні об'єкти накладаються на реальний світ та можуть взаємодіяти з ним.

У цьому випадку користувач залишається в реальному середовищі, поки до нього додається цифровий контент; крім того, користувач може взаємодіяти з віртуальними об'єктами. Змішана реальність поєднує в собі елементи віртуальної та доповненої реальності виходячи із поставлених задач. Зараз вона має багато сфер для вивчення та набула широкого застосування при у медицині. Змішана реальність має тенденцію поєднувати кращі характеристики віртуальної реальності і доповненої реальності. Змішана реальність поєднує в собі обидва аспекти віртуального світу з аспектами реального світу. У MR користувач залишається в реальному середовищі до якого додається VR [13]. Приклад реалізації представлено на рисунку 1.4.



Рисунок 1.4 – Приклад змішаної реальності (MR)

1.2 Програмно-технічні засоби 3D моделювання

3D Max програмне забезпечення для 3D-моделювання, анімації і візуалізації при створенні ігор і проектуванні, розробляється і видається компанією Autodesk. Також включає в себе функції геометричного моделювання, приклад моделювання представлено на рисунку 1.5. 3D Max надає наступні способи редагування графіки:

- полігональне моделювання використовується для моделювання складних об'єктів;
- моделювання на базі примітивів;
- моделювання на основі сплайнів (ліній).

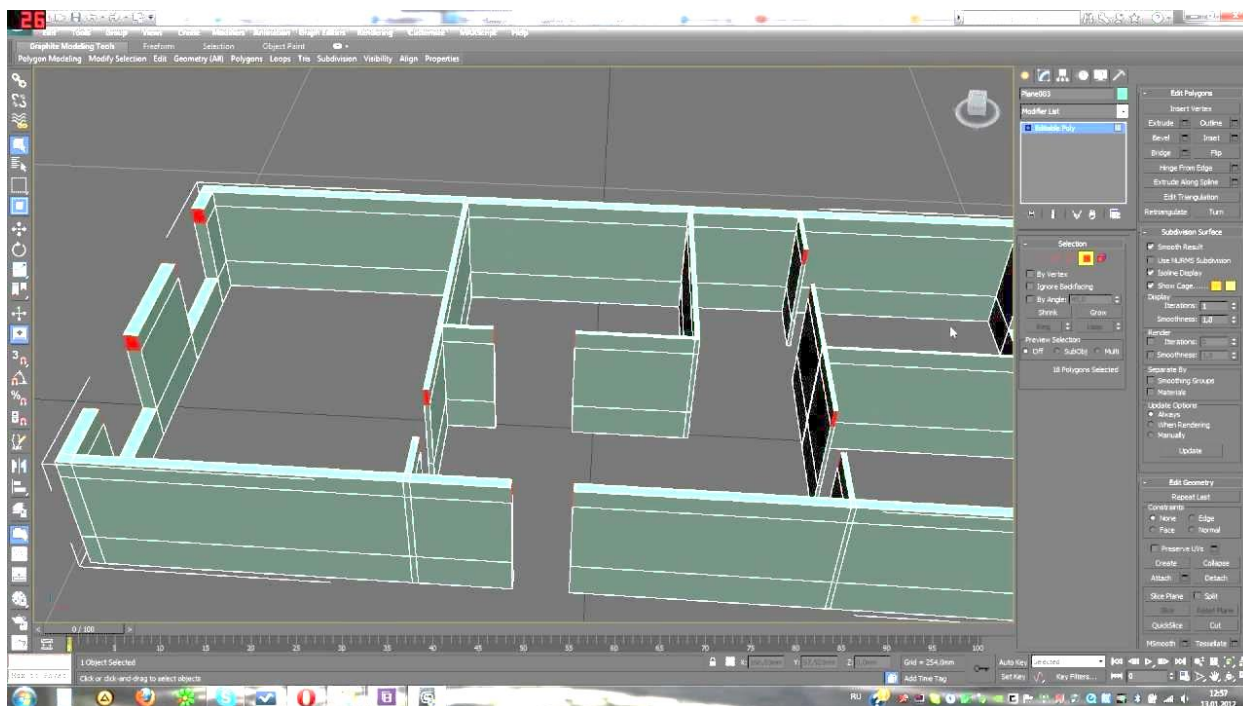


Рисунок 1.5 – Приклад полігонального моделювання

Візуалізація є заключним етапом роботи над сценою. Після візуалізації проявляються властивості матеріалів об'єктів та ефекти освітлення, які застосовані в сцені. Візуалізація виконується за допомогою математичних алгоритмів та робить прорахунок кожного пікселя. Однак час розрахунку залежить від складності завдання [14].

Лазерне сканування технологія, що дозволяє створити цифрову тривимірну модель об'єкта. Більшість моделей лазерних сканерів мають вбудовану відео – або фотокамеру, завдяки чому фінальний об'єкт може бути відтворено як у реальності. Вигляд 3D-сканера наведено на рисунку 1.6.



Рисунок 1.6 – Вигляд 3D-сканера

Отримані дані обробляються в програмних продуктах. 3D-сканери представляють дані у вигляді хмари точок. За допомогою ПО можна усунути помилки в відсканованій моделі, спроектувати повноцінні параметричні твердотільні моделі, а також додати до моделі якості фізичного об'єкта.

3D-сканування вирішує наступні завдання:

- сканування концептуальних моделей кузова;
- калібрування і градування ємностей;
- контроль технологічного оснащення для вакуумного формування, в тому числі композитних матеріалів;
- сканування фасадів будівель для архітектурно-проектних робіт в середовищі AutoCAD;
- контроль геометрії (в тому числі геометричний контроль в процесі експлуатації для вимірювання зносу, вхідний і вихідний контроль виробів)
- зворотне проектування (для відновлення і / або оптимізації форми об'єкта, зворотний інжиніринг та побудова CAD-моделі);
- вимір реальних об'єктів для реконструкції або перепланування будівель, споруд, об'єктів.

Даний метод дозволяє відтворити об'єкти або елементи в разі відсутності конструкторської документації, також слід зазначити, що даний процес автоматизації надає високу точність вимірювань, дозволяє заощадити час і підвищити якість кінцевого продукту. В отриману 3D-модель вносяться необхідні корективи, можна змінити масштаб, геометрію і т.д. Щоб отримати кінцевий результат, а саме віртуальну модель, потрібно імпортувати інформацію до програмного забезпечення [15]. Задачею програмного забезпечення створення віртуальної тривимірної копії фізичного об'єкта для подальшого використання в 3D редакторах або технологічної підготовки виробництва (CAD).

Загалом проводиться сканування і створюються CAD-моделі таких різнопланових об'єктів, промислове приміщення, віртуальна збірка моста, сканування людей. Приклад реалізації сканування людей представлено на рисунку 1.7.



Рисунок 1.7 – Сканування бюсту

Принцип роботи лазерного сканера полягає в вимірі часу проходження лазерного променя від випромінювача до поверхні, що відбиває промінь і

повертається назад до приймача, тим самим визначає відстань до об'єкта (рис 1.8). Сканер складається з лазерного далекоміра, адаптованого для роботи з високою частотою і блоку розгортки лазерного променя. Як блок розгортки в сканері виступають сервопривід і полігональне дзеркало або призма. Сервопривід відхиляє промінь на задану величину в горизонтальній площині, при цьому повертається вся верхня частина сканера. Розгортка у вертикальній площині здійснюється за рахунок обертання або коливання дзеркала [16].

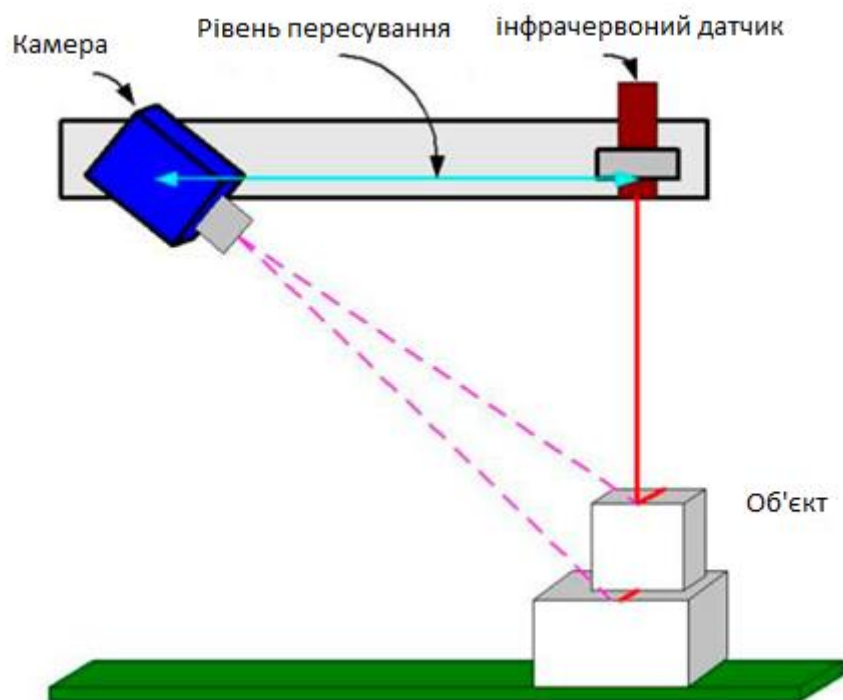


Рисунок 1.8 – Принцип роботи 3D сканера

Розрізняють дві основні технології 3D сканування контактну (контактує з об'єктом) та безконтактна (без контакту з об'єктом). Найбільш поширеним та актуальним є безконтактна технологія сканування, що дозволяє створювати моделі об'єктів просто надіславши на них лазерний промінь, світло, хвилі. Сканер застосовується на відстані і здатний створити копію важкодоступного об'єкта без фізичного контакту з ним. На ринку сканери з цієї групи

представляє модель XYZprinting. Недоліком є те, що при недостатності освітлення об'єкт потрібно додатково освітити [17].

Wrap – це наявність сценаріїв макросів для автоматизації процесів найпростіший спосіб зберегти, перетворити текстуру і фактуру поверхні, в тому числі і в карту текстури інноваційна можливість перетворення скручених поверхонь в плоскі для їх вимірювання, моделювання текстури і фактури, створення 2D-ескізів; простий і оперативний спосіб створити з області точок 3D-модель для друку;

Geomagic Control X – гнучкий програмний продукт для виявлення і вирішення проблем в області контролю якості, що пропонує багатофункціональні і інтуїтивно зрозумілі інструменти вимірювання, управління і аналізу.

Geomagic for Solidworks – програмне рішення для реверс-інжинірингу з широким набором функцій. Це набір програмних інструментів, який забезпечує розширені можливості по використанню хмар точок і багатокутників в процесі проектування. Являє собою плагін, сумісний з популярними моделями 3D-сканерів і підтримує імпорт стандартних форматів точкових і полігональних файлів. Geomagic for SolidWorks – це швидкісна автоматизована обробка хмари точок; потужні інструменти вирівнювання автоматична обробка поверхні формування поперечних перерізів сіток 3D-порівняння з аналізом відхилень початкового рівня контроль відхилень на всіх етапах проектування створення високоякісних твердотільних моделей інтеграція з промисловими 3D-сканерами для роботи за прямою схемою «сканер – SolidWorks» [18].

Готова CAD-модель може бути використана для виготовлення виробу як адитивними, так і традиційними методами. При контролі якості в серійному виробництві 3D-сканер, в порівнянні з класичними інструментами виміру, дозволяє істотно заощадити час і трудовитрати, в тому числі за рахунок часткової автоматизації процесу.

Unity елемент для розробки об'єктів віртуальної та доповненої реальності. Основним принципом роботи в Unity3D є керування та трансформація об'єктів. За допомогою додавання різних компонентів та модифікаторів до об'єктів дозволяє удосконалювати та розширяти функціональні можливості об'єкту. За допомогою змінних компонента з'являється можливість впливати на взаємодію між об'єктами в сцені, що є основою для створення фізичних явищ. Приклад реалізації віртуального приміщення у Unity3D наведено на рисунку 1.9.

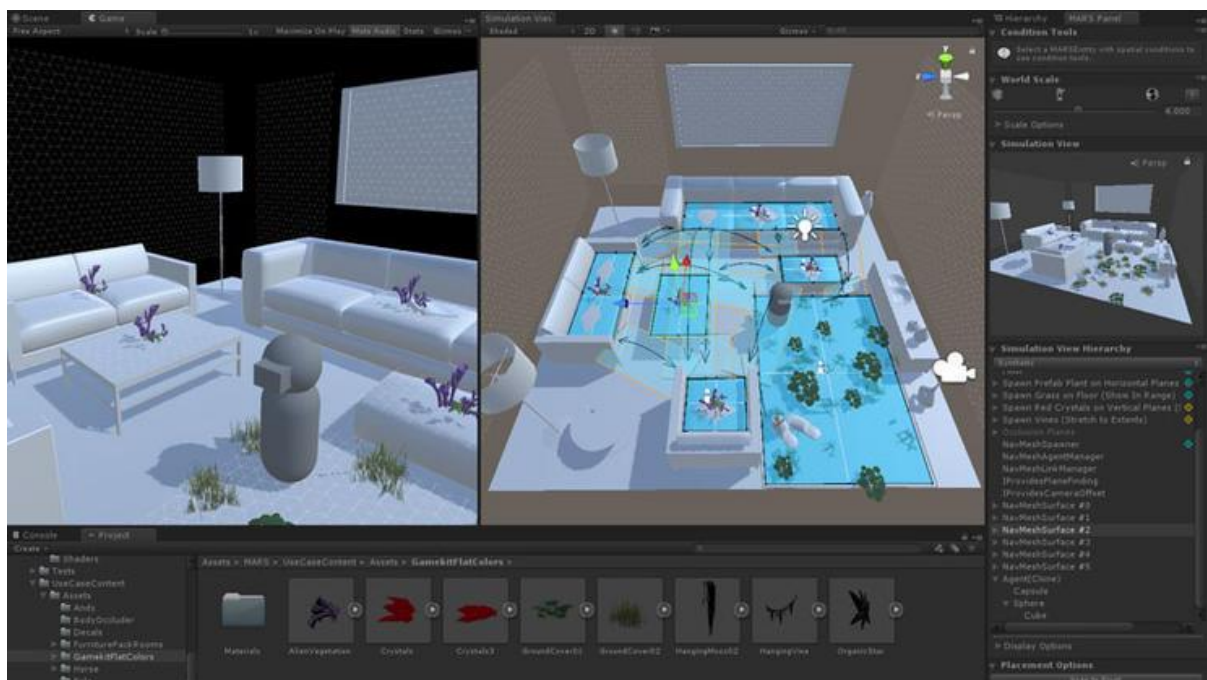


Рисунок 1.9 – Реалізація віртуального приміщення у Unity3D

В якості об'єктів у Unity використовуються ресурси (Assets), зображення (текстури), 3D моделі, звукові файли. Всі об'єкти мають один базовий компонент Transform, який задає положення у просторі, поворот та масштаб. В Unity3D надає можливість використовувати мови програмування C# JavaScript для написання скриптів, які можуть надавати додатковий функціонал або створювати поведінку об'єктів [19]. На сьогоднішній день Unity3D має великий функціонал для VR та AR розробки. AR Foundation дозволяє працювати з платформами доповненої реальності на декількох

платформах в Unity. Однак цей пакет являє собою інтерфейс для розробників Unity, але сам не реалізує ніяких функцій AR. Щоб використовувати AR Foundation, потрібні окремі пакети для цільових платформ, офіційно підтримуваних Unity:

- плагін ARCore XR для Android;
- плагін ARKit XR для iOS;
- плагін Magic Leap XR для Magic Leap;
- плагін Windows XR для HoloLens.

AR Foundation – це набір MonoBehaviour інтерфейсів і API для роботи з пристроями, які відстежують стан і орієнтацію пристрою в фізичному просторі, виявляє горизонтальні і вертикальні поверхні, а саме має систему калібрування, довільне положення і орієнтацію. Мають оцінка середньої температури кольору і яскравості в фізичному просторі та виявляють людські обличчя. Важливим доповненням до пристрою є відстеження 3D об'єктів та створення трикутної сітки, відповідної фізичного простору [20].

Unreal Engine – програма для створення віртуальних об'єктів, що розробляється і підтримується компанією Epic Games. Написана на мові C++ та сумісна з операційними системами: Microsoft Windows, Linux, Mac OS і Mac. Unreal Engine має можливість використовувати системи візуалізації Direct3D, OpenGL, Pixomatic, PowerVR. Приклад реалізації віртуального приміщення у Unreal Engine наведено на рисунку 1.10.

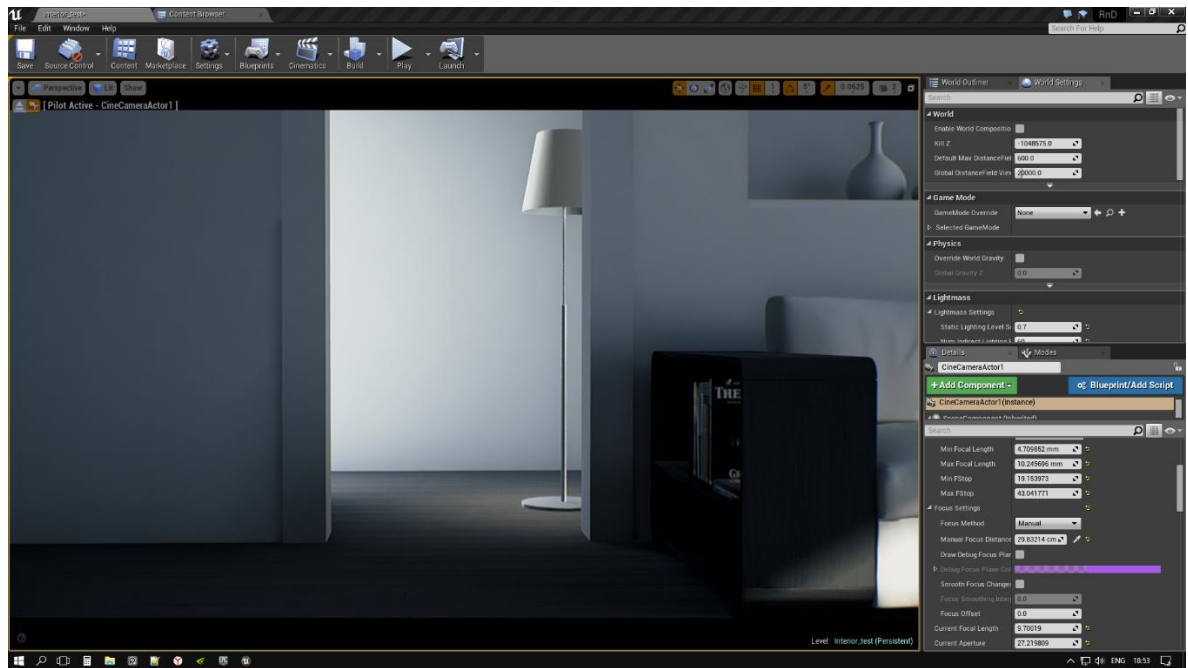


Рисунок 1.10 – Реалізація доповненої реальності у Unreal Engine

Окрім програмування на мові C ++ Unreal Engine має нодову систему програмування Blueprints (візуальне програмування). Система візуальних сценаріїв Blueprints в Unreal Engine – це повна система сценаріїв ігрового процесу, заснована на використанні інтерфейсу для створення елементів ігрового процесу з редактора Unreal Editor [21].

У Blueprints візуалізувати процес виконання можна під час роботи гри в редакторі, що може допомогти з налагодженням сцени та заощадить час. Дані вузлів створені у відповідності з типами змінних. Вхідні висновки оцінюються при виконанні вузла, відстежуючи дрони даних справа наліво, поки остаточний результат не буде вираховано і переданий вузла (рис 1.11).

Вузли з висновками виконання зберігають значення своїх висновків виведення при виконанні, в той час як вузли без висновків виконання (чисті вузли) повторно оцінюють свої висновки кожен раз, коли виконується вузол, підключений до їх висновків [22].

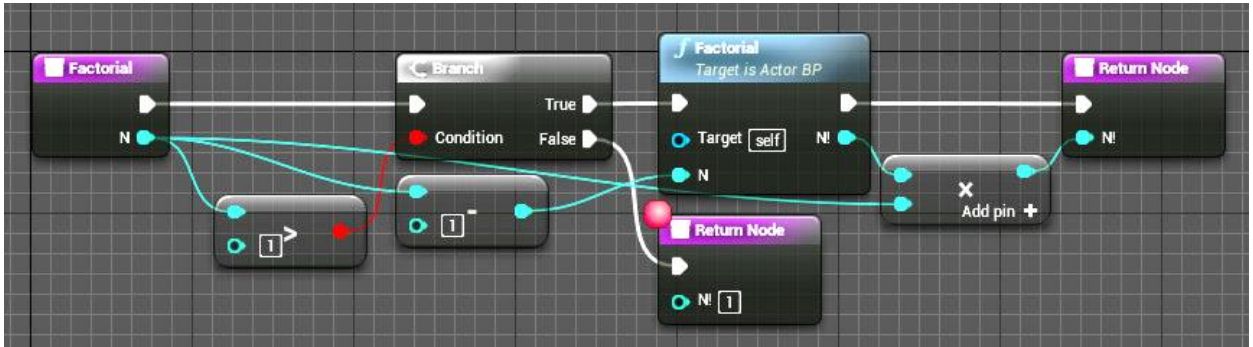


Рисунок 1.11 – Blueprints в Unreal Engine

Ця система надзвичайно гнучка і потужна, оскільки сильно скорочує витрати часу та надає дизайнерам можливість використовувати великий спектр концепцій і інструментів, без знань мови програмування. Також важливим додатком є те, що у інших програм немає можливості використовувати зручне візуальне програмування. Слід зазначити, що команда (ком'юніті) створює власні блюпринти та додаються нові параметри.

Unreal Engine надає рішення для створення віртуальної реальності (VR), доповненої реальності (AR) і змішаної реальності (MR) завдяки вбудованій інтеграції з платформами Oculus VR, SteamVR, Google VR, HoloLens 2, Magic Leap, Windows Mixed Reality. А завдяки підтримці OpenXR дозволяється адаптувати програми до нових пристроїв в майбутньому. Unreal Engine 4 (UE4) тепер пропонує кілька різних способів запуску проекту віртуальної реальності (VR) в залежності від потреб. Unreal Engine на рівні підтримує більшість з представлених на ринку рішень, що позбавляє розробників від необхідності однозначного вибору конкретного VR-шолома на перших етапах проекту. Крім того, в UE4 для VR реалізований високий рівень абстракції, що дозволяє легко впоратися зі зміною пристрою[23] .

Gear VR – окуляри розроблені компанією Samsung. Gear VR розроблені для доповненої та віртуальної реальності та мають не велику ціну на відміну від інших аналогів (рис 1.12). Дані окуляри використовуються здебільш для перегляду відео або віртуальних екскурсій. До недоліків відноситься те ,що

сумісні Gear VR лише із смартфонами та не має сумісності з навушниками [24].



Рисунок 1.12 – Samsung Gear VR

HTC Vive розроблений в результаті партнерства HTC та Valve. Vive має два контролери які допомагають із пересуванням та взаємодією із об'єктами. Дані окуляри розроблені не тільки для перегляду але і для повноцінного занурення у VR ігри. Vive забезпечує відстеження обертання положення шолома і включених контролерів руху (рис 1.13). На сьогоднішній день HTC Vive є одним із найкращих VR гарнітур [25].



Рисунок 1.13 – HTC Vive

1.3 Приклади реалізації віртуальних 3D-моделей виробничих приміщень

Віртуальні прогулянки на сьогоднішній день дозволяють не виходячи із приміщення пройтись по історичним пам'ятникам, музеям або іншими визначним пам'яткам. Компанія Google представила безкоштовну VR-екскурсію по Версальському палацу, яка доступна для власників шоломів віртуальної реальності HTC Vive або Oculus Rift. Вони зможуть детально розглянути всі інтер'єри комплексу, в тому числі Дзеркальний зал і парадну королівську спальню (рис 1.14).

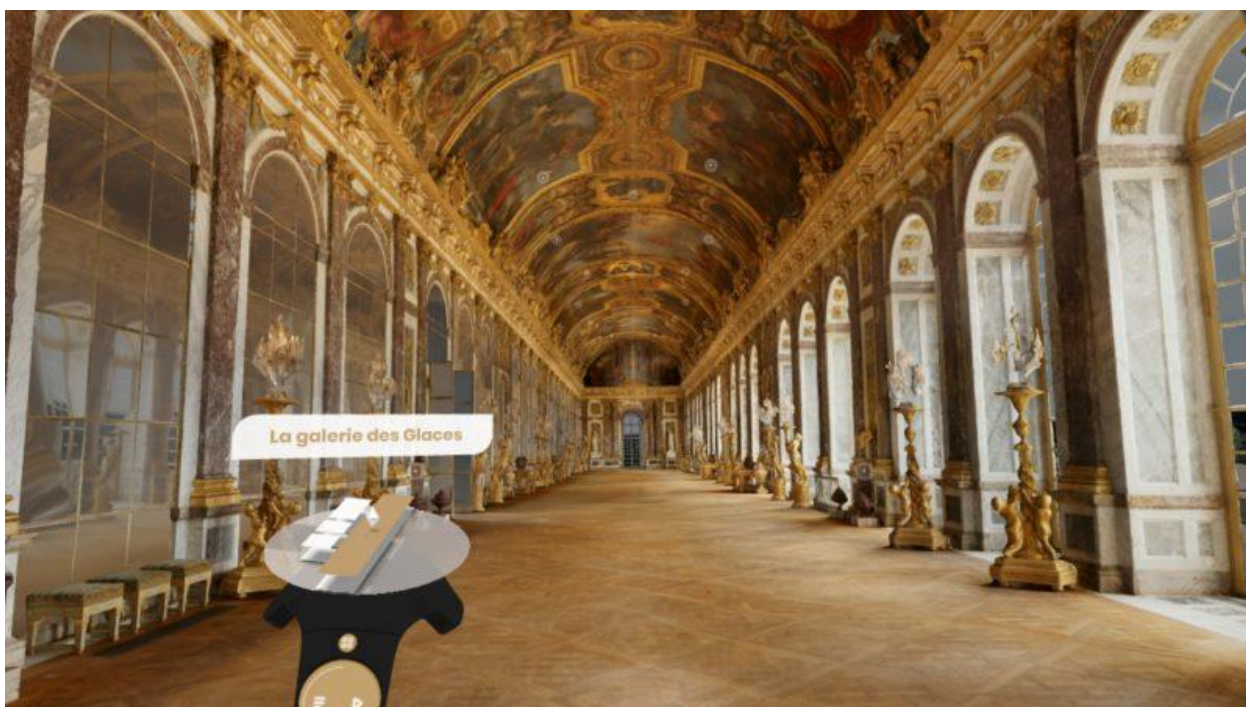


Рисунок 1.14 – Версальський палац

При створенні туру було зроблено 132 тисячі фотографій колишньої резиденції французьких королів, розташованої в передмісті Парижа [26].

Віртуальна реальність також використовується у сфері освіти проект vAcademia дозволяє студентам та викладачам проводити лекції, семінари, практики у віртуальній реальності. Перевагою платформи vAcademia є можливість викладачам проводити заняття зі своїми студентами, організовуючи процес дистанційного навчання, при цьому кожне заняття може бути записано (рис 1.15).



Рисунок 1.15 – Віртуальна аудиторія vAcademia

Інтеграція vAcademia в освітню систему дає можливість поєднувати усталені способи дистанційного навчання з інноваційним підходом навчання у віртуальному світі, планувати і організовувати процес навчання [27].

На замовлення Вінницької міської ради студенти ВНТУ зреалізували соціально-культурний проект «Розробка тривимірних моделей для сьогодення та історичного минулого Вінниці» (рис 1.16). Тривимірна графіка не тільки достовірно відтворює форму об'єкта і його конструктивні особливості, але й точно передає градації кольорів. А це є визначальним при створенні ілюзії об'ємності тривимірного об'єкта на двовимірному екрані. Отож, саме 3-D графікою, яка нині є найреалістичнішою, послуговувались студенти ВНТУ. Працюючи над запропонованим проектом, створили понад 100 моделей [28].

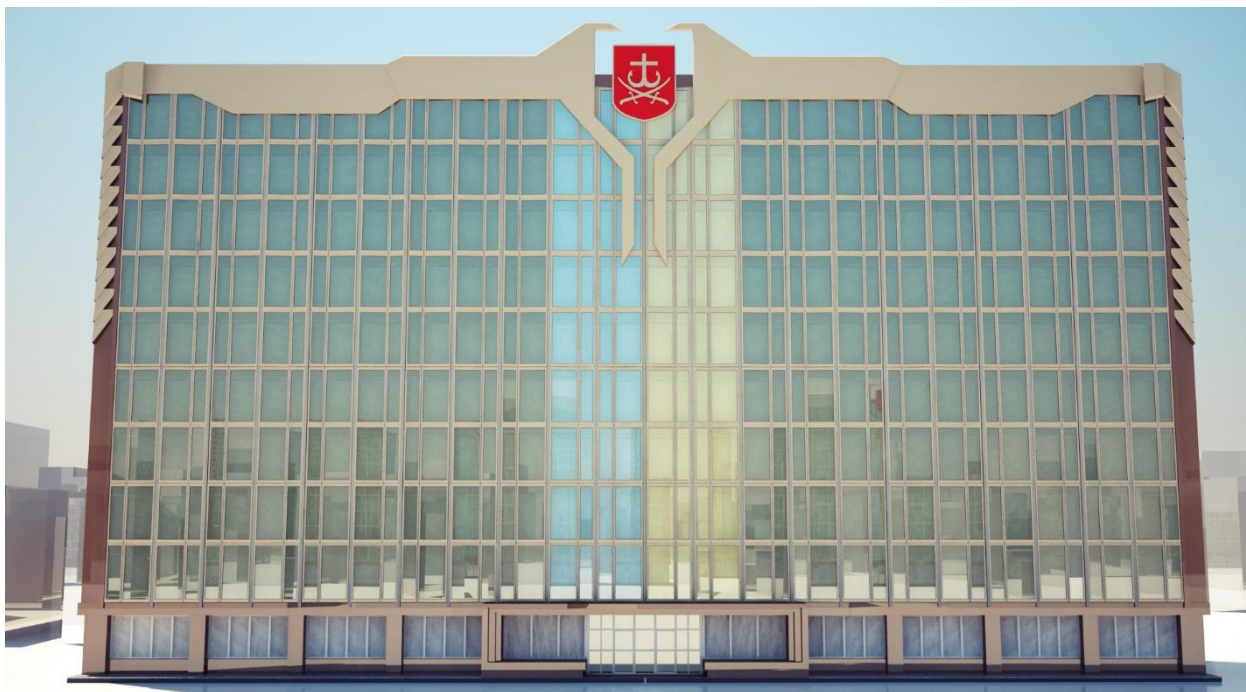


Рисунок 1.16 – Тривимірна модель будівлі

Lecture VR – це програма VR від Immersive VR Education, яка імітує аудиторію у віртуальній реальності, додавши спеціальні ефекти. Приклад аудиторії Lecture VR представлена на рисунку 1.17.



Рисунок 1.17 – Аудиторія Lecture VR

Даний формат дозволяє відтворювати історичні лекції та має можливість підключатися кожному бажаючому [29].

Стартап Surgical Theater створив VR-систему, в якій на основі даних пацієнта можна будувати модель черепа, судин, план конкретної операції. Розширена платформа хірургічної навігації розширює робочий процес хірурга в операційній за рахунок інтеграції і поліпшення існуючої хірургічної навігаційної системи разом з іншими інструментами і технологіями при використанні можливостей Precision VR. (рис 1.18).



Рисунок 1.18 – VR-система Surgical Theater

SNAP революціонує способи виконання складних черепних хірургічних втручань та в інших хірургічних додатках. Платформа дозволяє хірургам застосовувати хірургічні плани для конкретних пацієнтів під час реальних операцій, маючи можливість бачити і знати, що буде далі в хірургічній області. Детальна візуалізація може привести до набагато більш точним хірургічним процедурам, які можуть оптимізувати і скоротити час відновлення пацієнта [30].

Discovery VR дозволяє користувачу переглянути деякі програми Discovery за допомогою шолому віртуальної реальності (рис 1.19). Даний контент підходить для перегляду та наочного навчання в школах або інститутах. Завдяки даній розробці складається цілковите враження присутності при зануренні у глибоководні океани або підйоми на гори [31].



Рисунок 1.19 – Discovery VR

1.4 Висновки до розділу та постановка мети дослідження

В даному розділі проаналізовано розвиток концепцій віртуальної та доповненої реальності. Наведена різниця між віртуальною доповненою та змішаною реальністю та наведені приклади реалізації різних компаній в даній галузі розробки. Розглянуто та обрана актуальна та переважно доступна VR гарнітура. В якості VR буде використатися НСТ VIVE так як у співвідношенні ціни та функціональних можливостей є найбільш актуальним варіантом. Розглянуті та проаналізовані системи створення віртуальної реальності а саме Unity та Unreal Engine. Також розглянуті принцип роботи 3D сканера який спрощує моделювання та відтворення об'єктів. Однак даний метод

відтворення не працює належним чином коли об'єкт лише в розробці та не існує фактично, отже даний метод формування може використовуватися для реконструкції та відтворення реальних фактичних будівель або приміщень, також важливим параметром є вартість сканера. Для виконання магістерської роботи було обрано Unreal Engine, тому, що він має більший функціонал та адаптований із багатьма VR окулярами, також слід зазначити, що він має вбудовані шаблони для підключення гарнітури шолому та має можливість конвертувати та адаптовувати сцени для підключення в Web додатки та має адаптований конвертор сцени із 3D Max.

2 ФОРМУВАННЯ ВІРТУАЛЬНОЇ 3D-МОДЕЛІ ВИРОБНИЧОГО ПРИМІЩЕННЯ

2.1 Системи координат засобів 3D-моделювання та основні операції

Система координат – комплекс визначень, які дозволяють визначати положення і переміщення точки або тіла у просторі. Сукупність чисел визначає положення точки. Найбільш поширеною системою координат є декартова, однак для окремих задач використовуються сферичні, циліндричні та полярні системи координат. У тривимірній графіці за основу прийнята декартова система координат, яка в свою чергу використовуються для різних об'єктів:

- локальний простір ;
- світовий простір ;
- простір вікна перегляду («View space»);
- відсікання простір (Clip space);
- екранний простір (Screen space).

Локальне простір – це простір де об'єкт знаходиться у даний проміжок часу. Після створення об'єкта він буде перебувати в точці $(0,0,0)$ (x,y,z) . Загалом локальний простір показує де знаходиться вибраний об'єкт.

Матриця моделі – це матриця перетворення, яка здійснює масштабування та або обертає об'єкт, щоб помістити його в тому місці орієнтації навколишнього світу, в якому він повинен знаходитися.

Простір вікна перегляду – це те, що люди зазвичай називають камерою OpenGL (іноді також відоме як «простір камери» або «простір поля зору»). Простір вікна перегляду – це результат перетворення ваших координат світового простору в координати, які знаходяться перед очима користувача. Таким чином, це простір, який оглядається з точки зору камери. Зазвичай, це досягається за допомогою комбінації трансляцій і поворотів для переміщення повороту сцени так, щоб деякі елементи трансформувалися в позицію перед

камерою. Дані комбіновані перетворення найчастіше зберігаються в матриці виду, яка перетворює світові координати в координати простору вікна перегляду.

Допоміжна система координат визначає, як буде рухатися, повертатися і масштабуватися об'єкт при завданні відносної трансформації. Відносна трансформація визначає, як повинна змінитися трансформація об'єкта в порівнянні з вихідною, наявної на даний момент. Якщо натиснути правою кнопкою миші по значку тієї чи іншої трансформації, з'явиться діалог Transform Type-In.

Діалог має дві групи полів: Absolute: World і Offset: поточна система координат.

Absolute – абсолютні координати, які отсчитиваются відносно World.

Offset – зсув, в цю групу полів вводяться значення, на які потрібно змінити поточні координати об'єкта, і відраховуються вони не в Світовий системі координат, а в допоміжній, прив'язаною до об'єкта. Так само, допоміжна система координат визначає, як буде трансформуватися об'єкт за допомогою миші.

У 3D Max існує 7 варіантів позиціонування об'єкта по координатам, залежно від того, як потрібно розташувати об'єкт.

View – система координат щодо виду (обраного вікна проекції). Тобто, кожного разу, коли ви переміщується в інше вікно проекції, система координат повертається таким чином, щоб вісь z була спрямована перпендикулярно координатної сітки.

Screen – вісь z спрямована перпендикулярно екрану. Якщо прямо зараз зробити активним вид Perspective, то в інших вікнах проекції осі координат будуть відображатися під кутом.

Parent – координати відраховуються щодо системи координат батьківського об'єкта.

Local – переміщення, обертання і масштабування здійснюється відносно локальної системи координат об'єкту. Це особливо буває зручно при обертанні, коли нам треба повернути довгастий об'єкт навколо власної осі.

Gimbal – подібно Local, але призначене спеціально для Ейлерівського (Euler) обертання, де осі координат не обов'язково можуть бути перпендикулярні один одному.

Grid – якщо у вас є допоміжна сітка, розташована під кутом до основних сіток, то координати можна відраховувати щодо неї.

Pick – можна вибрати будь-який об'єкт, і відраховувати координати щодо нього. Системи координат представлено на рисунку 2.1.

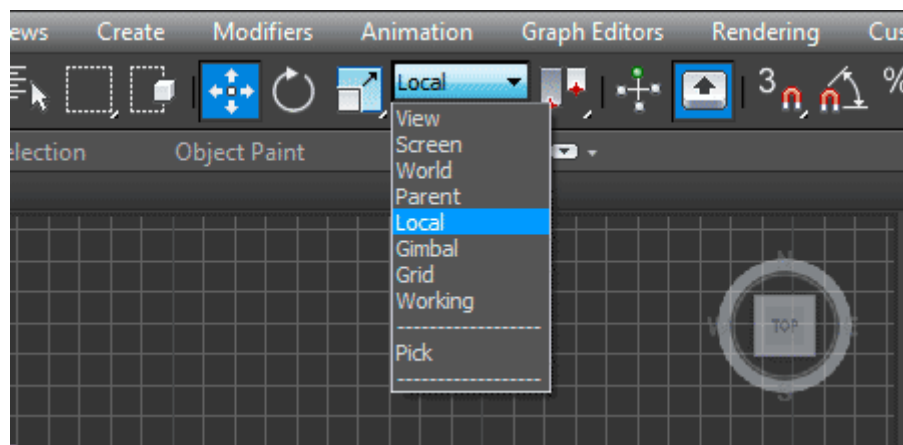


Рисунок 2.1 – Системи координат у 3D Max

2.2 Формування моделей стандартних об'єктів

В процесі 3D моделювання створюються геометричні моделі, тобто моделі, що відображають геометричні властивості об'єктів. Розрізняють каркасні (дротяні), поверхневі і об'ємні (твердотільні) геометричні моделі. Для кожної моделі відомі координати кінцевих точок і вказана їх приналежність ребрам або поверхням (рис 2.2).

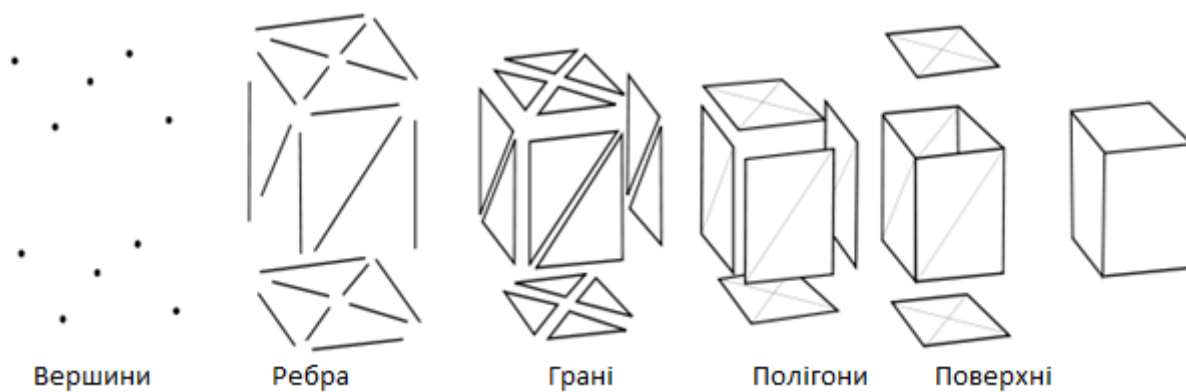


Рисунок 2.2 – Сегментація моделі

Кінцевою метою 3D моделювання є побудова математичних моделей геометрії об'єктів. Моделі можуть використовуватись для проведення досліджень і виробництва деталей. Геометричне моделювання вивчає методи побудови математичної моделі, що описує геометричні властивості предметів навколишнього світу. Моделювання 3D об'єктів базується на диференціальній геометрії та обчислювальній математиці. Для окремих задач будують тверді тіла вони мають певні загальні властивості та розрахувати щільність металу або провести експеримент на міцність конструкції (рис 2.3).

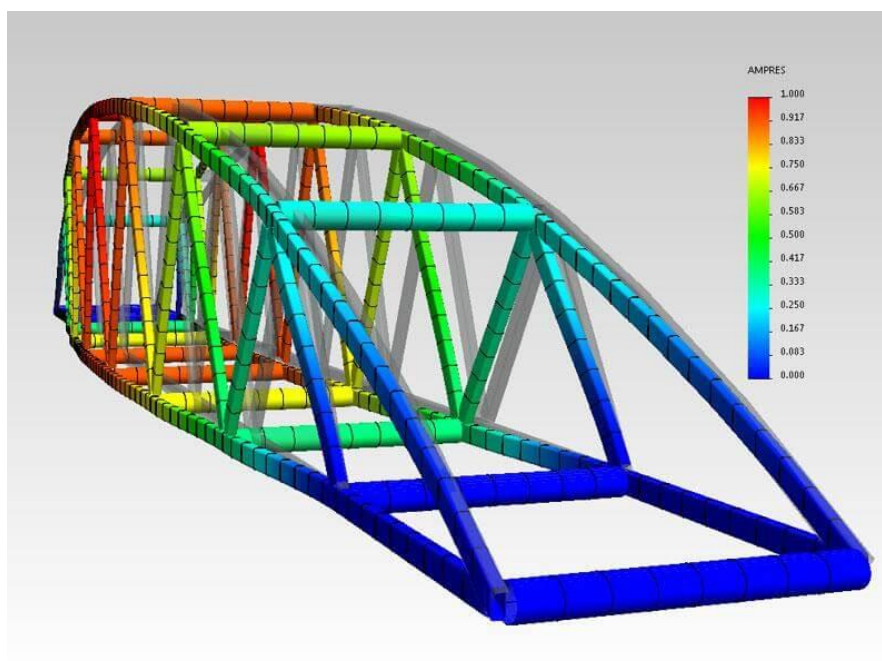


Рисунок 2.3 – Симуляція міцності конструкції

Геометричні об'єкти служать основними елементами математичної моделі геометрії реальних або уявних об'єктів.

У системах каркасного моделювання форму об'єкту представляють у вигляді набору характеристичних ліній (рис 2.4). Дана модель є кресленням форми. Зміна форми здійснюється шляхом зміни положення точок або розміром ребер (сегментів) [32].

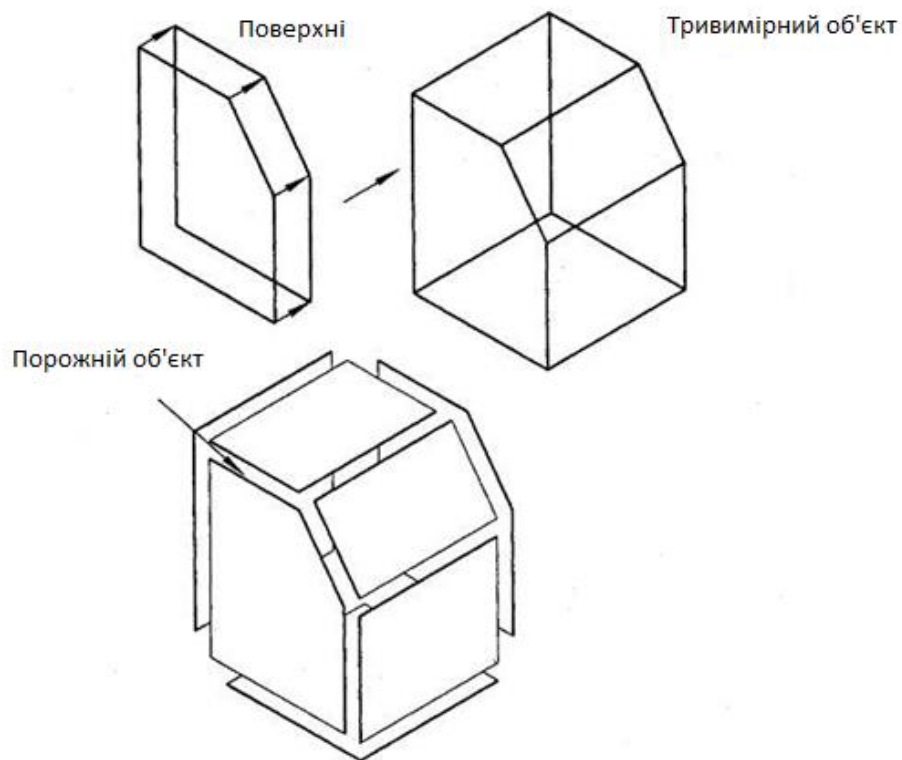


Рисунок 2.4 – Каркасне моделювання

Відомості про зв'язність описують приналежність точок до конкретних кривих. Без цих відомостей неможливо створити геометричну сітку та модель в цілому. Так як для моделювання потрібно постійно описувати криві та точки у просторі системи каркасного моделювання були замінені системами поверхневого і твердотільного моделювання.

Системи твердотільного моделювання (solid modeling systems) призначені для роботи з об'єктами, які складені із замкнутого об'єму(solid). У

системах твердотільного моделювання, на відміну від систем каркасного і поверхневого моделювання, неможливо створювати моделі, які не мають замкнутий контур. Математична модель побудови надає редактору відомості, по яким можна розрахувати об'єм тіла, вагу для різних матеріалів та їх міцність. Приклад твердотільного моделювання представлено на рисунку 2.5.

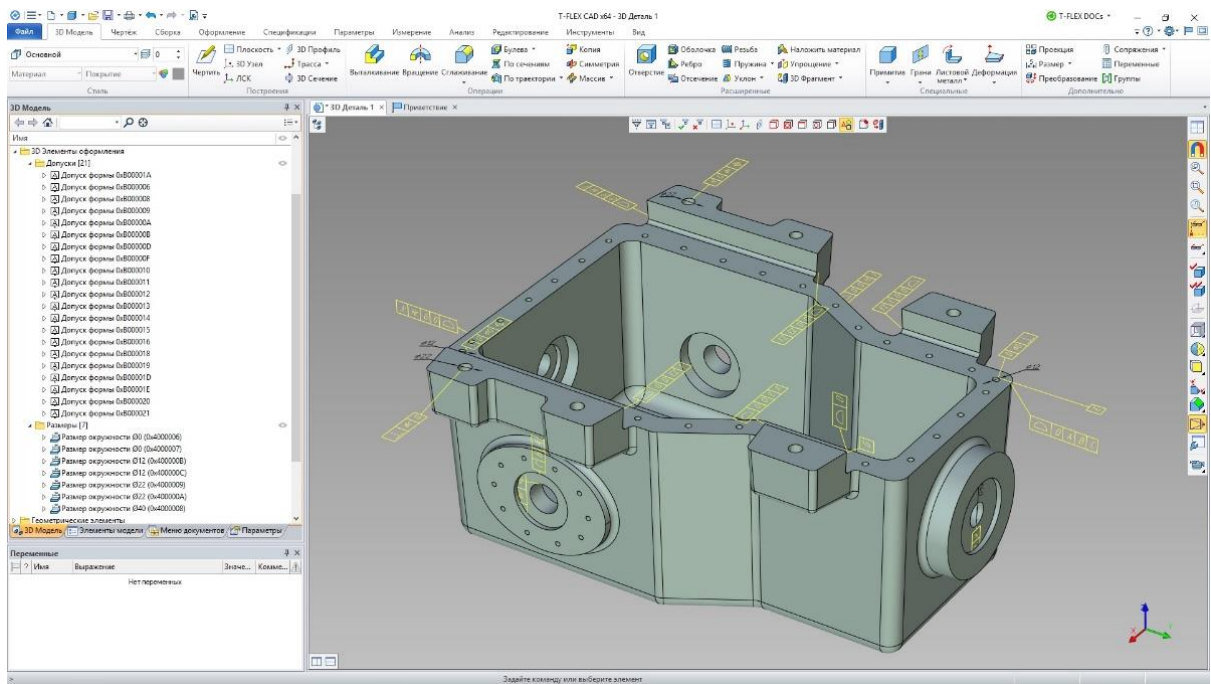


Рисунок 2.5 – Твердотільна модель виробу

Твердотільна система моделювання широко застосовується для фрези верстату з ЧПК, для виготовлення деталі із заготовки. Проте, створення моделі у вигляді замкнутого об'єму вимагає більшої кількості початкових даних у порівнянні з кількістю даних для математичного опису. Такі системи моделювання дозволяють працювати з поверхнями і замкнутими об'ємами одночасно.

Основними принципами моделювання твердотільних моделей є створення простих форм на основі об'ємних примітивів. Також функції додавання і віднімання об'єму – булеві операції. Функція з'єднання двох або більше площин дозволяє створювати об'ємне тіло. Слід зазначити, що у сучасних системах твердотільного моделювання дадані і функції

поверхневого моделювання, що надає більшої гнучкості і функціональності при створенні моделі [32].

У системах поверхневого моделювання (surface modeling systems) математичний опис включає відомості про лінії точки та поверхні(полігони). Поверхнева модель відображає форму деталі за допомогою завдання поверхонь, що обмежують її, наприклад, у вигляді сукупності даних про грані, ребра і вершини. Особливе місце займають моделі деталей з поверхнями складної форми (рис 2.6). Системи поверхневого моделювання використовують для створення моделей зі складними поверхнями [32].

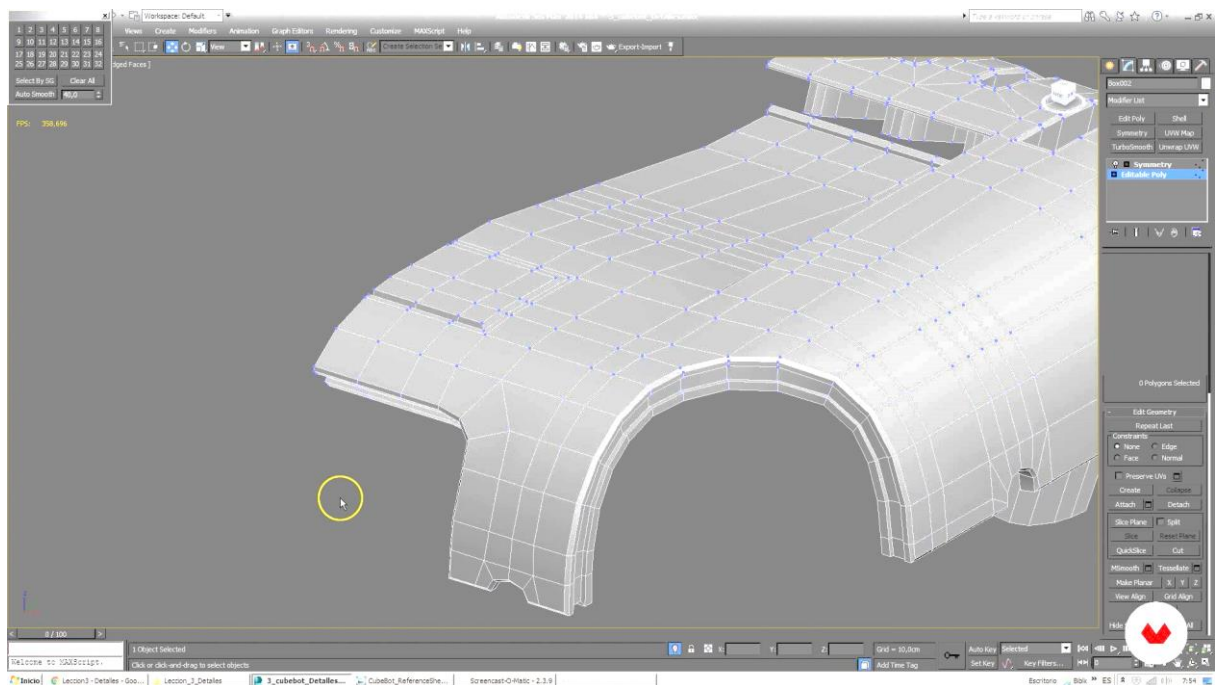


Рисунок 2.6 – Поверхневе моделювання об'єкту

Сплайн формується за допомогою ділянок функцій кожна з яких є поліномом. Найпростішим сплайном є ламана лінія. Вона складається з відрізків, які послідовно з'єднують задані точки. Значення параметру в кожній наступній точці повинно бути більше значення параметру в попередній точці $t_i < t_{i+1}$. Радіус-вектор ламаної визначають виразом:

$$\mathbf{r}(t) = \mathbf{p}_i(1-w) + \mathbf{p}_{i+1}w, \quad w = \frac{t-t_i}{t_{i+1}-t_i}, \quad t_0 \leq t \leq t_n, \quad (2.1)$$

де $t_i \leq t \leq t_{i+1}$. Параметр w називають місцевим параметром на ділянці кривої між точками \mathbf{p}_i і \mathbf{p}_{i+1} . Перша похідна ламаної лінії в точках \mathbf{p}_i розривається по довжині і по напрямку. Параметр ламаної лінії змінюється в одномірному просторі. У цьому просторі для визначення параметру t можливо використовувати будь-яку систему координат. Для параметру можна використовувати систему координат, де його значення в точці \mathbf{p}_i дорівнює номеру точки: $t_i = i$. Таку параметризацію називають рівномірною, а параметрична довжина ламаної в такому випадку дорівнює числу точок мінус один. Ламана може бути замкнутою, у цьому випадку перша характеристична точка одночасно є і останньою. Ламана має ряд корисних властивостей: її точки легко обчислювати, легко редагувати (вставити нову точку, видалити або зрушити існуючу), легко розрізати на частини, кожна з яких також буде ламаною лінією.

Завдяки простоті завдання і можливості зручно маніпулювати формою, криві Без'є знайшли широке застосування у комп'ютерній графіці для моделювання гладких ліній (рис 2.6). Оскільки криву Без'є повністю визначає опукла оболонка з опорних точок, останні можуть бути відображені і використані для наочного керування формою лінії.

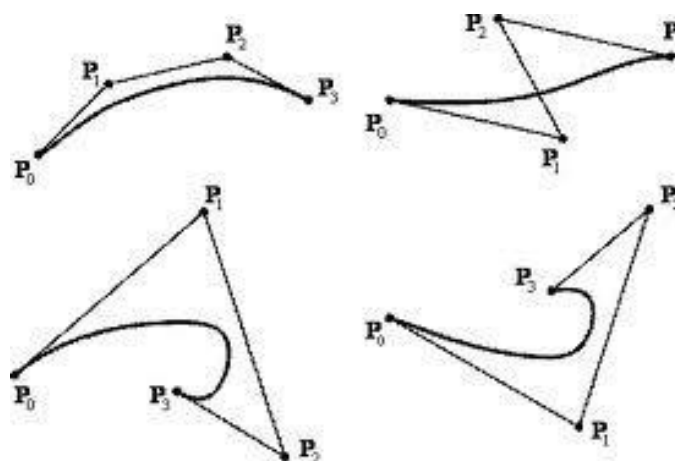


Рисунок 2.6 – Крива Без'є

Крім того, афінні перетворення кривої (перенос, масштабування, обертання) також легко можуть бути здійснені шляхом застосування трансформацій до опорних точок. Наявність опуклої оболонки значно полегшує задачу про точки перетинання кривих Без'є: якщо не перетинаються опуклі оболонки, то не перетинаються й самі криві. Координати сплайнів є поліномами параметру кривої. У загальному випадку сегменти таких кривих описують векторними функціями виду

$$\mathbf{r}(t) = \sum_{i=0}^n t^i \mathbf{a}_i . \quad (2.2)$$

Суть векторів \mathbf{a}_i , полягає тільки в тому, що їх компоненти є коефіцієнтами поліномів для компонентів результуючого вектору. Інакше кажучи, вектори \mathbf{a}_i , не несуть геометричної інформації кожний окремо. Але якщо перегрупувати праву частину, то можна одержати вираз, у якому векторні величини мають певний геометричний зміст. Це можна спостерігати на прикладі інтерполяційної формули Лагранжа, де радіус-вектор кривої представлено у вигляді:

$$\mathbf{r}(t) = \sum_{i=0}^n L_i(t) \mathbf{p}_i, \quad (2.3)$$

Векторні коефіцієнти \mathbf{p}_i були радіус-векторами характеристичних точок кривої, а залежність кривої від параметру була зосереджена в скалярних функціях $L_i(t)$, індивідуальних для кожної характеристичної точки \mathbf{p}_i . Криву Без'є описують формулою:

$$\mathbf{r}(t) = \sum_{i=0}^n B_i^n(t) \mathbf{p}_i = \sum_{i=0}^n \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i} \mathbf{p}_i, \quad 0 \leq t \leq 1, \quad (2.4)$$

де \mathbf{p}_i – радіус-вектори точок, а n – число цих точок мінус один

Для побудови кривої, що проходить через задані точки, використовують складові криві Без'є. Крива Без'є завжди торкається своєї характеристичної ламаної на кінцях. У загальному випадку в точках з'єднання перша похідна стрибком мінятиме своє значення, а всі вищі похідні зазнаватимуть розрив. Для безперервності першої похідної складової кривої Без'є потрібне торкання і рівність довжин крайніх ділянок характеристичних ламаних у місці стикування. Параметрична довжина складової кривої Без'є дорівнює сумі параметричних довжин частин, тобто числу цих частин. Якщо вершини кривої Без'є лежать на одній прямій, крива перетворюється у відрізок прямої. Відрізком прямої лінії є і крива Без'є першого ступеня.

З декількох кривих можна побудувати складну криву – найбільш загальний тип кривої. Вони характерні тим, що перша похідна радіус-вектору в точках стикування терпить розрив або по довжині або по довжині та напрямку. Розглянуті вище сплайнові складові криві формували з кривих одного типу, і в них у загальному випадку перша похідна радіус-вектору терпіла розрив по довжині. Криві, що утворять складову криву, називатимемо сегментами. При побудові складної кривої повинні бути виконані певні умови: початок кожного наступного сегменту повинен збігатися з кінцем

попереднього сегменту. Якщо сегменти складної кривої стикуються не гладко, то складна крива буде мати злами. У загальному випадку в місцях стику сегментів похідні складної кривої терплять розрив по довжині й напрямку. Замкнуту складну криву називатимемо контуром. Для контуру початок першого сегменту повинен збігатися з кінцем останнього сегменту. Складна крива. Нехай складна крива містить n сегментів $r_i(w_i)$, $w_i \in [w_{i \min}, w_{i \max}]$, $i = 1, 2, \dots, n$. Початкове значення параметру t складної кривої приймемо рівним нулю. Параметричну довжину складної кривої приймемо рівній сумі параметричних довжин кривих, що її складають

$$t_{\min} = 0, \quad t_{\max} = \sum_{i=1}^n (w_{i \max} - w_{i \min}). \quad (2.5)$$

При обчисленні радіус-вектору складової кривої спочатку необхідно визначити той сегмент, на який потрапимо, рухаючись по параметру. Потім потрібно визначити відповідне значення власного параметру цього сегменту та за допомогою його значення обчислити радіус-вектор сегменту або його похідної.

$$\sum_{i=1}^{k-1} (w_{i \max} - w_{i \min}) \leq t \leq \sum_{i=1}^k (w_{i \max} - w_{i \min}) \quad (2.6)$$

Радіус-вектор складової кривої визначають рівнянням

$$\mathbf{r}(t) = \mathbf{r}_k \left[w_{k \min} + t - \sum_{i=1}^{k-1} (w_{i \max} - w_{i \min}) \right] = \mathbf{r}_k(w_k), \quad 0 \leq t \leq t_{\max} \quad (2.7)$$

де параметр k -го сегмента дорівнює

$$w_{k \min} + t - \sum_{i=1}^{k-1} (w_{i \max} - w_{i \min}). \quad (2.8)$$

Лініями можна описати окремі геометричні властивості об'єктів. Криві лінії використовують для створення поверхонь і тіл. Лінії описують за допомогою скалярних величин, векторів, чи інших ліній. Лінії можуть бути просторовими і двомірними. Криві можуть бути замкненими або розімкненими. Для замкнутої кривої двомірна лінія описується векторною функцією. Двомірні вектори є базисними векторами деякої декартової системи координат двомірного простору. У багатьох випадках двомірні лінії аналогічні просторовим лініям. У загальному випадку лінію можна представити у вигляді системи двох рівнянь, яким задовольняють координати радіусвектору точок лінії [32].

Перший етап розробки представляє собою, збір інформації про об'єкт, пошук креслень або фотографій об'єкту.

Другий крок у розробці – це створення моделі. На даному етапі відбувається процес створення від загальної до кінцевої форми, а саме створюється базова форма і дороблюється деталізація об'єкту. Використовують такі програми як, 3D MAX Blender 3D Coat або інші програми в залежності від встановлених задач.

Підготовка текстур і розробка матеріалів. Для того щоб карти текстур лягали правильно на модель, без артефактів і дефектів використовують розгортку. Розгортка зазвичай готується в тій же програмі, що і для моделювання, але існують і сторонні програми, призначені для розгортки такі як UV layout. Після створення розгортки моделі йде процес текстурування. Текстурування накладення зображення на модель для додання їй кольору, забарвлення або рельєфу. Текстури дозволяють додати поверхні ті властивості об'єкта, які вимагали б високий полігонаж і навантажували б систему.

Заключний етап створення моделі – це рендер або імпорт в ігровий двигун. На перших етапах рендера необхідно працювати з низькою розподільною здатністю для швидкого виявлення недоліків і перегляду готового зображення. Для отримання найкращого результату рендер

необхідно виробляти по проходам, тобто рендерити окремо задній фон, передній фон, відображення, світло, тіні і т.д. для того, щоб отримати якомога більше контролю для пост обробки. На рисунку 2.7 представлено алгоритм моделювання об'єкту.



Рисунок 2.7 – Етапи створення моделі

3Ds Max є редактором тривимірної графіки (рис 2.8). Починаючи з 2005 року, програма випускається під звичним вже ім'ям Autodesk 3Ds Max [33].

3D Studio Max дозволяє редагувати, перетворювати і модифікувати(скручувати, згинати, згладжувати, знімати фаски). Використовуючи ці інструменти можна створювати об'єкти будь-якої складності.



Рисунок 2.8 – 3D Max

3D Max вважається професійним інструментом архітекторів і дизайнерів, які займаються моделюванням та візуалізацією інтер'єрних та екстер'єрних рішень. 3D Max дозволяє виконувати багато функцій починаючи від простого моделювання до анімованими відео або розробкою ігрового персонажів. Даний пакет тривимірної графіки має багато модулів, які розширюють його функціонал: система частинок, механізм розрахунку фізики, симуляція поведінки твердих і м'яких тіл, симуляція вогню, диму та інших ефектів [34].

Основною функцією програми є створення та редагування 3D графіки. 3D Max надає наступні способи проектування:

- полігональне моделювання яке є найпоширенішим та використовується для моделювання складних об'єктів;
- моделювання на базі примітивів, стандартних геометричних фігур;
- моделювання на основі ліній.

Моделювання на основі примітивів. 3Ds Max містить вбудовану бібліотеку стандартних об'єктів, так званих примітивів. Загалом створення моделей починається з них, після чого до них застосовують різноманітні

модифікатори. Стандартні геометричні примітиви наведені на рисунку 2.9.

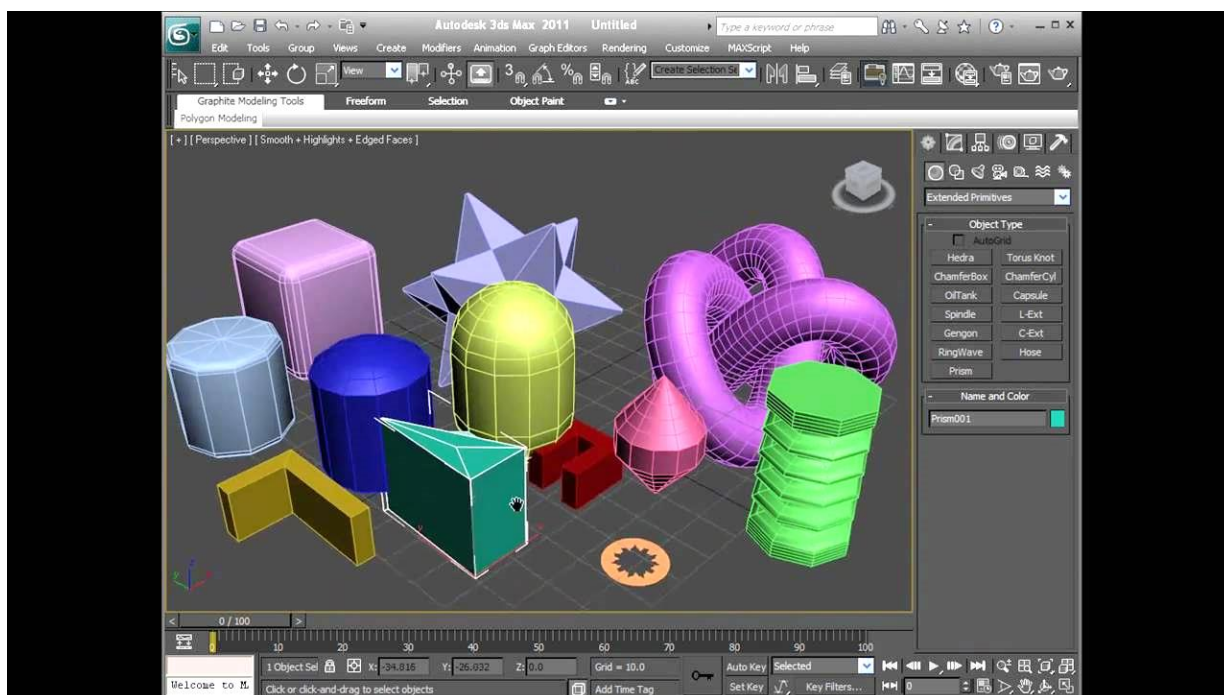


Рисунок 2.9 – Примітиви 3Ds Max

Полігональне моделювання використовується для розробки моделей складної форми. Процес моделювання починається з примітивної форми та встановлення її реальних габаритів, після цього форма конвертується в режим додавання полігонів (Edit Poly), що представлено на рисунку 2.10 .

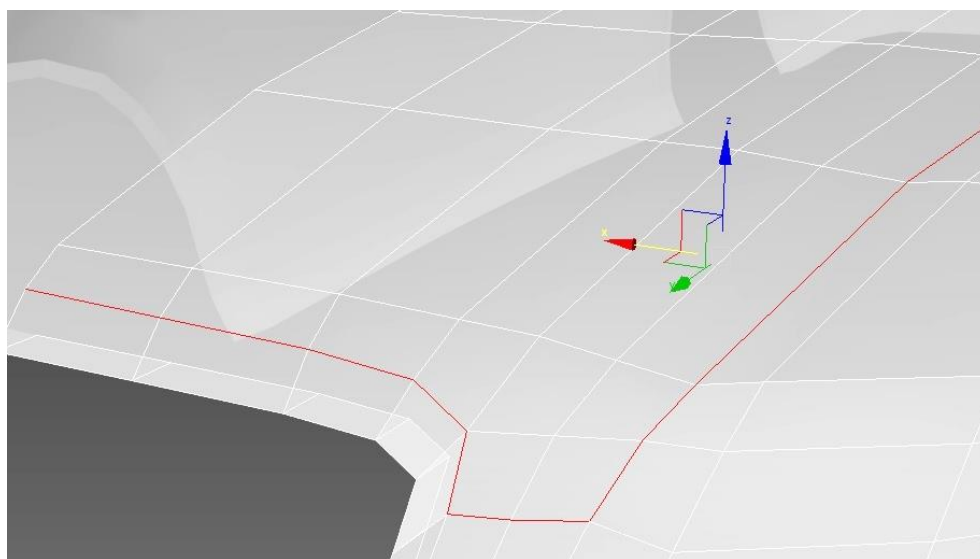


Рисунок 2.10 – Полігональне моделювання

Прикладами операцій, які викликають параметри, є: з'єднання об'єктів між собою за допомогою одного з модифікаторів Edit. При цьому використовується термін підоб'єктів відноситься до чогось, що можна вибрати і маніпулювати. Загальновідомим прикладом підоб'єкту є одна з граней, що утворюють каркас. За допомогою модифікатора Edit Mesh можна вибрати підоб'єкти, наприклад, грань, після чого її можна переміщати, обертати, видаляти або видаляти (рис 2.11). Прикладами підоб'єктів є:

- вершини;
- ребра об'єктів;
- поверхні об'єктів;
- гізмо і центри модифікаторів.

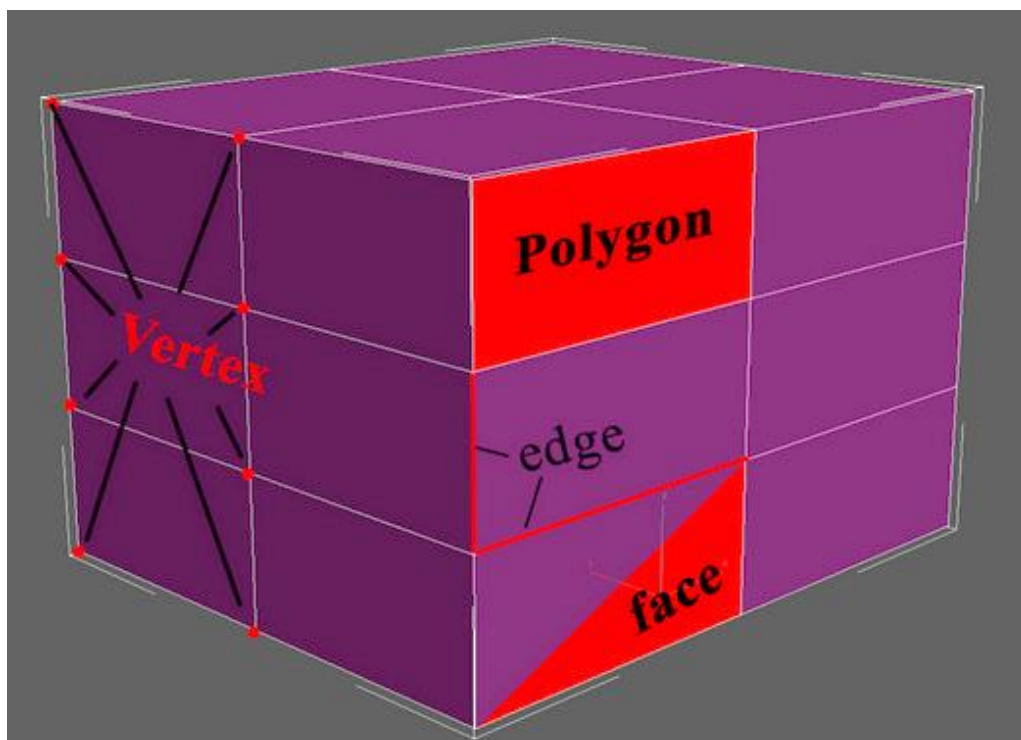


Рисунок 2.11 – Приклади підоб'єктів в Edit Mesh

Модифікатори є головним інструментом моделювання, оскільки користувач управляє порядком застосування модифікаторів. Вплив модифікатора на об'єкт є постійним незалежно від розташування об'єкта.

З метою управління розмірами сцени і швидкістю реакції корисно

зберігати параметричні об'єкти з мінімальною сегментацією і збільшувати її тільки в разі потреби.

Параметри згладжування керують автоматичним додаванням груп згладжування до об'єкта. Деякі об'єкти, наприклад Torus, забезпечують зручні опції для згладжування, які важко здійснити непараметричним способом. Згладжування для конкретних граней можна зробити допомогою модифікаторів EditMesh або Smooth .

Меню Modifiers (Модифікатори) є більш розширеним списком і дозволяє призначати модифікатори, не переходячи на вкладку Modify (Зміна) командній панелі. Перед присвоєнням параметричного об'єкту модифікатора необхідно виділити сам об'єкт, до якого потрібно застосувати модифікатор. При цьому в меню будуть активні тільки ті модифікатори, які можуть бути застосовані до даного примітиву. До одного об'єкту можна застосувати кілька модифікаторів послідовно. Усі застосовані модифікатори відображаються списком у стеці модифікаторів знизу вверху.

Autodesk 3D Max сумісна з багатьма модулями моделювання світла, використовуваних матеріалів і різних ефектів. Додаток надає можливість гнучкого управління настройками, включаючи експозицію, глибину різкості, і багато іншого.

Вікно Material Editor (редактор матеріалів) в 3Ds Max реалізовано по вузловому принципу, тобто кожна функція винесена в окреме діалогове вікно, за рахунок чого управляти матеріалами легко і зручно. У числі візуалізаторів існують такі модулі, як Arnold, V-Ray, Mental Ray, RenderMan, FinalRender, Luxrender і багато інших [35].

3Ds Max підтримує і функціонує з файлами різних програм. Є можливість експорту та імпорту проектів з інших і в інші додатки.

Blender – це безкоштовне програмне забезпечення для створення і редагування тривимірної графіки (рис 2.12). Дана програма має відкритий вихідний код, доступність та функціонал для візуалізації, анімації, створення комп'ютерних ігор і навіть скульптинга. Також важливим функціональним

рішенням є Game Blender який має увесь функціонал для створення не великих проектів для ігор, має движок динаміки та колізії. Також він дозволяє створювати окремі real-time додатки. Важливою особливістю пакету Blender є його невеликий розмір в порівнянні з іншими пакетами для 3D-модельовання [36].



Рисунок 2.12 – Blender

Blender використовує Python, який дозволяє редагувати інтерфейс, додавати нові модифікатори та додатки для програми. Даний програмний продукт доступний на різних операційних системах ОС Windows, GNU / Linux і Mac OSX. 3D модельовання, як і в 3D Max створюється на основі примітивів (рис 2.13), полігонів, NURBS-кривих, кривих Безьє, метасфер, булевих операцій, Subdivision Surface і базових інструментів для скульптінга.

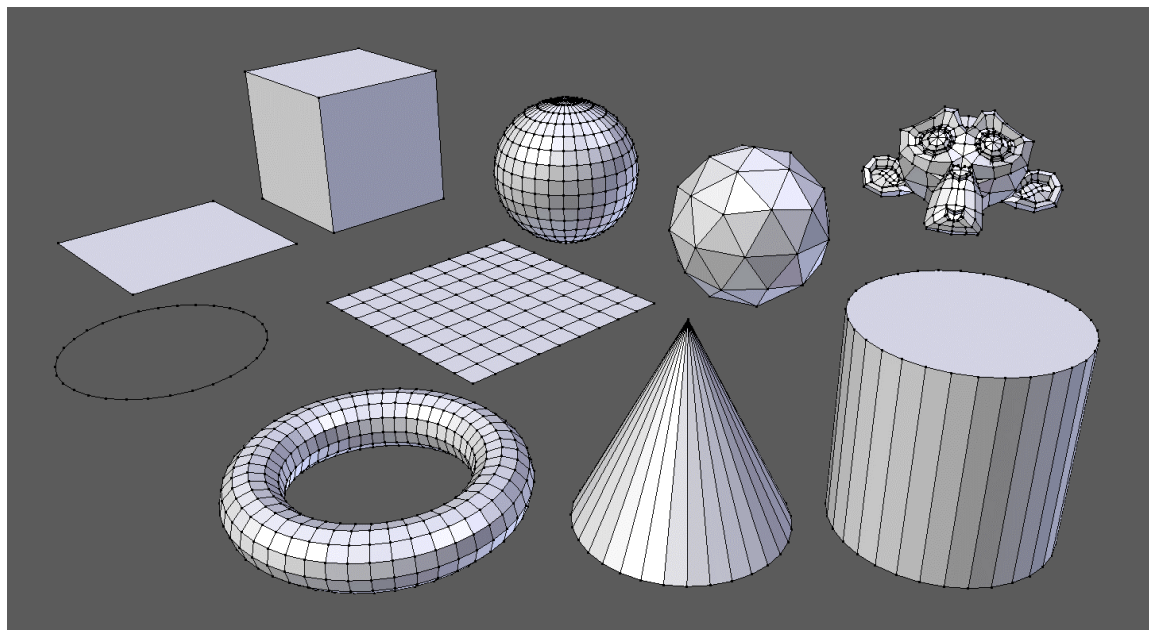


Рисунок 2.13 – Примітиви blender

Анімація у Blender має великий спектр функцій рігінг (скелетна анімація), інверсна кінематика, сіткова деформація, обмежувачі, ключові кадри, редагування вагових коефіцієнтів вершин, нелінійна анімація. Також реалізована динаміка твердих і м'яких тіл, а також анімація частинок.

Текстурування і набори шейдерів. Програма дозволяє накладати кілька текстур на один об'єкт, і оснащена рядом інструментів для текстурування, включаючи UV-мапінг і часткове настроювання текстур. Використання шейдерів додає гнучкості в роботі з матеріалами [37].

Можливість малювання. Blender надає можливість створювати начерки різними типами кистей прямо у вікні програми. Поточне призначення такої функції – допомога у створенні 2D анімації, для чого ця функція також оснащена можливістю гнучкого налаштування, зокрема, роботи з шарами.

Пакет оснащується декількома вбудованими інструментами візуалізації, а також підтримує інтеграцію з різними зовнішніми рендерами.

Blender Cycles – це найперший пакет візуалізації програми, він використовується за умовчанням. Універсальні вбудовані механізми рендерінга і інтеграція з зовнішнім рендерер YafRay, LuxRender і багатьма іншими [38].

Eevee – це новий рендер в реальному часі. Він може працювати як в якості фінального рендера, так і в якості движка, керуючого попереднього перегляду в реальному часі при створенні об'єктів [39].

Підтримується велика кількість функцій, ось деякі з них:

- шейдер Principled BSDF;
- освітлення навколишнього середовища і HDRI;
- відображення і заломлення в екранному просторі;
- непрямий світло через світлові зонди;
- м'які і контактні тіні;
- підповерхневе розсіювання і рендеринг обсягів;
- глибина різкості, розмиття руху камери, блюм.

Порівняння двох систем рендерінгу у Blender показано на рисунку 2.14.

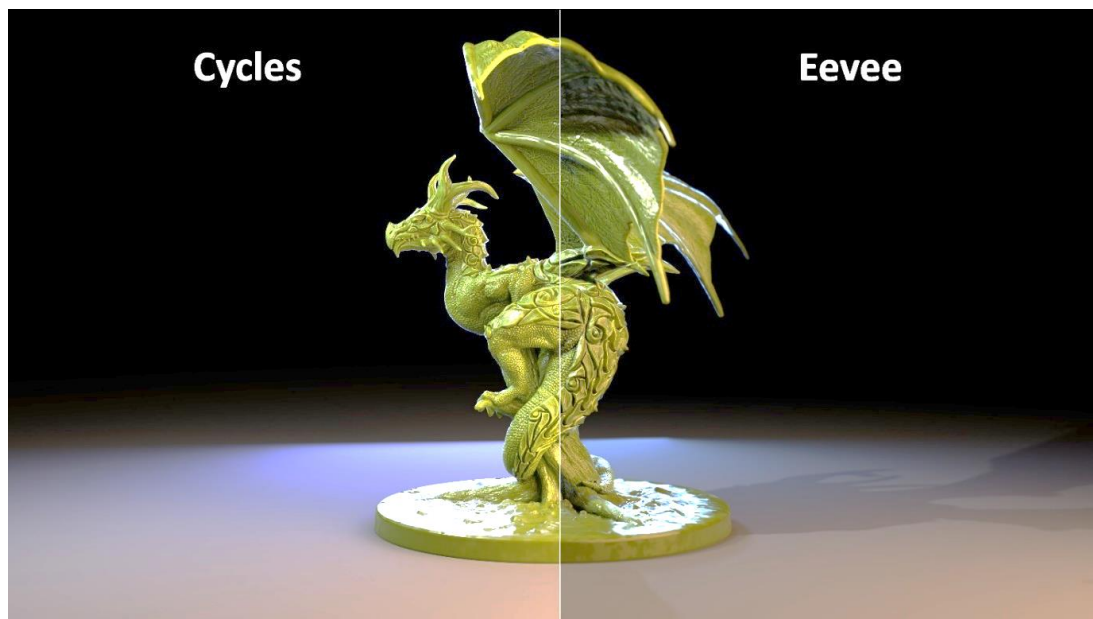


Рисунок 2.14 – система рендерінгу Eevee

2.3 Рендерінг сформованих моделей

Освітлення сцени, є одним із найважливіших аспектів в створенні сцени. Реальність зображення об'єкта залежить від матеріалу, що використовується для накладання на об'єкт (дерево, пластмаса, сталь).

Візуалізація в контексті тривимірної графіки – це процес створення та прорахунку матеріалів і освітлення на геометрії. Найбільш актуальною технікою візуалізації є трасування променів, оскільки воно імітує природне відбиття, заломлення та затінення світла за допомогою 3D поверхонь. У процесі візуалізації променів прослідковуються промені від камери до сцени, щоб визначити площу кінцевого зображення (рис 2.15).

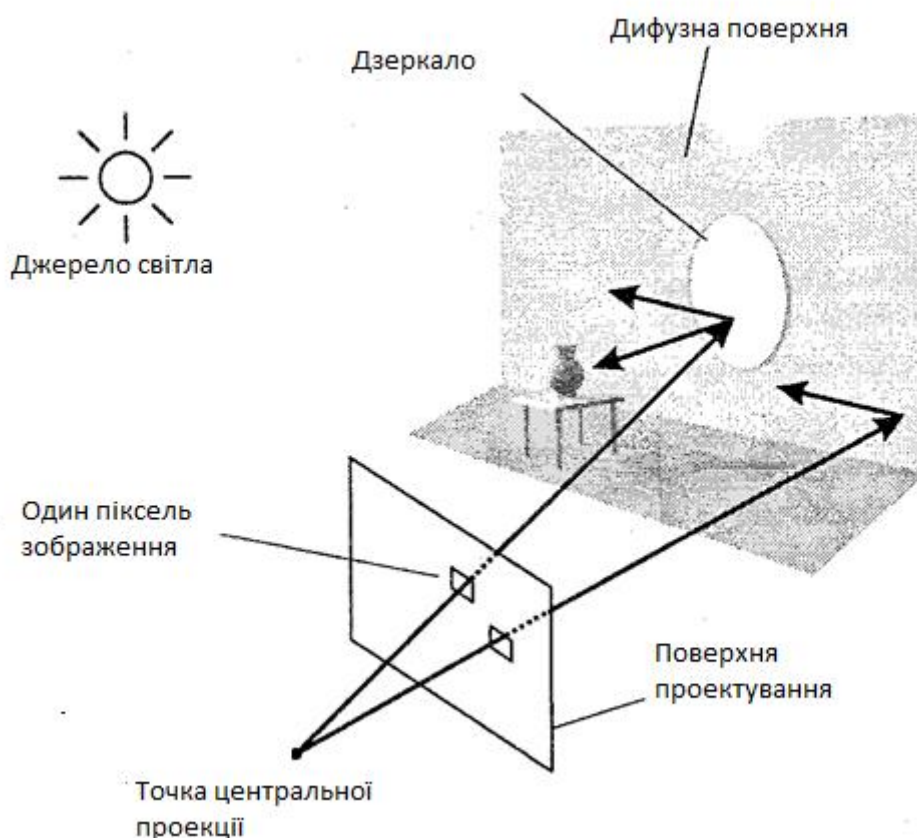


Рисунок 2.15 – Прорахунок освітлення

Основним завданням будь-якої програми рендеринга є обчислення освітленості і кольору довільної точки тривимірної сцени. В даному випадку у системі V-Ray було реалізовано 4 важливі компоненти .

Першим компонентом є освітлення об'єктів від джерел світла, які перебувають у видимості, коли об'єкт і джерело можна з'єднати прямою лінією, облік просторових розмірів джерела світла дозволяє отримувати краї

тіней об'єктів. До цього також відноситься визначення загасання світла з відстанню. В розрахунках освітленості використовується закон квадратичного загасання інтенсивності.

Другий компонент освітленості об'єктів визначається дзеркальним відображенням. Для цього був розроблений метод трасування променів – ray tracing method. Однак недоліком цього методу було те, що стеження за світлом починалось від камери до першої поверхні елемента і в залежності від поверхні промінь відбивався від неї, що приводило зайвої кутоватості об'єктів. Надалі була розроблена модифікація distribution ray tracing. Суть DRT в тому, що при кожному перетині трасуючого променя з поверхнями вздовж його траєкторії, з кожної точки перетину будується не один, а кілька променів.

Третя компонента освітленості об'єкта розраховує дифузні відбивання світла навколишніми об'єктами. На першому проході рендерінгу виконується трасування фотонів від джерел світла до поверхонь, і створюються карти для них. На другому проході виконується зворотне трасування променів від камери, а фотонні карти використовуються для розрахунку дифузійної освітленості точок перетину променів зворотного трасування з поверхнями.

Четверта компонента освітленості займається спеціальним випадком і розраховує світлові ефекти, що виникають в результаті фокусування через заломлення або відбиття променів світла в деякій області поверхні (caustic ефект освітлення). Розрахунок caustic-ефектів освітлення може бути виконаний методом фотонних карт, але при цьому потрібно локальна фотонна карта тривимірної сцени [5].

Даний додаток сумісний із модулями світла та матеріалами і дозволяє керувати експозицією, налаштування фільтри, редагувати баланс білого, давати ефект туману та розсіювання ньому присутні алгоритми для прорахунку глобального світла. Слід відзначити що усі налаштування можливі завдяки інтерфейсу, однак розробник надає можливість використовувати загальні скрипти для загрузки або збереження мап світла або звернення до окремих функцій прорахунку світла.

Матеріали створюються на підставі карт текстур, згорток координат, зашумлення, параметрів растрових, структур, анімації [5]. На рисунках 2.16-2.17 представлені вікна редагування матеріалів та основні властивості базового матеріалу.

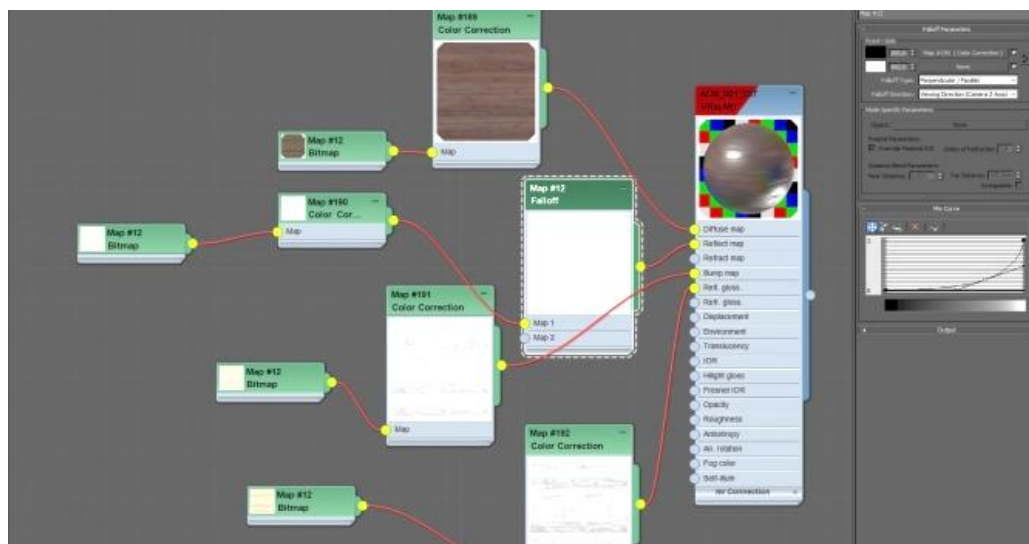


Рисунок 2.16 – Вікно редактора матеріалів

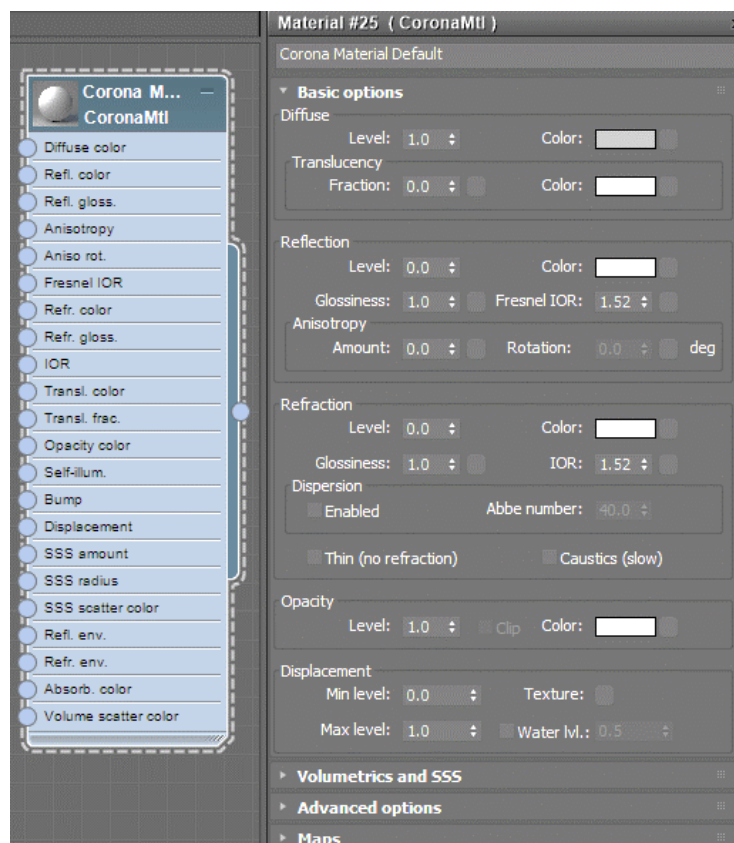


Рисунок 2.17 – Властивості матеріалів

Проекційні координати вказують, як необхідно розмістити на поверхні об'єкта растрові карти текстур, що входять до складу матеріалу. Вибір системи координат залежить від характеру геометричної мозаїки, до якої призначається матеріал.

Існує 7 видів проекційних координат:

- lofted – вбудовані для об'єктів на основі перетину;
- object Created – створювані разом з об'єктами;
- planar – плоска;
- spherical – сферична;
- box – прямокутна тривимірна;
- cylindrical – циліндрична;
- shrink wrap – обтягуюча.

Реалізуються ці способи за допомогою модифікатора UVW map, який також дозволяє визначити розмір текстури та встановити кількість повторень (рис 2.18). Для складових матеріалів є розділ по каналам щоб спростити управління тою чи іншою текстурою.

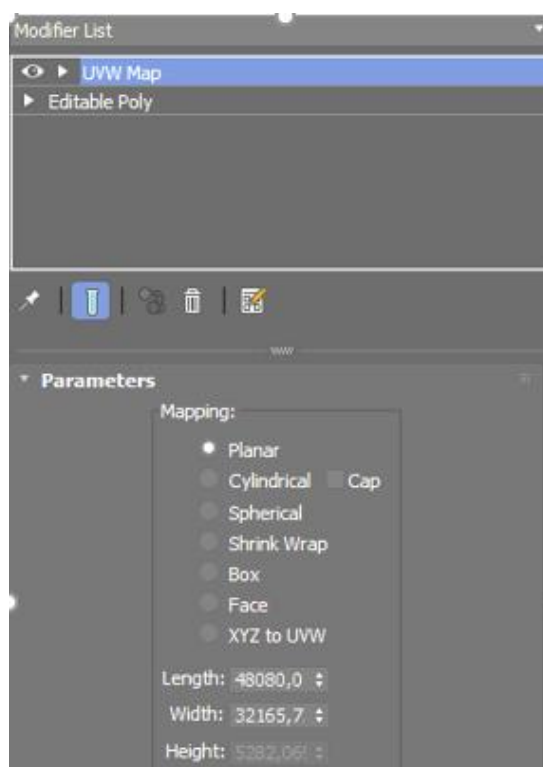


Рисунок 2.18 – Список модифікаторів

У 3D MAX в якості систем архітектурної візуалізації використовуються V-ray та Corona Render системи, які є найбільш поширеними та оптимізованими для даної сфери діяльності. Corona Renderer – це модуль візуалізації, доступний для Autodesk 3ds Max. Corona Renderer була розроблена у 2009, на сьогоднішній день компанія Chaos Group викупила даний проект, що дало можливість для подальшого розширення функціональних модулів. Цей графічний движок використовується з метою створення тривимірних моделей архітектурних об'єктів, проте предметна візуалізація corona render теж можлива.

Початок роботи з даним модулем починається налаштування та вибору Corona Renderer в якості системи візуалізації (рис 2.19).

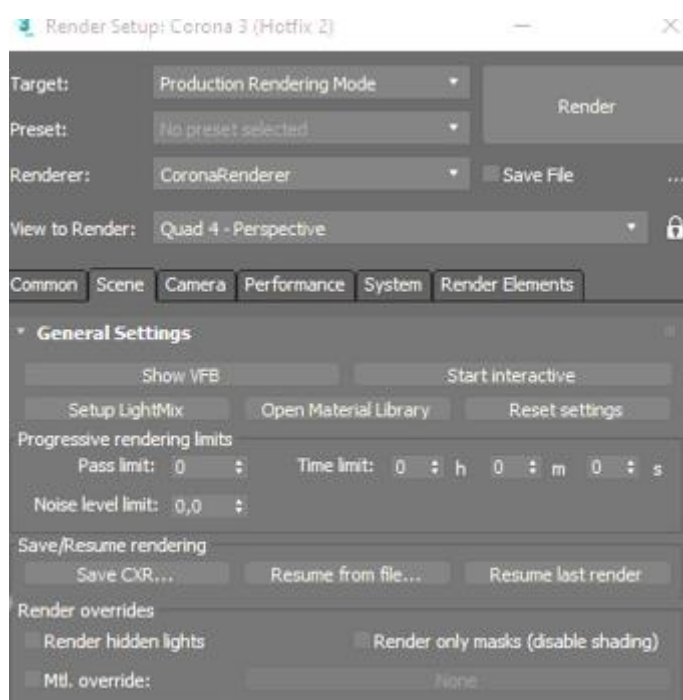


Рисунок 2.19 – Render Setup

Завдяки збереженню рендера у форматі CXR можна продовжити після перезавантаження комп'ютера, або на іншому комп'ютері.

Resume from file продовжує рендер завантаженої картинки. Потрібно відкрити відповідну сцену проекту та завантажити файл.

Resume last render – якщо візуалізація була перервана або встановлена на паузу, у буфері VFB зберігається прорахунок сцени, це дозволяє продовжити процес візуалізації.

Denoising – це функція, яка прибирає «зашумлення» з рендеру за спеціальними алгоритмами. Дана функція дозволяє значно скоротити час на візуалізацію, однак недоліком цієї функції є те, що вона замилує ті частки пікселів, що може сказатися на якості.

Corona Renderer дозволяє налаштовувати глобальне освітлення за допомогою неба та сонця і дозволяє додавати HDRI карти вже з запеченим освітленням та тінями. Спосіб встановлення глобального освітлення представлено на рисунку 2.20.

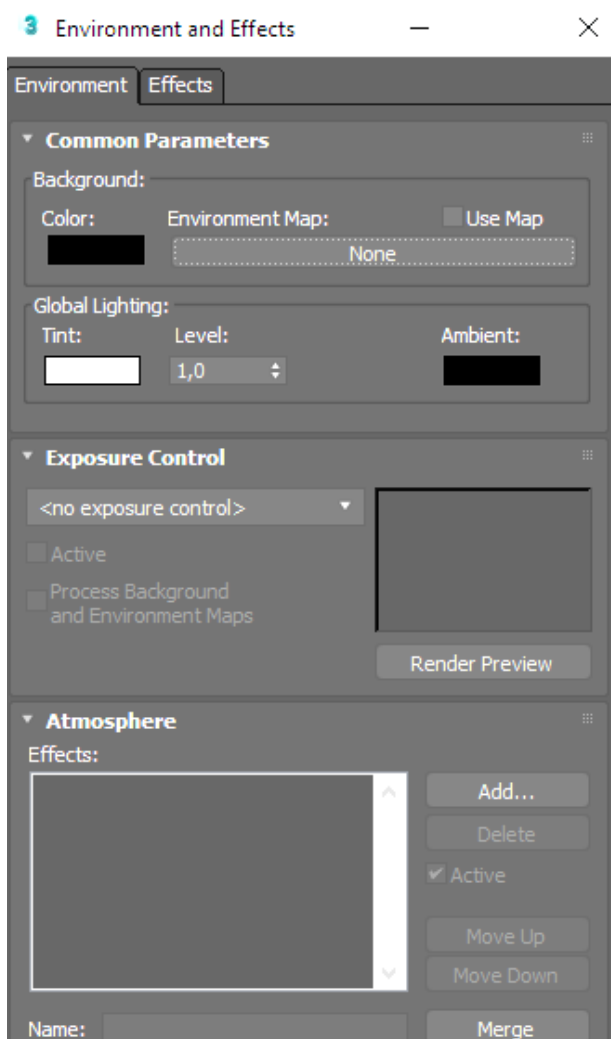


Рисунок 2.20 – Environment map

До переваг Corona Renderer слід віднести групове управління освітленням light mix, дана функція об'єднує в групи джерела освітлення для кращого контролю над ними і також зменшує навантаження на ПК. Для цього потрібно додати його в сцену та вказати групи джерел освітлення (рис 2.21).

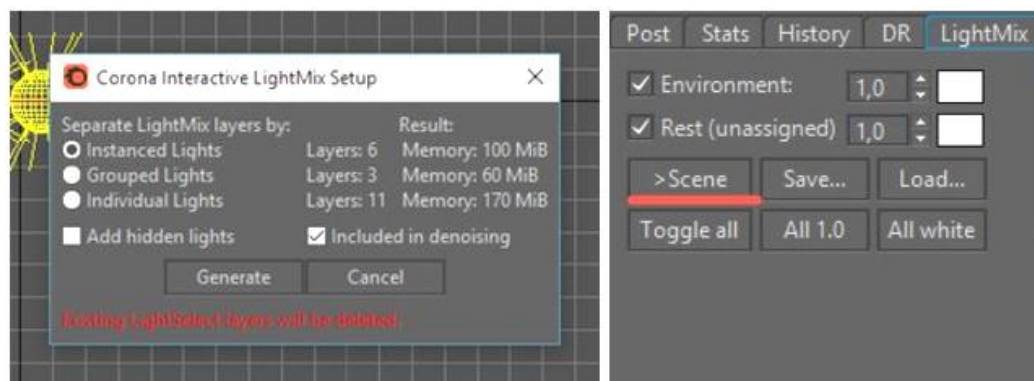


Рисунок 2.21 – LightMix

Instanced Lights – групування світильників в LightMix які створені як дочірні копії.

Grouped Light – групування світильників, які об'єднані в групу.

Другою перевагою є Corona Material Library вбудована бібліотека матеріалів високої якості (рис 2.22).

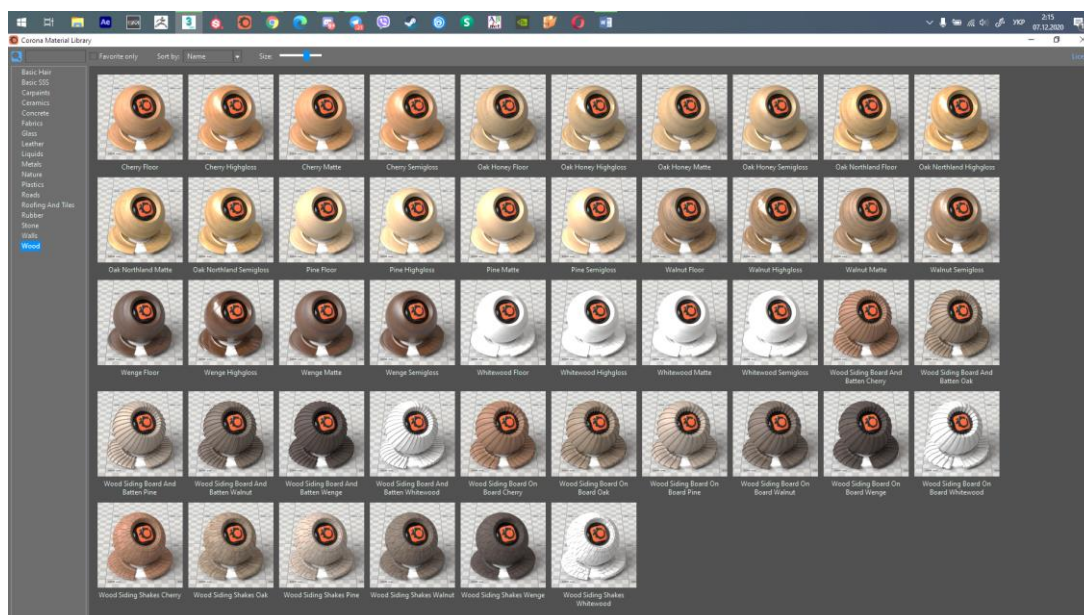


Рисунок 2.22 – Corona Material Library

V-Ray система рендерингу розроблена компанією, Chaos Group (Болгарія). Перша бета-версія рендерера V-Ray з'явилася в 2000 році. Сумісний із старими версіями 3dsMax. V-Ray використовується у багатьох сферах завдяки своїй гнучкості та широкому набору інструментів, таких як архітектура та кіно індустрія. Також в нього вбудована V-Ray Sky і V-Ray Sun. V-Ray працює як плагін для Autodesk 3ds Max, Cinema 4D, SketchUp, Rhino, Autodesk Maya. В даному модулі присутні кілька алгоритмів прорахунку глобального освітлення Global Illumination (рис 2.23):

- Light Cache;
- Photon Map (фотонна карта);
- Irradiance Map;
- Brute Force (QMC).

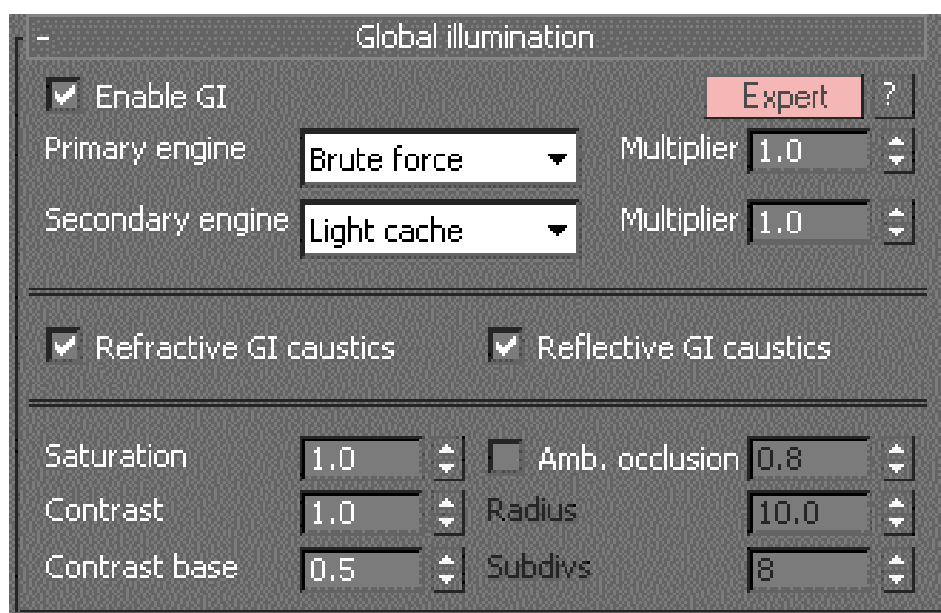


Рисунок 2.23 – Global Illumination

Є можливість вибору різних алгоритмів для прорахунку відображення і віддзеркалення глобального освітлення.

Основні переваги V-Ray:

- підтримує розподілений рендеринг на декількох комп'ютерах (Distributive rendering);
- має безліч гнучких налаштувань якості зображення для отримання хорошого результату за прийнятний час;
- має в своєму функціоналі великий і постійно розширюваний набір матеріалів, здатних імітувати ті або інші властивості об'єктів, таких як SSS (підповерхневе розсіювання);
- додає в 3ds Max процедурні текстури і утиліти, що полегшують роботу зі сценою (наприклад, v-ray multisub texture і v-ray edges texture);
- підтримує виведення власних Render Elements для подальшого складання в програмах композітингу.

Починаючи з версії V-Ray 2.0, V-Ray RT входить в штатну поставку плагіна до 3ds Max. При використанні рендерера в реальному часі, V-Ray RT частково бере настройки зі звичайного V-Ray (настройки освітлення, колірного загасання, навколишнього середовища), а для обчислень використовує власний алгоритм. V-Ray RT здатний робити обчислення за допомогою процесора або ж відеокарти .

WebGL (Web Graphics Library) програмна бібліотека для мови JavaScript , призначена для візуалізації інтерактивної тривимірної і двомірної графіки. WebGL додає в веб тривимірну графіку, вводячи API, який побудований на основі OpenGL, що дозволяє його використовувати в елементах HTML5. Основні можливості бібліотеки:

- додавання двомірного контенту в контекст WebGL;
- використання шейдерів для завдання кольору та текстури в WebGL;
- анімація об'єктів за допомогою WebGL;
- створення 3D за допомогою WebGL;
- освітлення WebGL;

Найважливішим новим інструментом для розширених додатків HTML5 є «полотно» (Canvas) – поверхня для моделювання. Полотно є відокремленою від всіх інших елементів HTML, тому що для роботи з ним потрібна мова

JavaScript. Іншого способу для креслення фігур на сайті немає. Тобто, полотно, по суті, є засобом програмування [17]. WebGL додає в веб тривимірну графіку, вводячи API, який побудований на основі OpenGL ES 2.0, що дозволяє його використовувати в елементах HTML5. Можливості Canvas у HTML5:

- додавання двомірного контенту в контекст WebGL;
- використання шейдерів для завдання кольору в WebGL;
- створення 3D за допомогою WebGL;
- Використання текстур WebGL;
- освітлення WebGL;
- Анімація текстур WebGL.

Можливості обробки відео в HTML5 поки що обмежені рішенням проблем програмування відеоплеєра і його елементів управління за допомогою Javascript та CSS. спеціальні API для відео поки не розроблені.

Одним із популярних способів використання Web-технологій в освіті є квестові Web-тести [11]. Приклад використання WebGL представлено на рисунку 2.24.

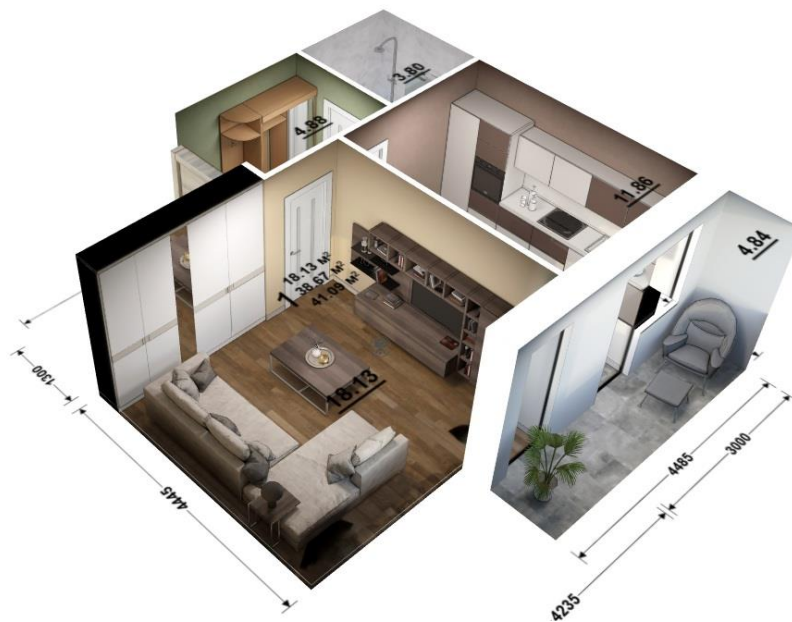


Рисунок 2.24 – Додавання тривимірної графіки у WebGL

2.4 Висновки до розділу

В даному розділі було розглянуто формування віртуальної реальності, методи формування 3D моделей, як математично створюються примітивні форми, методи створення ліній та їх опис. Були розглянуті та проаналізовані методи будування освітлення та відбиття променів. Також були розглянуті сучасні 3D редактори та їх властивості, системи візуалізацій та створення матеріалів. Слід зазначити, що для окремих задач можуть використовуватися різні 3D редактори. В даній роботі в якості 3D редактора використовується 3D MAX, тому, що він має більший функціонал та використовується для архітектурної візуалізації, також має 2 найпотужніших системи візуалізації та має багато плагінів для пришвидшення моделювання окремих поверхонь на відміну від Blender, який застосовується для моделювання ігрових персонажів.

Слід зазначити, що при зрівнянні швидкодії Blender опереджає 3D MAX. Однак 3D MAX опереджає у швидкості візуалізації при використанні системи рендерінгу V-ray. Тому було прийнято рішення використовувати V-ray, як основну систему візуалізації. В завершенні була розглянута бібліотека WebGL для імпортування 3D графіки до веб сайтів.

3 РЕАЛІЗАЦІЯ 3D-МОДЕЛІ ВИРОБНИЧИХ ПРИМІЩЕНЬ

3.1 Формування вхідної інформації про виробничі приміщення

Для визначення послідовності формування, розробки та візуалізації(рендерінгу) 3D-моделей виробничих приміщень пропонується інформаційна модель, зображена на рисунку 3.1.



Рисунок 3.1 – Інформаційна модель формування виробничих приміщень

Відповідно до запропонованої моделі, процес 3D-моделювання приміщень виробничого типу (як і інших виробничих об'єктів) складається з наступних етапів: створення креслень об'єктів, моделювання об'єктів з примітивів та використанням шаблонів, накладання текстур та налаштування властивостей матеріалів, експортування до засобів віртуальної реальності (т.з ігрового двигуна). Якщо після експортування на етапі тестування розроблена модель не задовольняє умовам проектування відбувається повернення до етапів моделювання та текстурування. Якщо ж розроблена модель задовольняє вимогам, відбувається модифікація 3D-моделі за рахунок додавання ефектів освітлення, редагування текстур. Результатом вказаних етапів є математичний обрахунок відображення моделей, текстур та освітлення і забезпечення, таким чином, візуалізації (рендерінгу) моделей. Якщо відображена модель не відповідає вимогам, має здійснюватися повернення до етапу проміжного тестування. Якщо модель задовольняє вимогам (в тому числі потребам замовника), відбувається її експортування у Web- або VR-формат.

Розглянемо детально окремі етапи інформаційної моделі.

Перш за все потрібно проаналізувати план приміщення. Встановити габарити вікон, колон та стін. Для більш точної реалізації план був перенесений до програми Archicad, з метою отримання креслень у форматі dwg. Перед створенням креслення потрібно створити новий робочий лист та накласти зображення плану, для цього їх слід перетягнути до листу. Надалі користуючись інформацією з плану, обрав інструмент лінія та повторив контури стін з точністю до 10мм, результат наведено на рисунку 3.2.

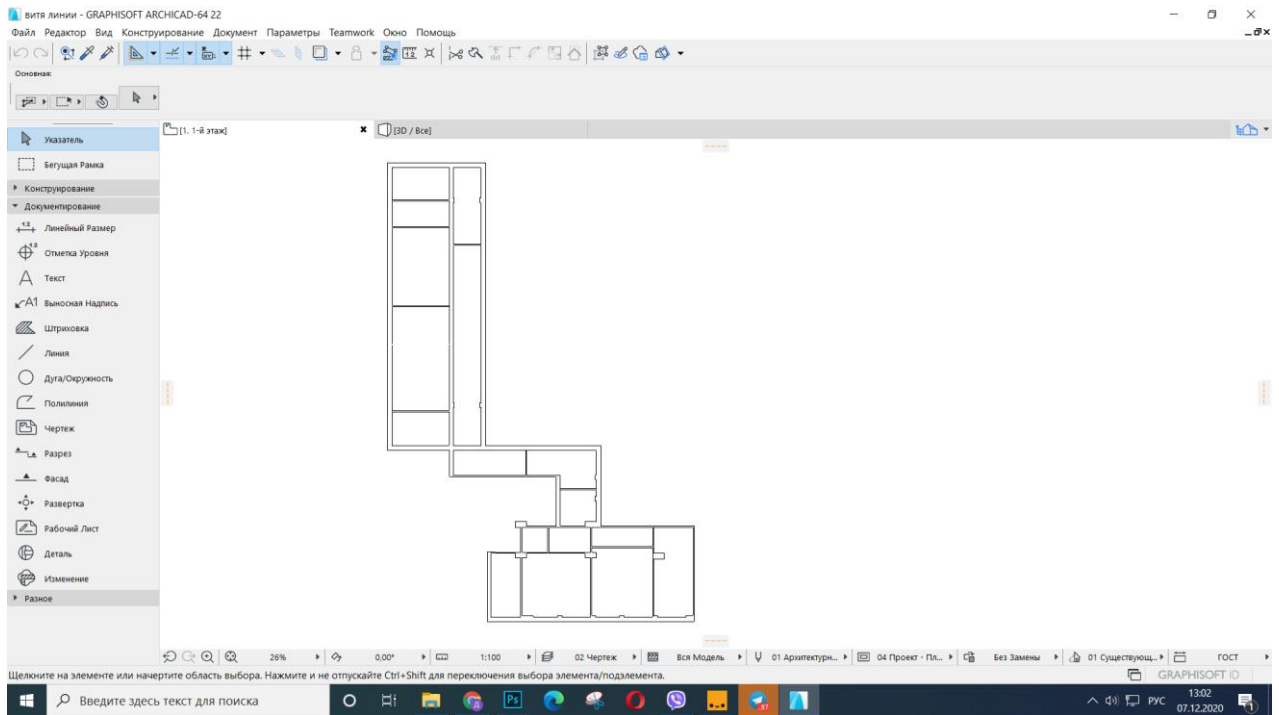


Рисунок 3.2 – Креслення у програмі Archicad

Після цього потрібно зберегти файл у форматі dwg, переходимо до кнопки файл, обираємо зберегти як, та вказуємо формат.

3.2 Моделювання приміщень та об'єктів в приміщеннях

Імпортуємо план приміщення у 3D Max. Імпортування креслення до 3D Max представлено на рисунку 3.3.

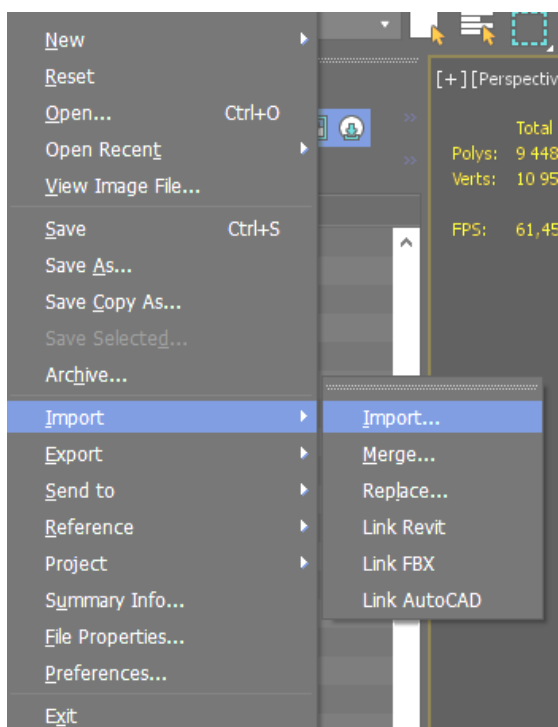


Рисунок 3.3 – Імпорт до 3D Мах

Після імпорту до 3D Мах отримуємо заготовку в якості сплайну (редагуємої лінії). Отримане креслення представлено у міліметрах та повністю відповідає реальному приміщенню. На рисунку 3.4 представлено імпортований план приміщення.

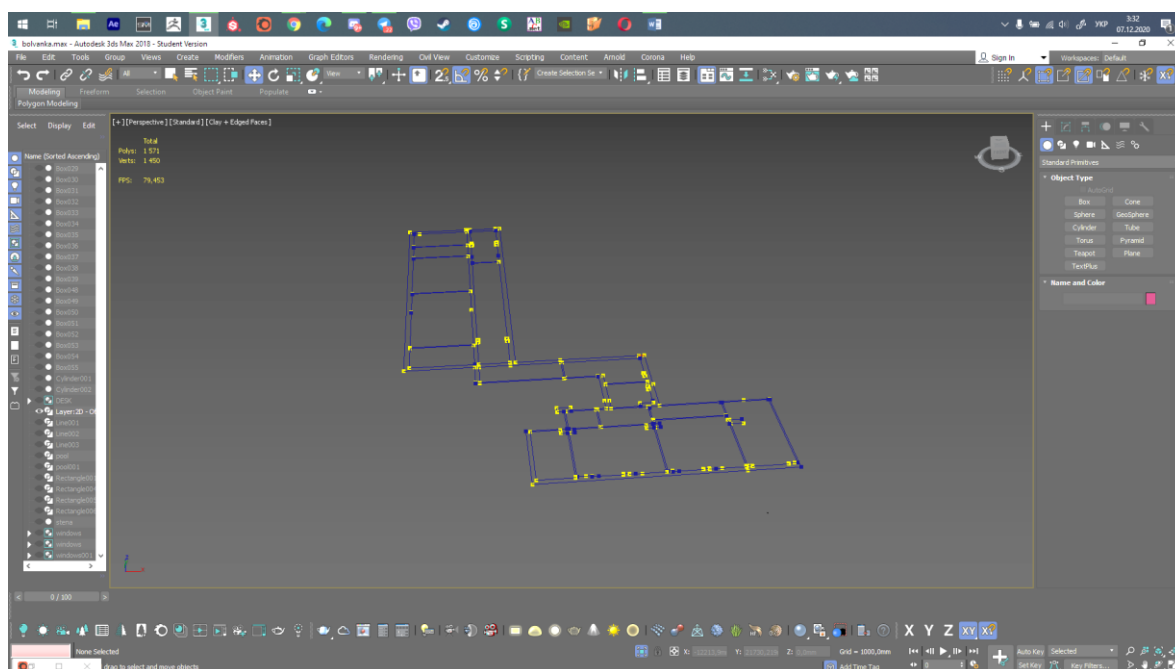


Рисунок 3.4 – План приміщення

Щоб видавити із даного сплайну стіни необхідно зайти у режим редагування точок та зклеїти вершини в яких були розриви (рис 3.5).

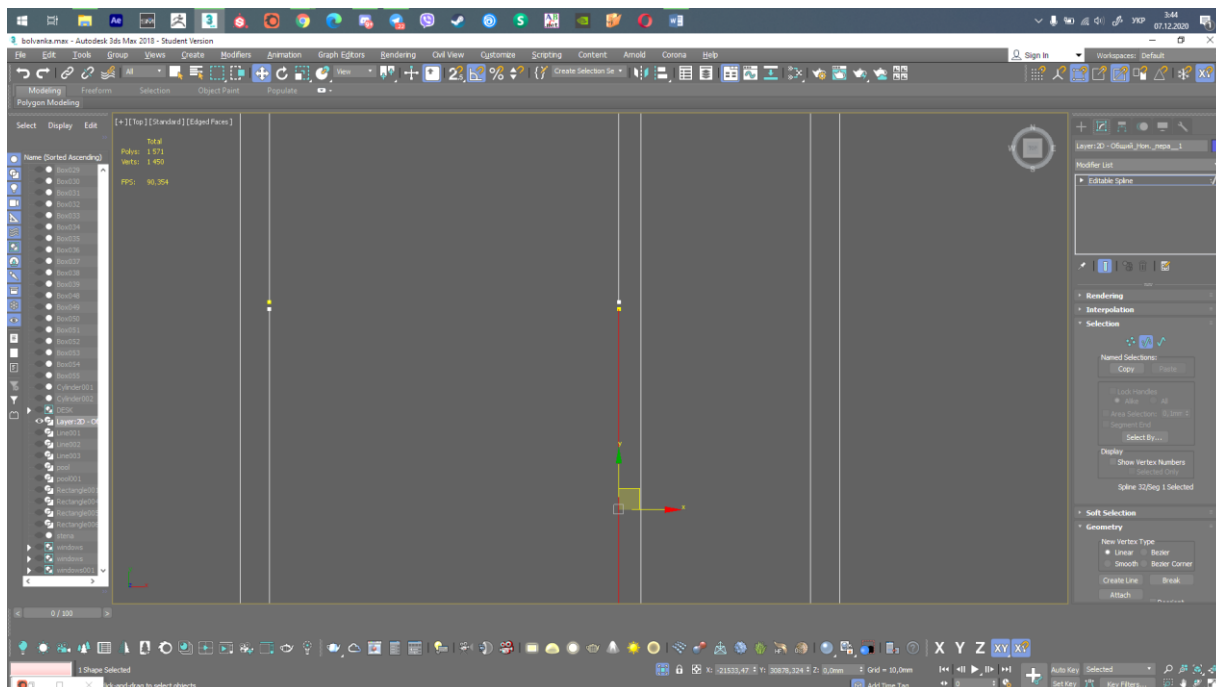


Рисунок 3.5 – З'єднання розривів

Наступним кроком до отриманого сплайну застосовуємо модифікатор Extrude та в параметрі Amount виставляємо 5000, що еквівалентно з висотою стін у 5 метрів (рис 3.6).

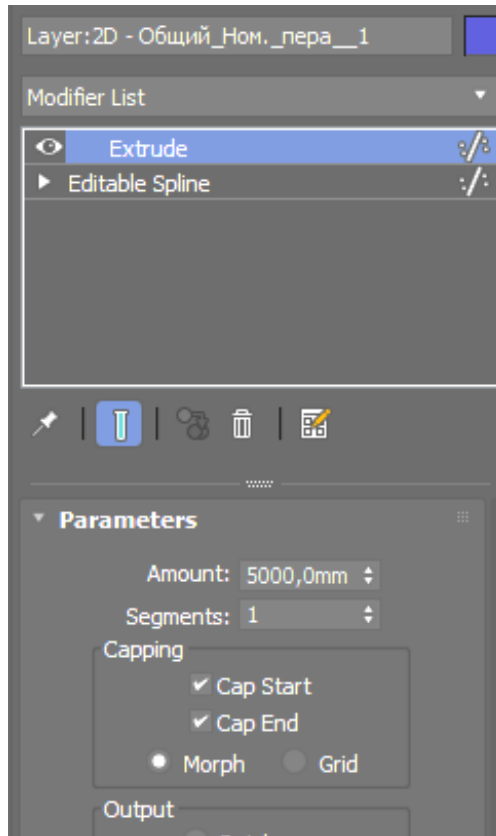


Рисунок 3.6 – Модифікатор Extrude

Наступним кроком у відтворенні приміщення є створення місця під вікна, для цього необхідно встановити розміри вікна та на базі результатів зробити Box із вхідних розмірів та розташувати їх згідно плану будівлі. Створюємо коробку з габаритами 2000x2000x3000 мм та за допомогою команди Move пересуваємо на місце, результат показано на рисунку 3.7.

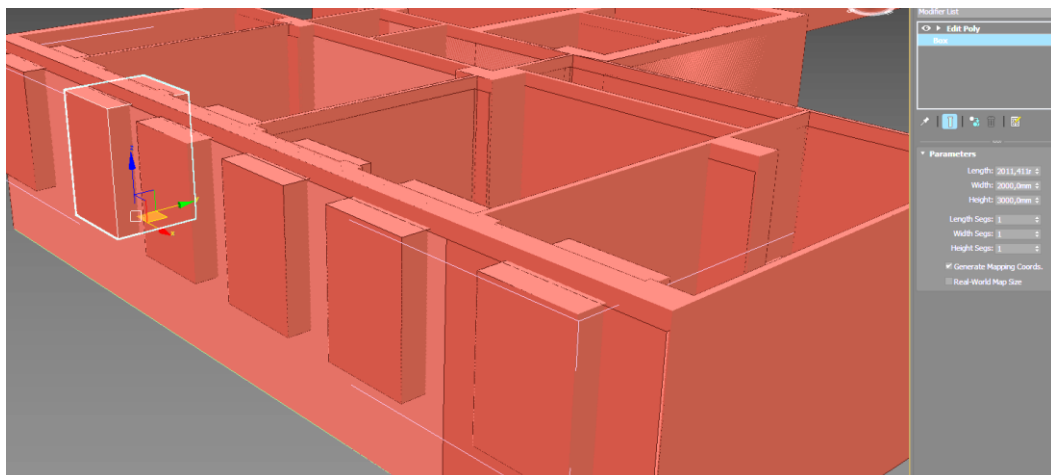


Рисунок 3.7 – Підготовка форм для вікон

Щоб створити отвір необхідно вибрати стіни та пейте у режим Compound Objects вибрати Boolean та встановити галку на Subtract та додати створені коробки (рис 3.8).

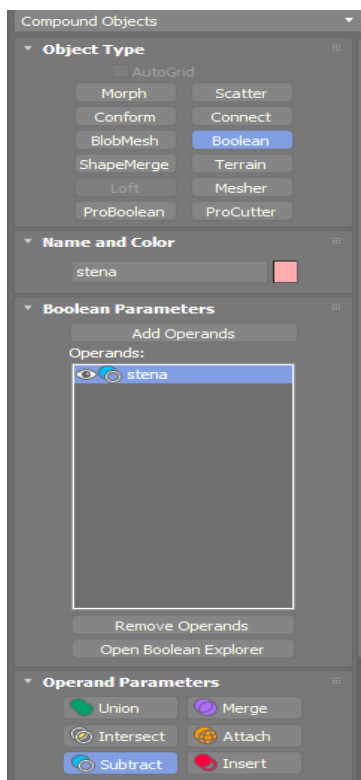


Рисунок 3.8 – Меню Boolean

Результат математичного віднімання представлено на рисунку 3.9

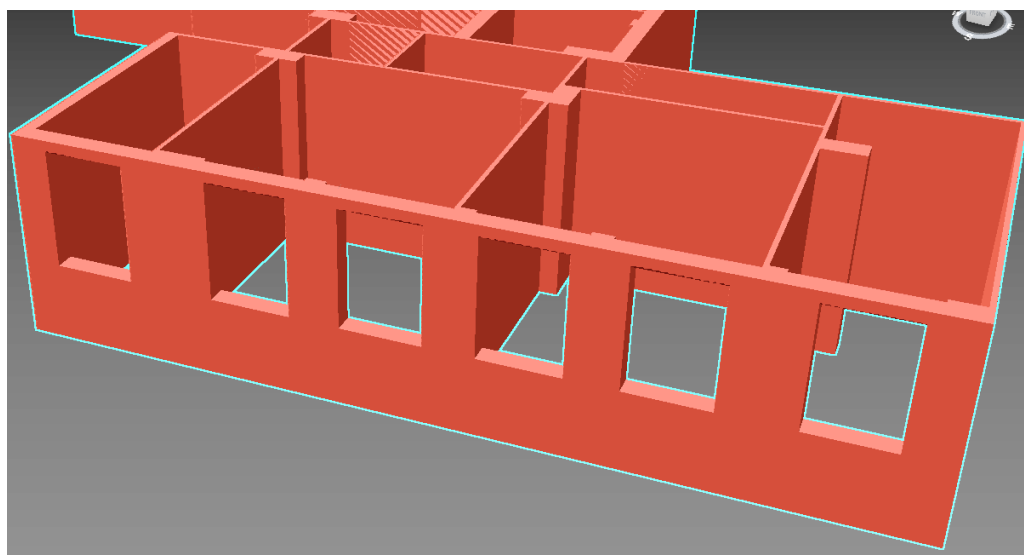


Рисунок 3.9 – Результат математичного віднімання

Наступним кроком є створення елементів декору. Спочатку слід створити парти, столи та стільці. Для створення парти необхідно створити Box з певними габаритами, що буде виступати в ролі стільниці, надалі потрібно створити металевий каркас, використовуємо Box та конвертуємо його в Edit Poly. Переходимо у режим редагування ребр виділяємо ребра та натискаємо Connect для точного позиціонування та можливості створювати нові елементи каркасу (рис 3.10).

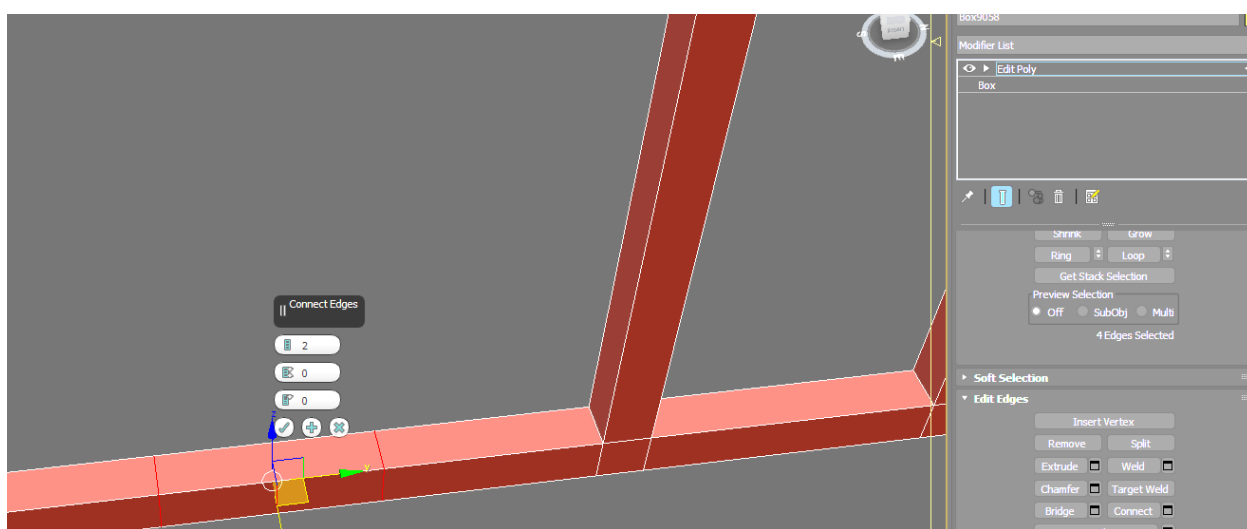


Рисунок 3.10 – Створення додаткових ребер

Після цього використовуємо функцію Mirror (віддзеркалювання) та виділяємо нові отримані полігони та натискаємо на інструмент Bridge для того, щоб їх з'єднати (рис 3.11).

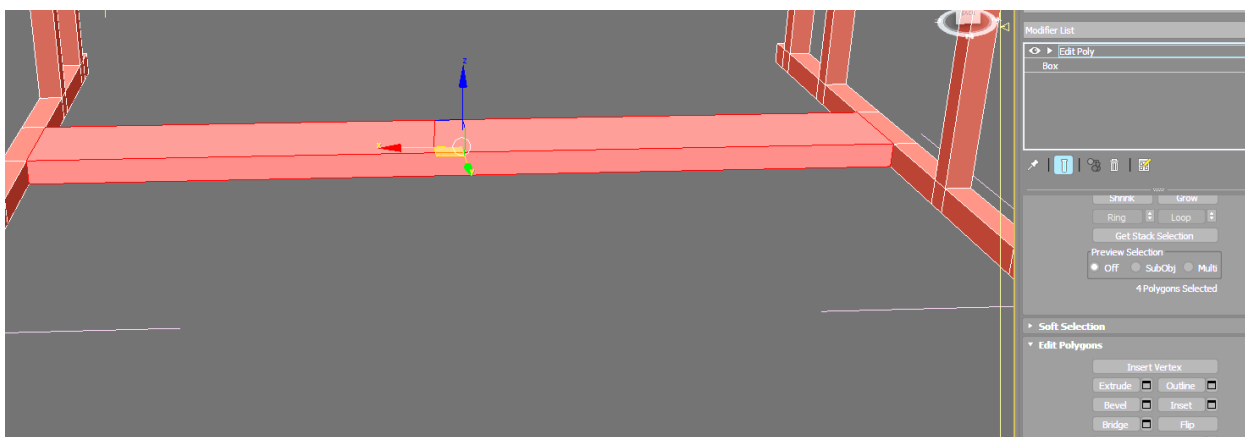


Рисунок 3.11 – Створення каркасу за допомогою інструмента Bridge

Надалі створюємо елементи спинки та об'єднуємо увесь об'єкт. Результат представлено на рисунку 3.12.

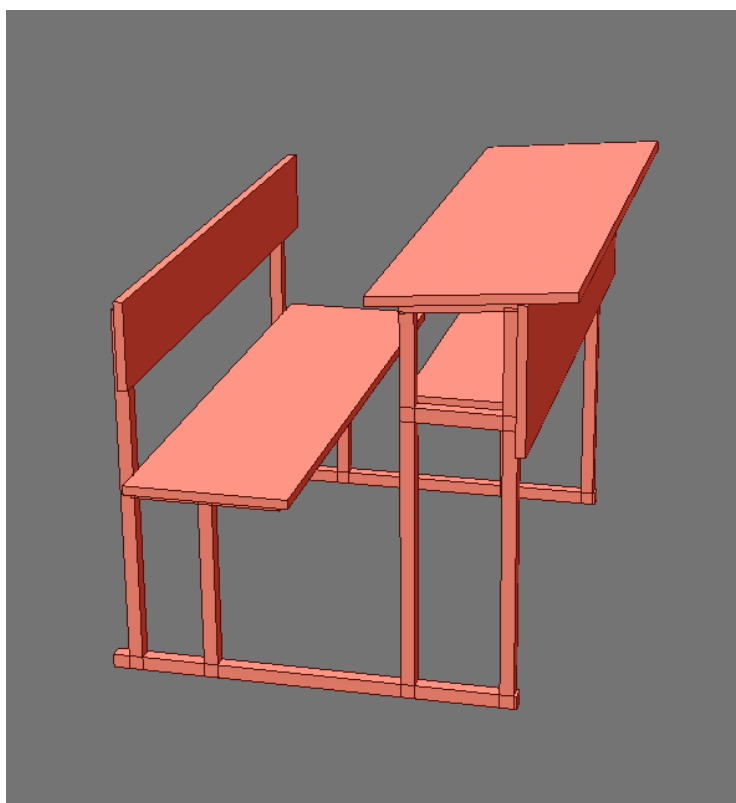


Рисунок 3.12 – Результат моделювання

Розміщуємо створені моделі згідно плану приміщення. Результат представлено на рисунку 3.13.

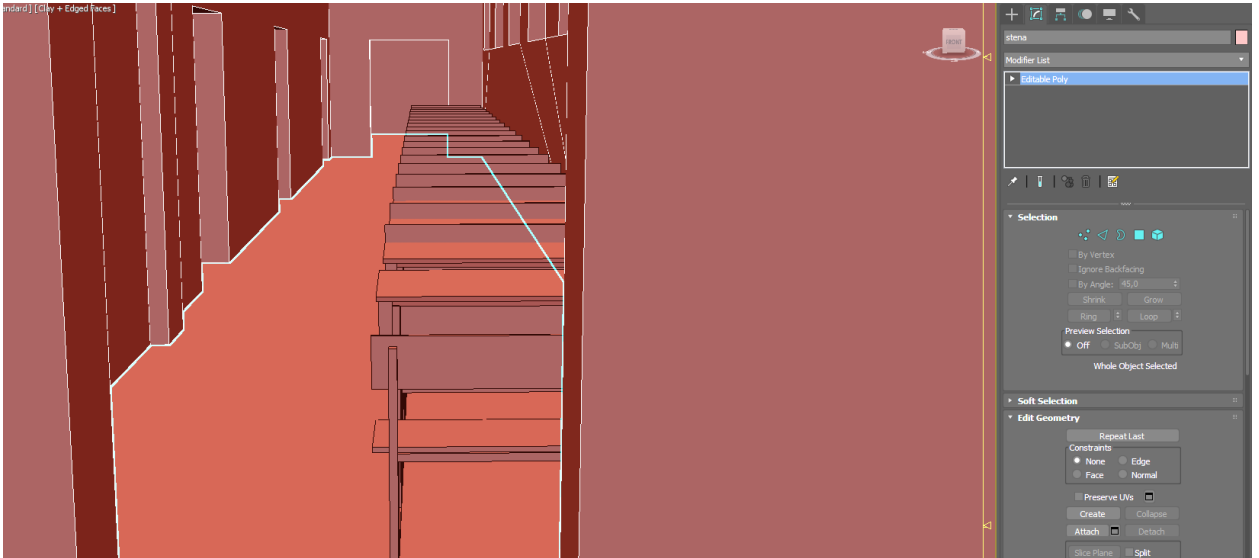


Рисунок 3.13 – Розміщення моделей

За такою же схемою створюємо інші столи та стільці (рис 3.14).

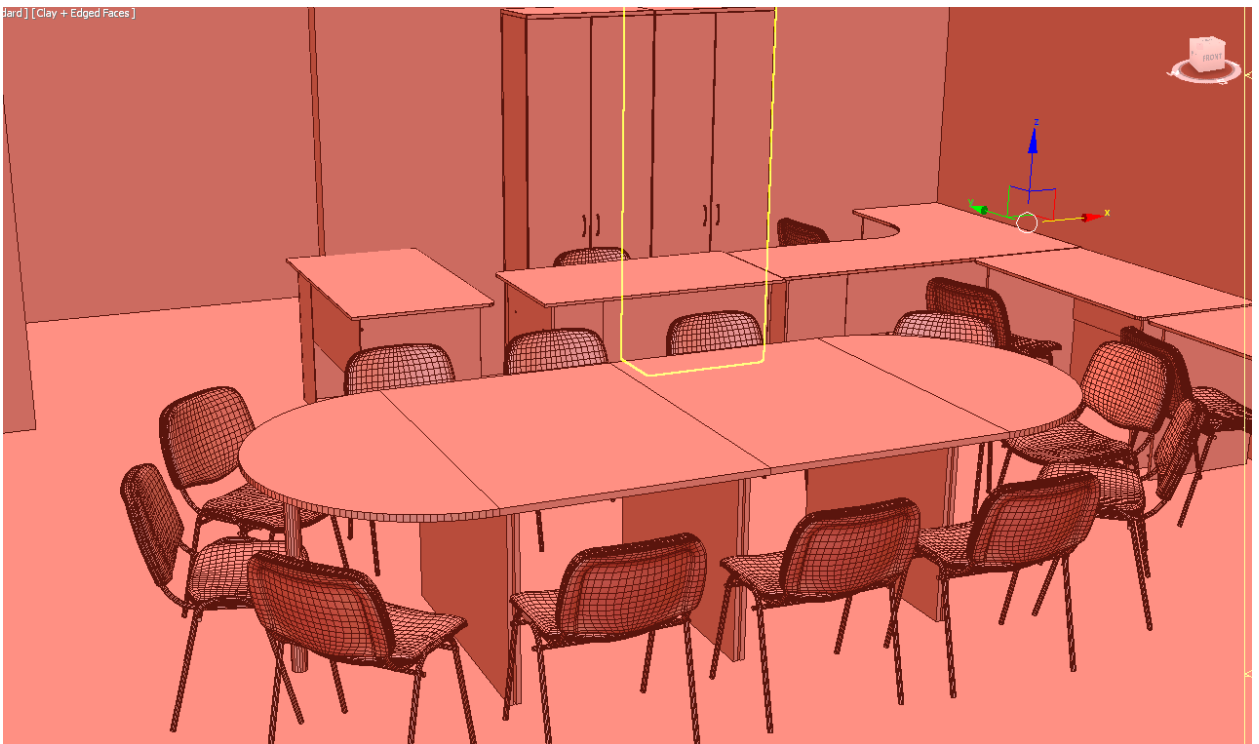


Рисунок 3.14 – Результат моделювання меблів

Окрім меблів необхідно відтворити технічне обладнання аудиторії, а саме роботів маніпуляторів, комп'ютери, 3D принтер, пересувних роботів, та плату Arduino. Для моделювання принтера потрібно створити його базові

форми. Першим етапом є створення каркасу, для цього створюємо декілька коробок, одна з них буде каркасом а іншими за допомогою Boolean будемо надавати форму (рис 3.15).

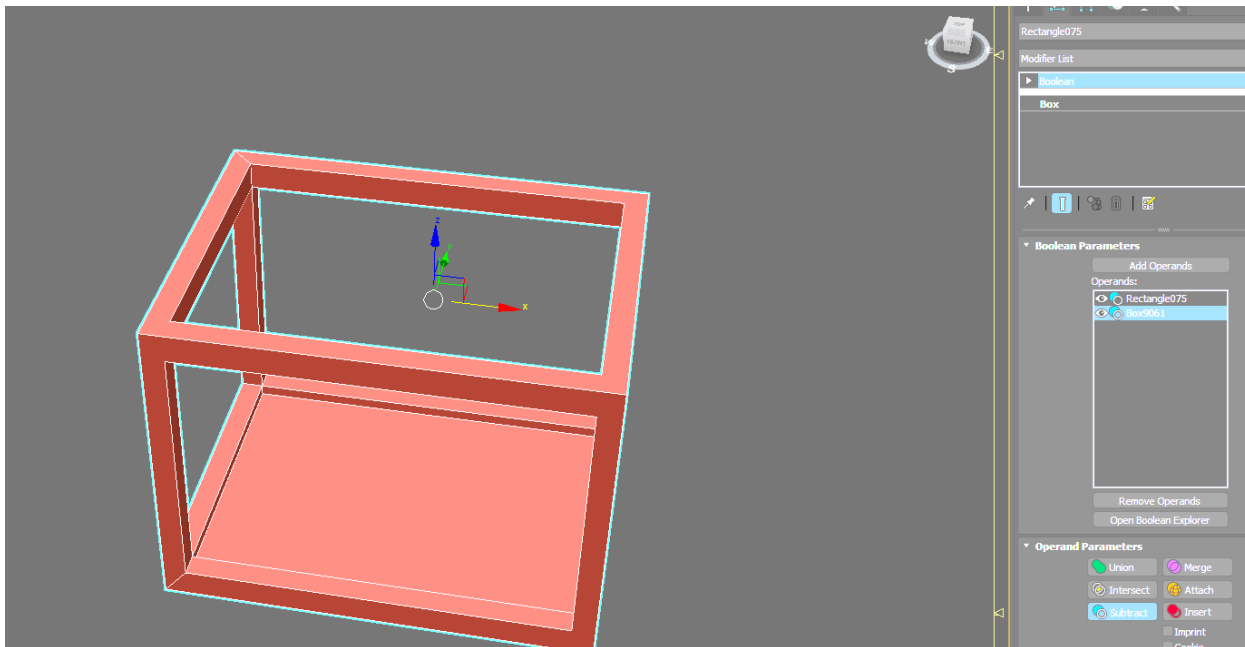


Рисунок 3.15 – Каркас 3D принтера

Надалі потрібно створити інші елементи каркасу. У меню створення об'єктів обираємо Splines та обираємо інструмент Line та відвоюємо задню панель принтера. Для згладжування контурів у функції Line обираємо інструмент Fillet, результат представлено на рисунку 3.16.

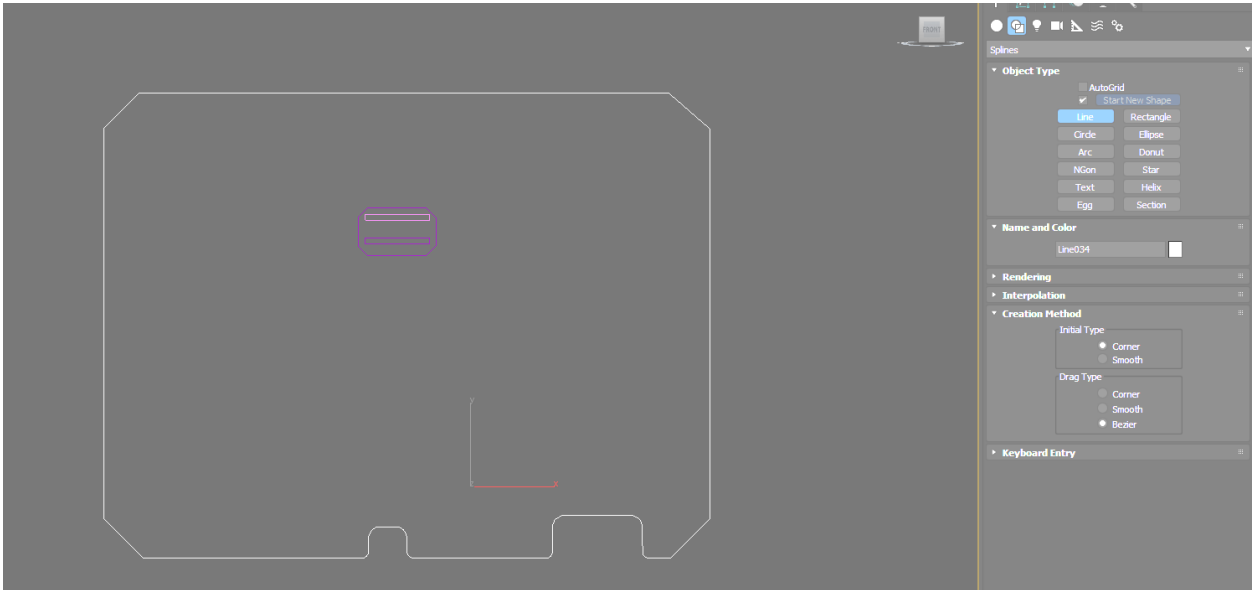


Рисунок 3.16 – Створення панелей

На отриманий Line додаємо модифікатор Extrude для того, щоб надати об'єму. Повторюємо дані операції для інших частин. Наступним кроком за допомогою примітивів і ліній створюємо рейки та гвинти, результат представлено на рисунку 3.17.

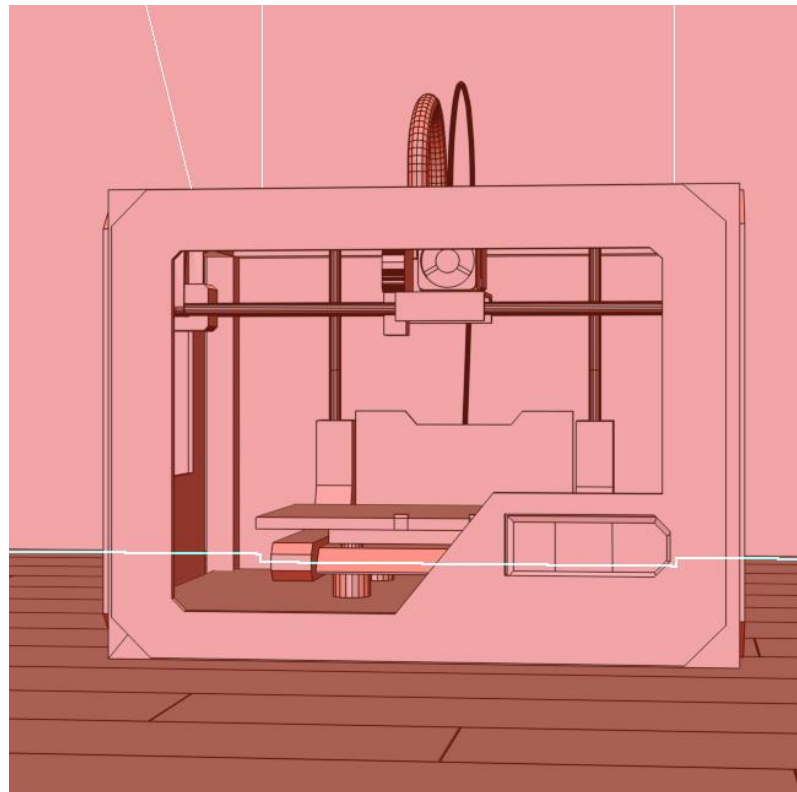


Рисунок 3.17 – 3D принтер

Моделювання маніпулятора проводилось за допомогою примітивів для отримання базової форми, наступним етапом є додавання до примітивів модифікатора Edit Poly, для створення фасок, переходимо до режиму редагування ребер та у меню модифікатора обираємо Chamfer та встановлюємо кількість сегментів та ширину фаски, для створення вставок необхідно перейти у режим редагування полігонів обираємо Inset та вказуємо товщину бортів, результат представлено на рисунку 3.18.

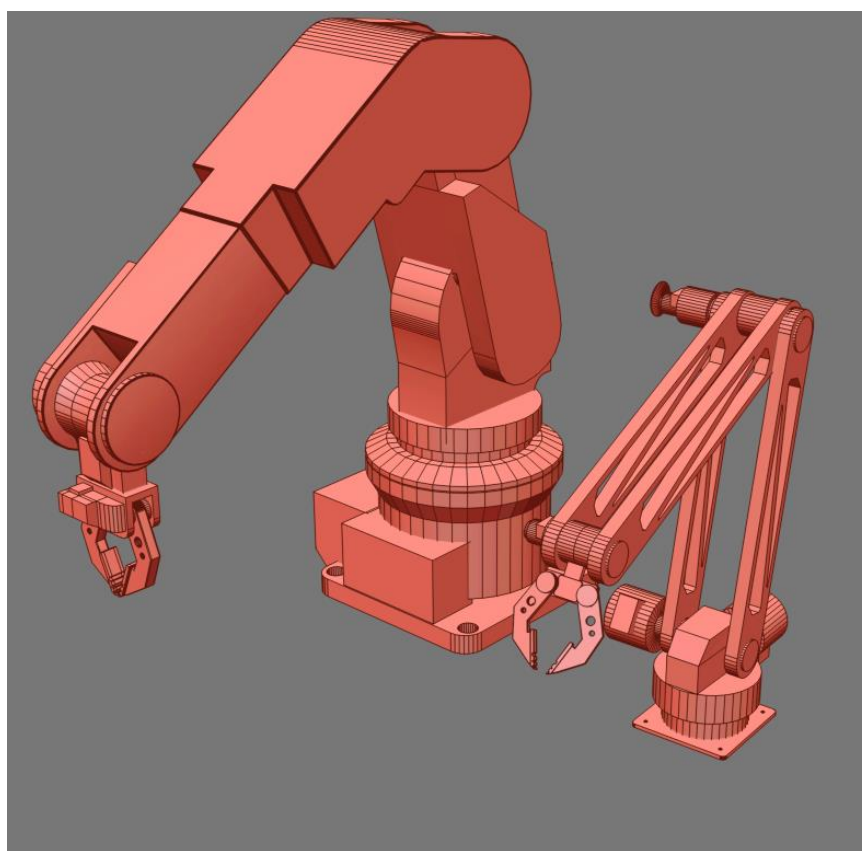


Рисунок 3.18 – Маніпулятор

Наступним кроком є розташування створених об'єктів в сцені (рис 3.19).

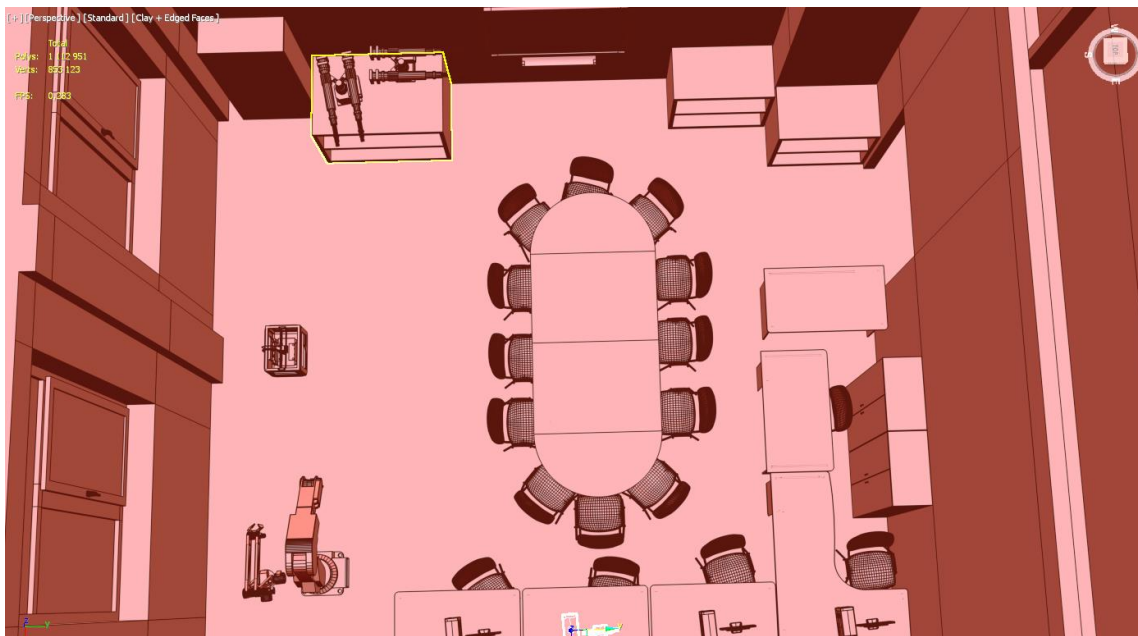


Рисунок 3.19 – Розміщення моделей у сцені

Наступним кроком необхідно створити матеріали для моделей. Загалом для стандартної візуалізації використовується Corona Render або V-ray, однак експорт матеріалів та освітлення у версії Unreal Engine 4.23 не підтримує жодну з даних систем рендерінгу. Тому будуть створюватися стандартні матеріали, які вбудовані у 3D Max. Для цього відкриваємо редактор матеріалів та перетаскуємо із списку Standart Material (рис 3.20).

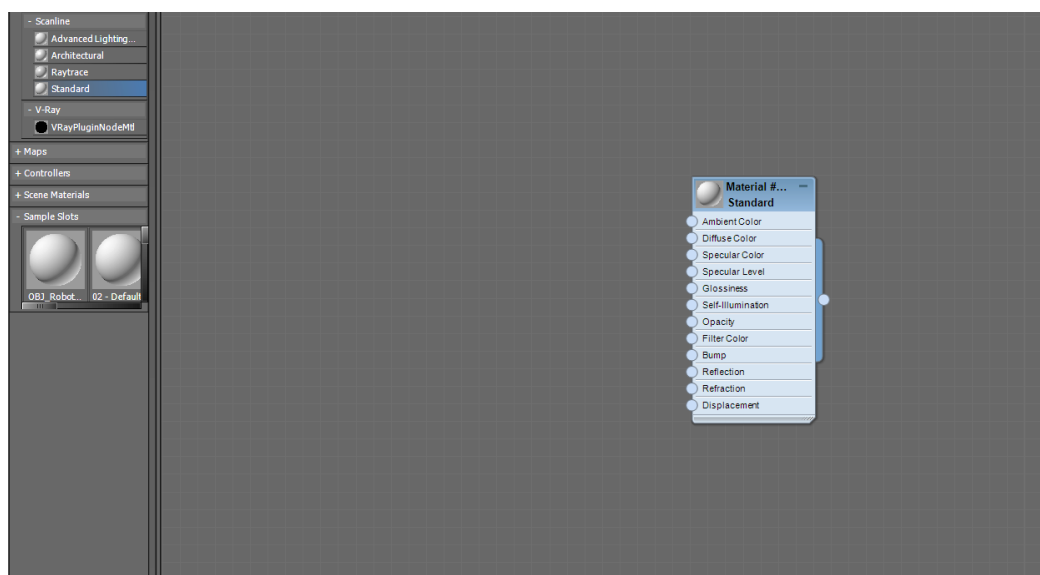


Рисунок 3.20 – Вікно редагування матеріалів

Далі потрібно додати текстуру для відображення матеріалу на об'єкті. Для цього знаходимо потрібну нам текстуру та перетягуємо її до редактора. Створимо матеріал плитки. Наскладуємо текстуру до позиції Diffuse Color, щоб матеріал відображав нашу текстуру. Наступним кроком потрібно встановити параметри відбиття світла і матовості та глянцю (Specular Level, Glossiness). Для того, щоб придати об'єму даній текстурі потрібно додати текстуру у параметр Bump та встановити його силу видавлювання (рис 3.21).

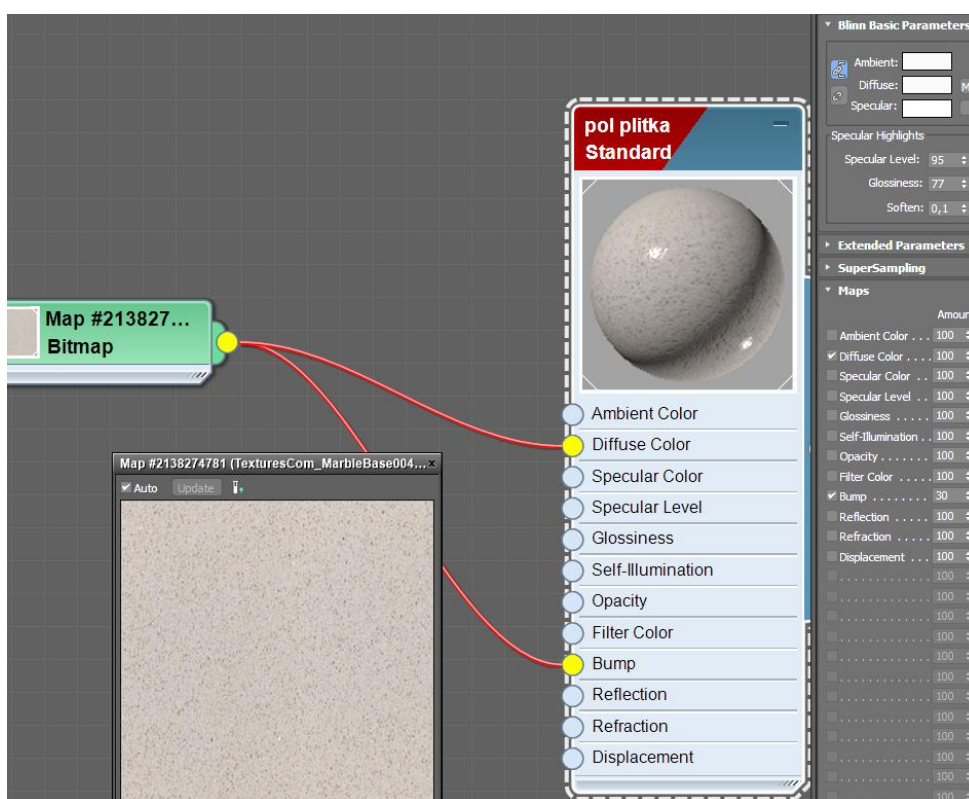


Рисунок 3.21 – Налаштування матеріалу плитки

Таким же чином створюємо інші матеріали в сцені, результат представлено на рисунку 3.22.

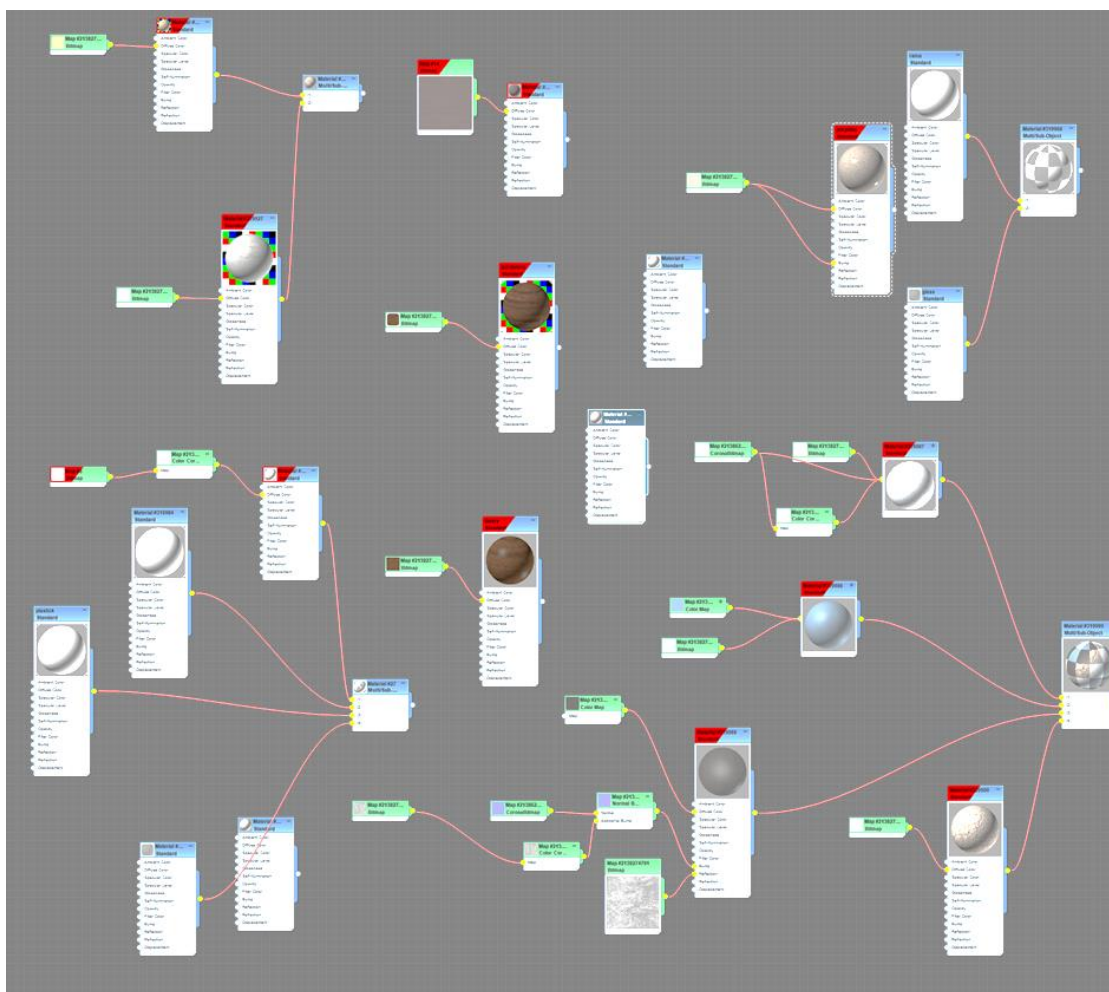


Рисунок 3.22 – Матеріали

Для того, щоб текстура відображалася правильно слід задати текстурні координати за допомогою модифікатора UVW Map або Unwrap UVW. Найбільш швидким та наглядним рішенням для накладення текстури є модифікатор UVW Map, який кубічно проектує текстуру на об'єкт. Однак, Unwrap UVW розгортує кожен полігон моделі, що надає більш реалістичний вигляд. До створеного об'єкту додаємо модифікатор Unwrap UVW та відкриваємо Editor де і буде відбуватися розгортка (рис 3.23).

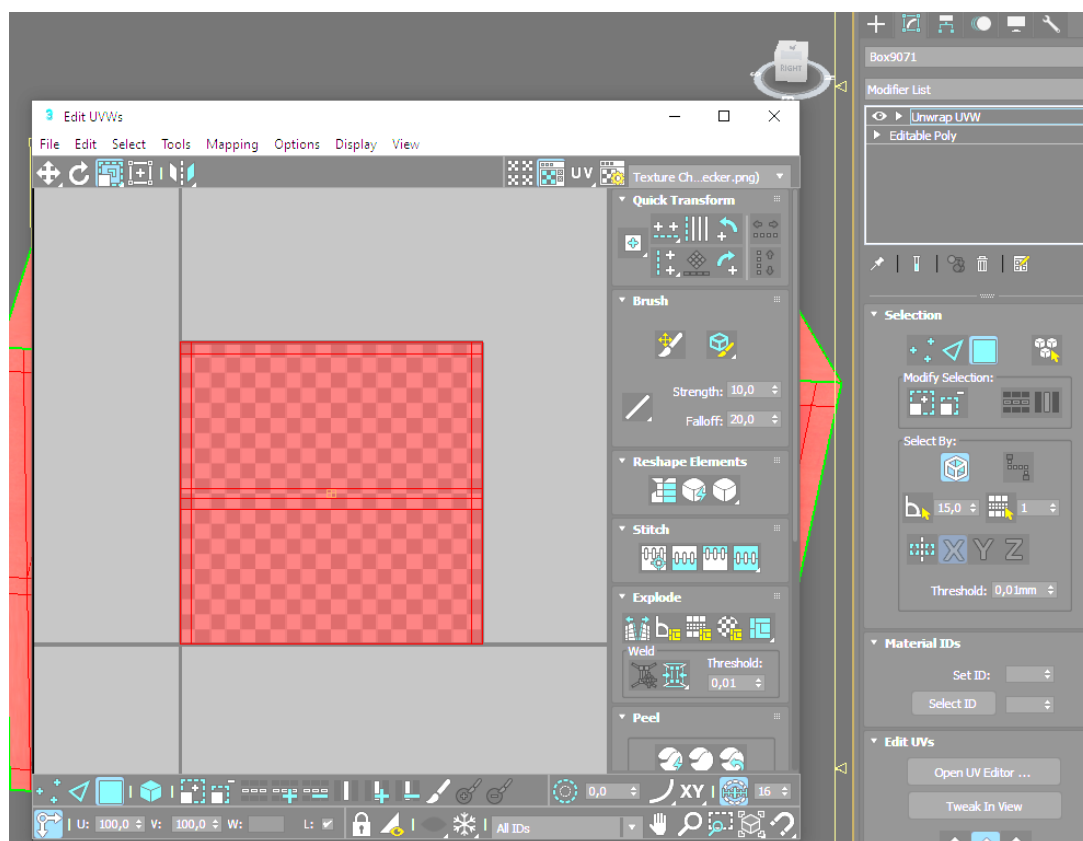


Рисунок 3.23 – UV Editor

Переходимо до вкладки Mapping та обираємо Flatten Mapping (рис 3.24).

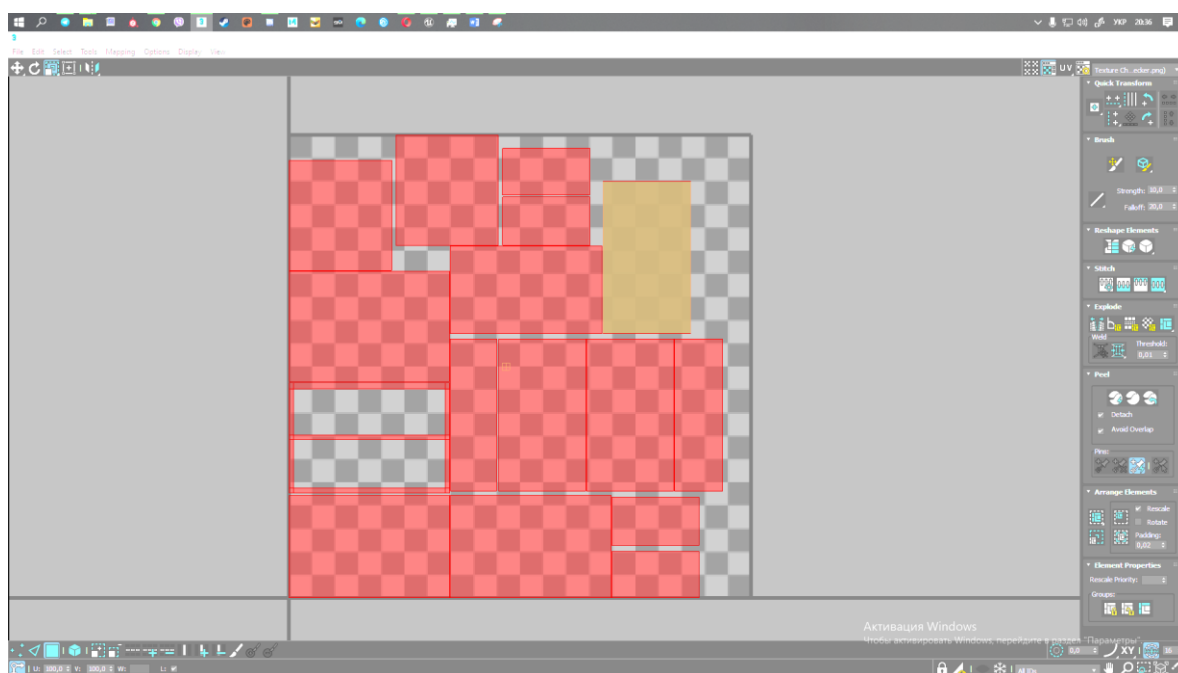


Рисунок 3.24 – Розгортка полігонів

Для того, щоб експортувати моделі до Unreal Engine потрібно створити ще одну розгортку однак по другому каналу, цей канал потрібен для запікання освітлення. Тому обираємо Map Channel 2 та розгортаємо модель, при розгортці з'являється діалогове вікно, натискаємо Abandon (рис 3.25).

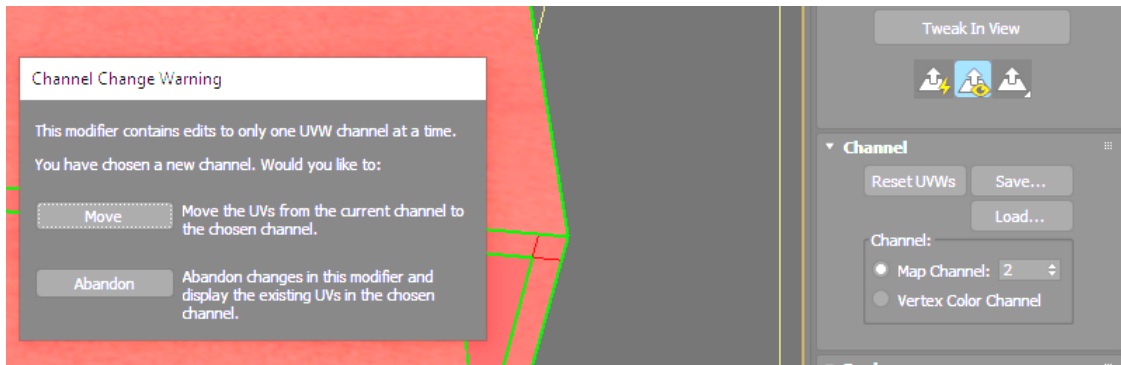


Рисунок 3.25 – Map Channel

Після цього кожен об'єкт потрібно конвертувати до Editable Poly. Це необхідно, тому, що Unreal Engine може сприймати лише геометричні об'єкти та в них не повинно бути відкритих модифікаторів. Тому виділяємо усі об'єкти натискаємо праву кнопку миші та обираємо Convert to та у списку обираємо Editable Poly (рис 3.26).

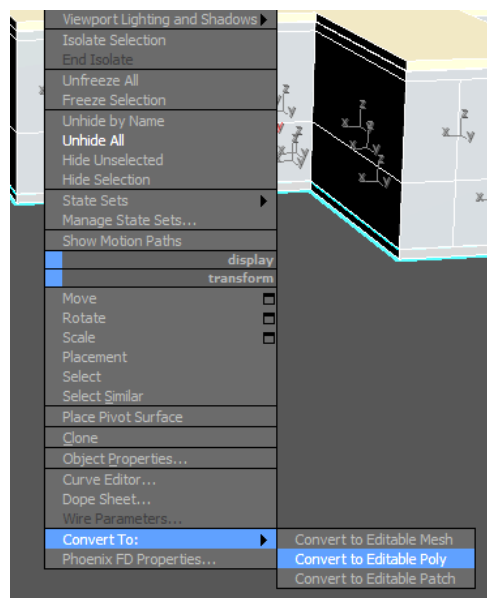


Рисунок 3.26 – Конвертування до Editable Poly

Надалі переходимо до меню File Export та обираємо Export (рис 2.27)

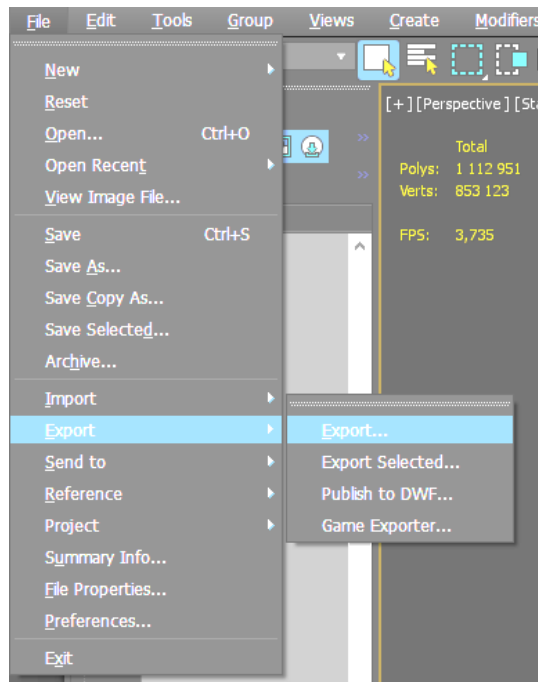


Рисунок 3.27 – Export у UE

3.3 Конвертування отриманих моделей у засоби віртуальної реальності

Для імпорту сцени у Unreal Engine необхідно створити новий порожній файл, натиснути Import та обирати файл підготовлений FBX файл.

Для того, щоб додати у робочу зону моделі, заходимо у Content Browser виділяємо усі об'єкти та перетягуємо у сцену, результат представлено на рисунку 3.28.



Рисунок 3.28 – Імпорт у Unreal Engine

В якості освітлення буде створено Sky Light для основного джерела світла та Point Light для імітації світильників. Для цього переходимо до меню Light та перетаскуємо в сцену джерела світла (рис 3.29).

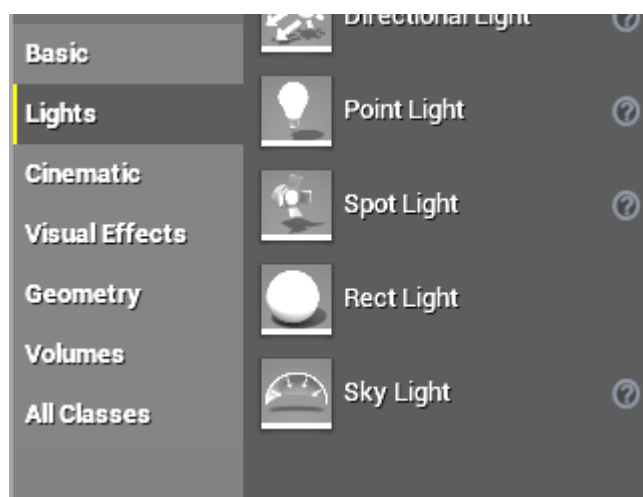


Рисунок 3.29 – Джерела світла у Unreal Engine

У меню Sky Light встановлюємо інтенсивність світла на 10, щоб у сцені був основне світло та встановлюємо галку на Cast Shadows, щоб при прорахунку освітлення об'єкти відкидали тіні (рис 3.30).

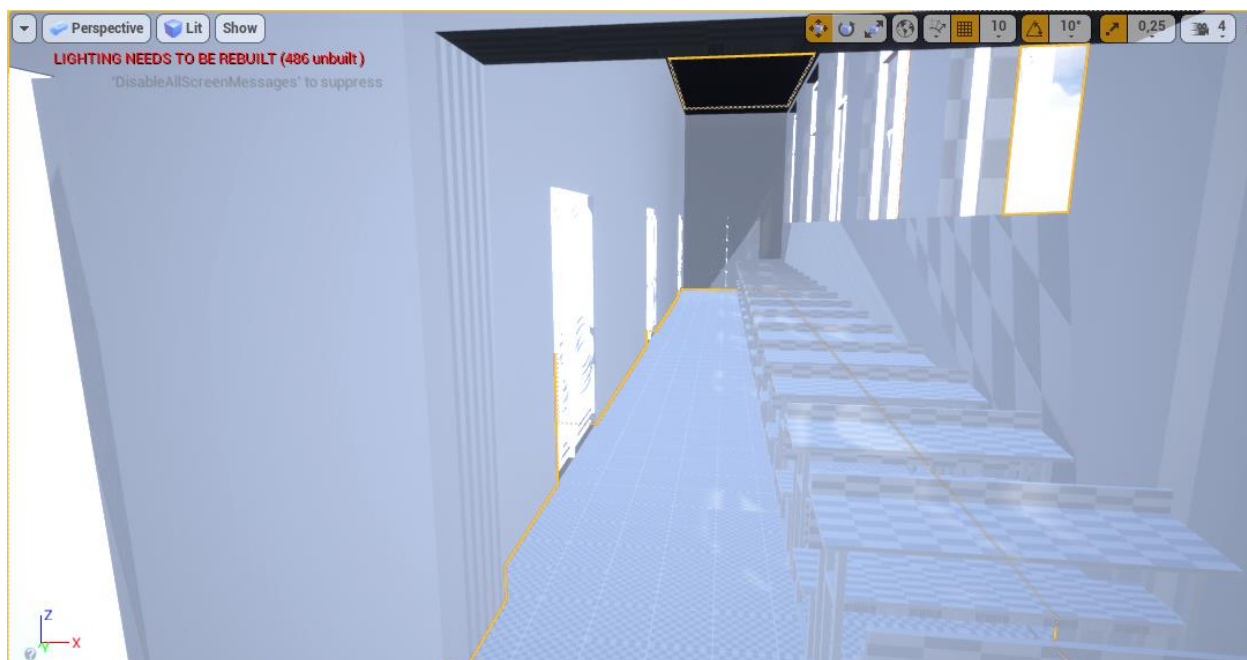


Рисунок 3.30 – Освітлення у Unreal Engine

Для того щоб змінення відображалися потрібно прорахувати освітлення. У навігаційному меню натискаємо Build. Дана операція має можливість прорахувати окремо світло або матеріали та має параметри якості прорахунку та запікання текстур (рис 3.31).

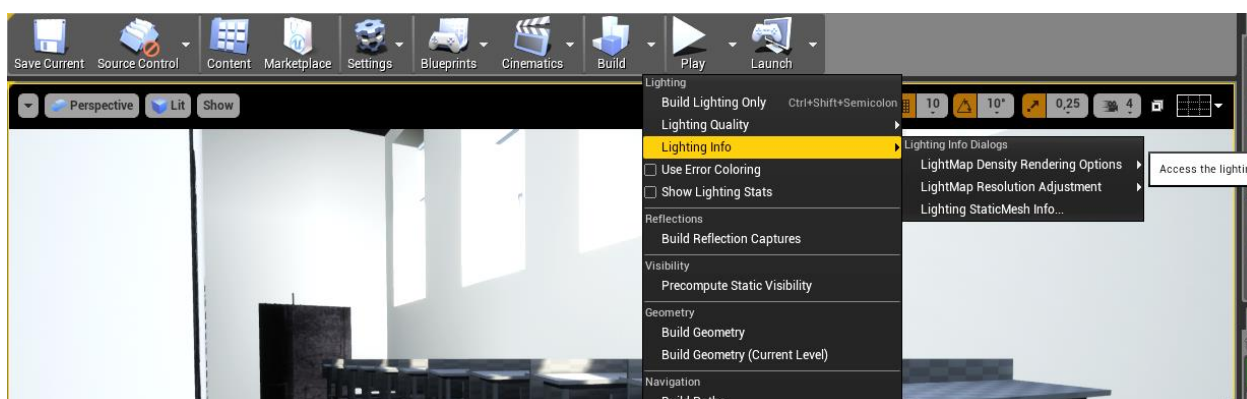


Рисунок 3.31 – Запікання освітлення

Наступним кроком є взаємодія з об'єктами, а саме створення колізій, щоб уникнути проходження крізь моделі та провалювання у текстури. Для цього переходимо до редактора моделі. Натиском правої клавіші миші по об'єкту на який ми встановлюємо колізію та обираємо Edit (рис 3.32).

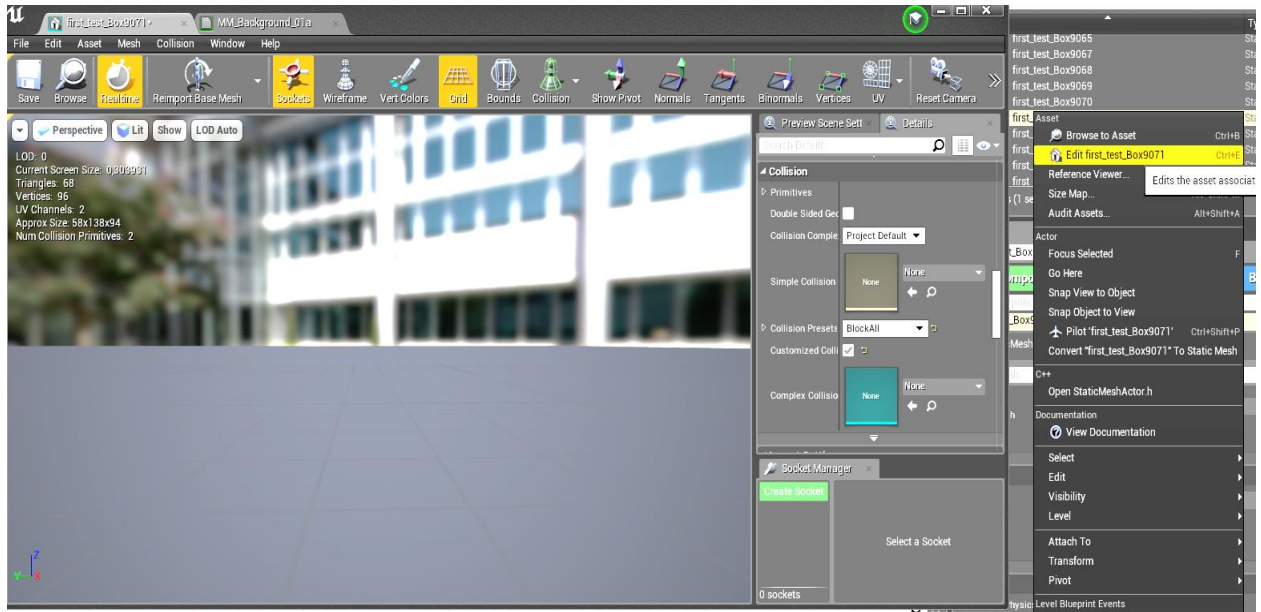


Рисунок 3.32 – Меню Edit моделі

Надалі переходимо до меню Collision та обираємо Add box Simplified Collision, що надасть моделі оболонку через яку не можливо пройти (рис 3.33). Слід зазначити, що даною колізією можна управляти та надавати залежності та виключення, створювати події при яких об'єкт буде пересуватися і тд.

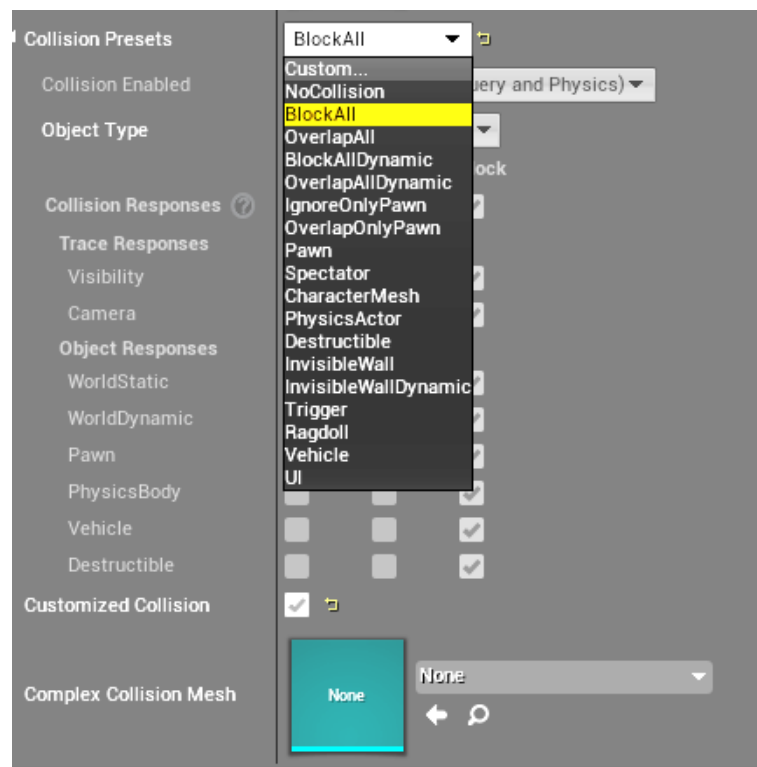


Рисунок 3.33 – Редагування колізії

Наступним кроком потрібно налаштувати матеріали для цього потрібно вибрати модель на якій будемо редагувати матеріал та перейти в меню Textures та відкрити Material Editor. Unreal Engine створить базовий матеріал без текстур та кольору, однак даний матеріал має багато нодів для підключення. Основними нодами є Base Color, Bump, Roughness, Metallic, Normal (рис 3.34).

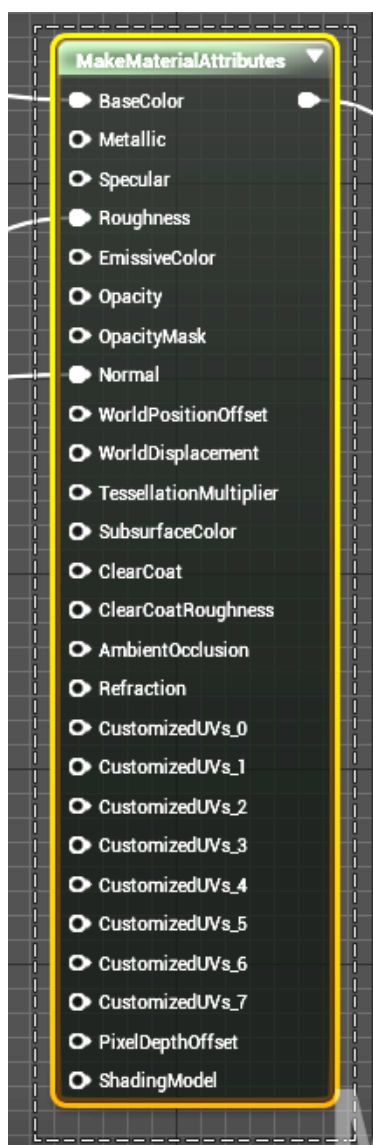


Рисунок 3.34 – Основні параметри матеріалу

Після створення базового матеріалу до редактора потрібно додати текстури та константи для створення матеріалу. Texture Sample має 3 входи на саму текстуру, текстурних координат або кількість повторень текстури та зменшення текстури на певну константу та 6 виходів каналів кольору. Lerp

приймає значення із каналів Texture Sample та використовується для параметру Roughness для надання матовості або глянцу. На рисунках 3.35-3.36 представлено вікно налаштування матеріалу та їх властивості.

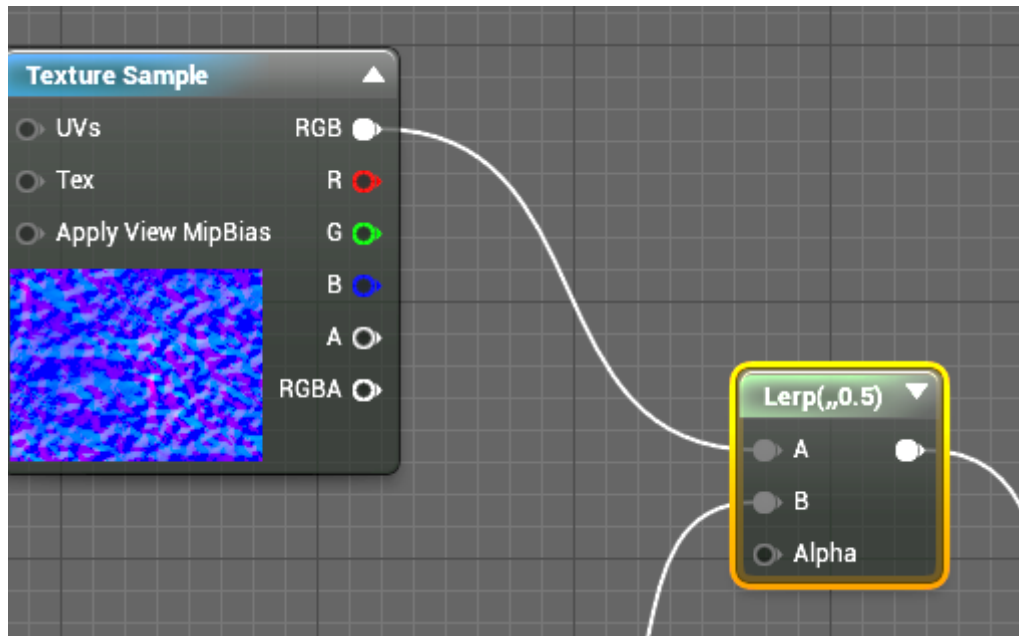


Рисунок 3.35 – Властивості Texture Sample

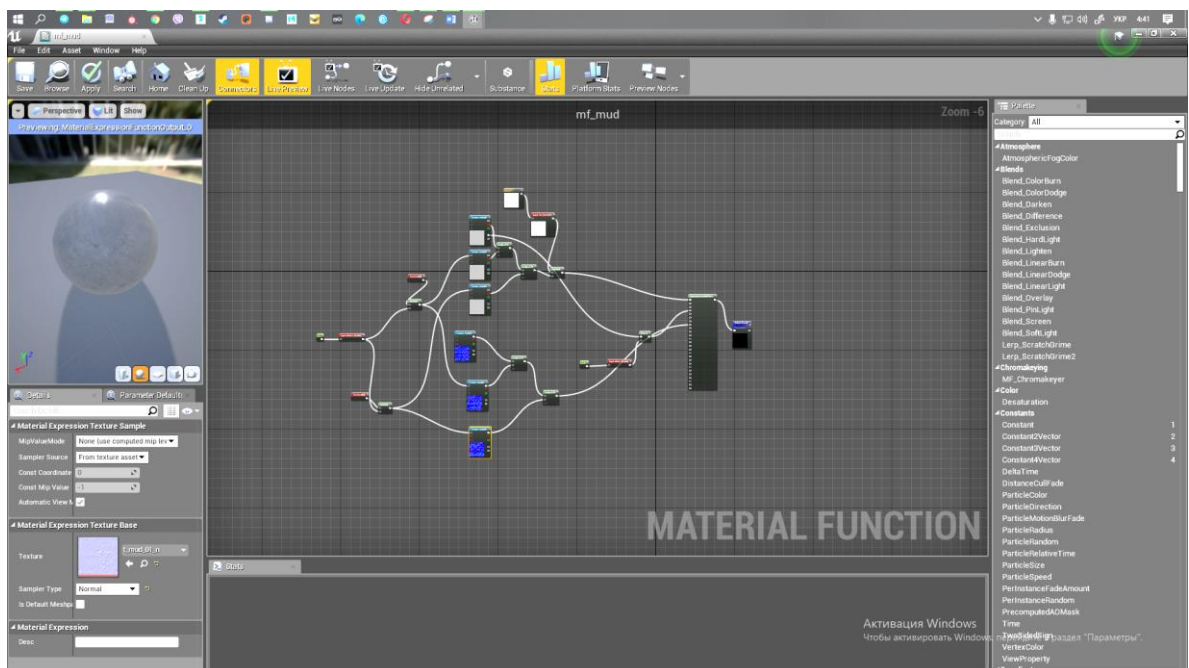


Рисунок 3.36 – Вікно редагування матеріалів

Після завершення редагування матеріалів зберігаємо файл та конвертуємо сцену у HTML5 (рис 3.37). Переходимо до File обираємо Package Project та HTML5.

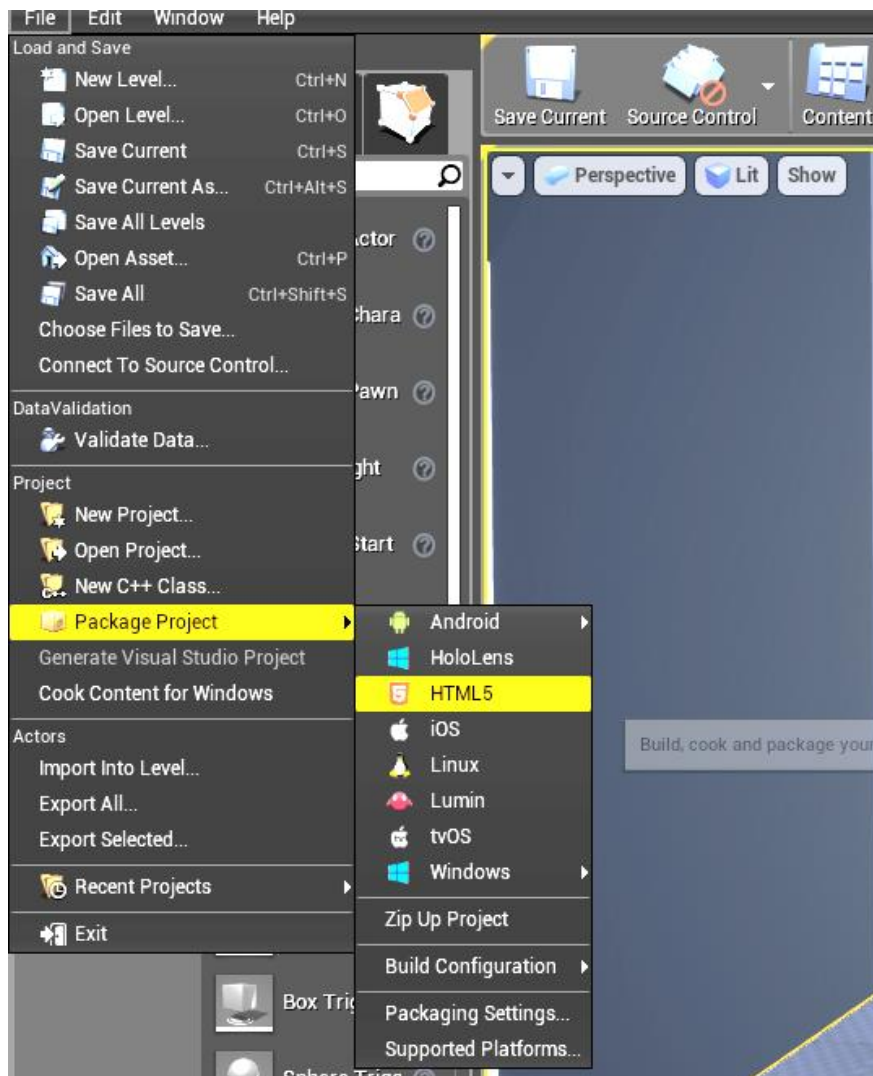


Рисунок 3.37 – Конвертація у Web формат

Після конвертації отримуємо файли, які можна розміщати на сайті. Створюємо html файл для відображення контенту у браузері, та підключаємо отримані javascript файли. Отримані файли представлені на рисунку 3.38.

.htaccess	01.12.2020 20:07	Файл "HTACCESS"	1 КБ
HTML5LaunchHelper	01.12.2020 20:03	Приложение	100 КБ
Readme	01.12.2020 20:07	Текстовый докум...	5 КБ
RunMacHTML5LaunchHelper.command	14.12.2020 6:24	Файл "COMMAND"	1 КБ
testeretetwe	14.12.2020 6:24	CSS-документ	3 КБ
testeretetwe.data	14.12.2020 6:24	Файл "DATA"	746 504 КБ
testeretetwe.data	14.12.2020 6:24	файл JavaScript	8 КБ
testeretetwe	14.12.2020 6:24	Chrome HTML Do...	3 КБ
testeretetwe.UE4	14.12.2020 6:24	файл JavaScript	56 КБ
UE4Game	01.12.2020 20:03	файл JavaScript	772 КБ
UE4Game.wasm	01.12.2020 20:03	Файл "WASM"	84 634 КБ
Utility	14.12.2020 6:24	файл JavaScript	55 КБ

Рисунок 3.38 – Отримані файли

3.4 Тестування розробленої 3D моделі

3.4.1 Тестування на локальному сервері

Для того, щоб протестувати розробку у браузері вводимо `http://localhost:8000/testeretetwe.html`, де 8000 це адреса порта підключення. Після чого на сторінку завантажуються скрипти та контент рисунок 3.39.

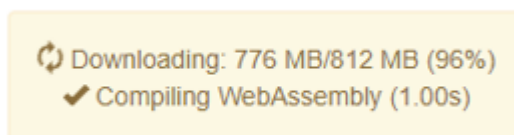


Рисунок 3.39 – Завантаження скриптів

Після завантаження сцени переходимо до Full screen та прогулюємося по приміщенню, перевіряємо об'єкти на колізії та артефакти у моделях (рис 3.40).

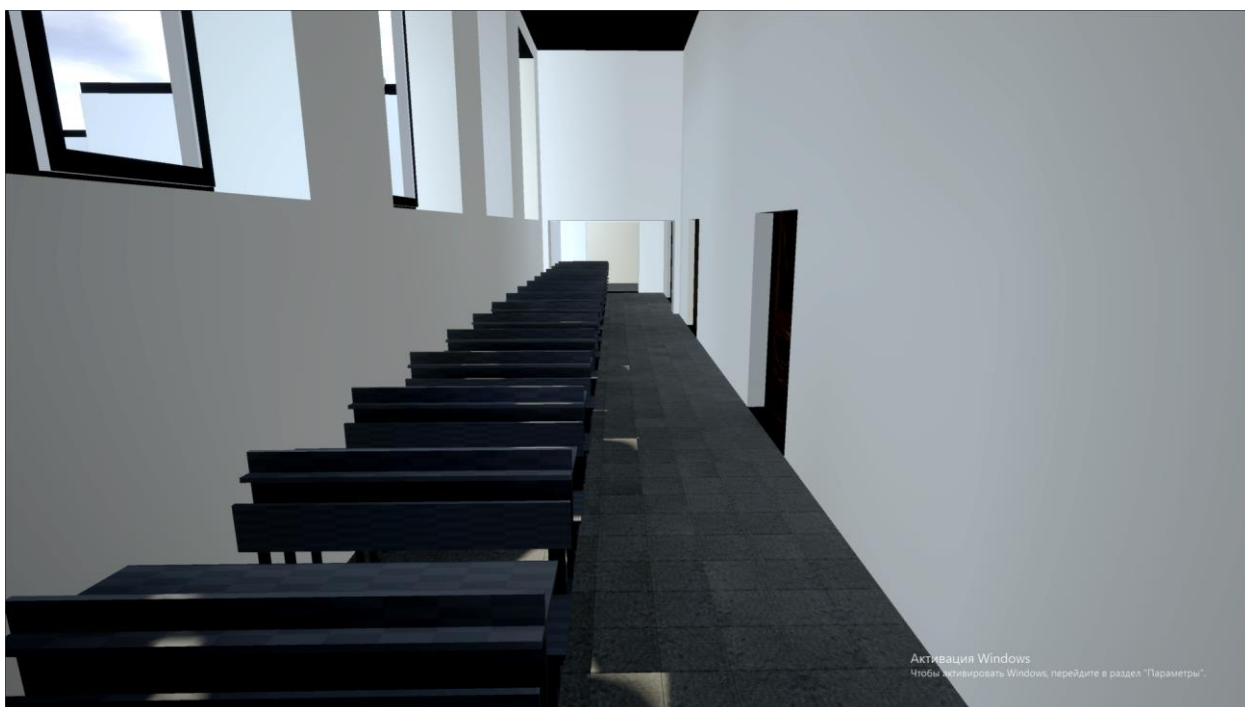


Рисунок 3.40 – Віртуальна модель

В ході тестування приміщення, артефактів знайдено не було, однак слід зазначити, що потрібно додати інтенсивності у джерелах освітлення. Відкриваємо вікно розробника та переходимо до вкладки network та відстежуємо загальну тривалість завантаження web додатку, яка дорівнює 4 секундам. Найбільш вагомим є завантаження та відображення згенерованого контенту. Вікно попереджень вказує на зміну масштабів канвасу та автоматично його перебудовує завдяки адаптації. Надалі слід розглянути швидкість завантаження усіх файлів рисунок 3.41.

Name	Status	Type	Initiator	Size	Time	Waterfall
testeretetwe.html	200	document	Other	2.7 kB	2 ms	
jquery-2.1.3.min.js	200	script	testeretetwe.html	(memory cache)	0 ms	
bootstrap.min.js	200	script	testeretetwe.html	(memory cache)	0 ms	
testeretetwe.css	200	stylesheet	testeretetwe.html	2.4 kB	2 ms	
testeretetwe.UE4.js	200	script	testeretetwe.html	57.2 kB	3 ms	
glyphicons-halflings-regular.woff2	200	font	bootstrap.min.css	(memory cache)	0 ms	
UE4Game.js	200	xhr	testeretetwe.UE4.js:906	791 kB	533 ms	
testeretetwe.data.js	200	xhr	testeretetwe.UE4.js:906	7.4 kB	9 ms	
Utility.js	200	xhr	testeretetwe.UE4.js:906	56.5 kB	10 ms	
testeretetwe.data	200	xhr	testeretetwe.UE4.js:906	764 MB	4.12 s	
UE4Game.wasm	200	xhr	testeretetwe.UE4.js:906	86.7 MB	1.13 s	
blob:http://localhost:8000/1912fd9f-2836-4ad3-b744-418b...	200	script	testeretetwe.UE4.js:977	0 B	3 ms	
blob:http://localhost:8000/d562431b-0206-4588-9098-e40...	200	script	testeretetwe.UE4.js:977	0 B	7 ms	
blob:http://localhost:8000/21779bf8-6b32-480c-9a29-97e...	200	script	testeretetwe.UE4.js:977	0 B	14 ms	

Рисунок 3.41 – Загальна тривалість завантаження

Нижче наведений фрагмент коду, який відповідає за створення canvas

```
function __setLetterbox(element, topBottom, leftRight) {
  if (JSEvents.isInternetExplorer()) {
    element.style.marginLeft = element.style.marginRight = leftRight + "px";
    element.style.marginTop = element.style.marginBottom = topBottom + "px";
  } else {
    element.style.paddingLeft = element.style.paddingRight = leftRight + "px";
    element.style.paddingTop = element.style.paddingBottom = topBottom + "px";
  }
}
```

```
function __hideEverythingExceptGivenElement(onlyVisibleElement) {
  var child = onlyVisibleElement;
  var parent = child.parentNode;
  var hiddenElements = [];
  while (child != document.body) {
    var children = parent.children;
    for (var i = 0; i < children.length; ++i) {
      if (children[i] != child) {
        hiddenElements.push({
          node: children[i],
          displayState: children[i].style.display
        });
        children[i].style.display = "none";
      }
    }
    child = parent;
    parent = parent.parentNode;
  }
  return hiddenElements;
}
```

```
var __restoreOldWindowedStyle = null;
```

```
var __specialEventTargets = [ 0, typeof document !== "undefined" ? document : 0, typeof
window !== "undefined" ? window : 0 ];
```

```
function __findEventTarget(target) {
    warnOnce("Rules for selecting event targets in HTML5 API are changing: instead of using
document.getElementById() that only can refer to elements by their DOM ID, new event target
selection mechanism uses the more flexible function document.querySelector() that can look up
element names, classes, and complex CSS selectors. Build with -s
DISABLE_DEPRECATED_FIND_EVENT_TARGET_BEHAVIOR=1 to change to the new
lookup rules. See https://github.com/emscripten-core/emscripten/pull/7977 for more details.");
    try {
        if (!target) return window;
        if (typeof target === "number") target = __specialEventTargets[target] ||
UTF8ToString(target);
        if (target === "#window") return window; else if (target === "#document") return
document; else if (target === "#screen") return screen; else if (target === "#canvas") return
Module["canvas"];
        return typeof target === "string" ? document.getElementById(target) : target;
    } catch (e) {
        return null;
    }
}
```

```
function __findCanvasEventTarget(target) {
    if (typeof target === "number") target = UTF8ToString(target);
    if (!target || target === "#canvas") {
        if (typeof GL !== "undefined" && GL.offscreenCanvases["canvas"]) return
GL.offscreenCanvases["canvas"];
        return Module["canvas"];
    }
    if (typeof GL !== "undefined" && GL.offscreenCanvases[target]) return
GL.offscreenCanvases[target];
    return __findEventTarget(target);
}
```

Функція яка повертає значення розміру canvas

```
function _emscripten_get_canvas_element_size(target, width, height) {
  var canvas = __findCanvasEventTarget(target);
  if (!canvas) return -4;
  HEAP32[width >> 2] = canvas.width;
  HEAP32[height >> 2] = canvas.height;
}
```

```
function __get_canvas_element_size(target) {
  var stackTop = stackSave();
  var w = stackAlloc(8);
  var h = w + 4;
  var targetInt = stackAlloc(target.id.length + 1);
  stringToUTF8(target.id, targetInt, target.id.length + 1);
  var ret = _emscripten_get_canvas_element_size(targetInt, w, h);
  var size = [ HEAP32[w >> 2], HEAP32[h >> 2] ];
  stackRestore(stackTop);
  return size;
}
```

```
function _emscripten_set_canvas_element_size(target, width, height) {
  console.error("emscripten_set_canvas_element_size(target=" + target + ",width=" +
width + ",height=" + height);
  var canvas = __findCanvasEventTarget(target);
  if (!canvas) return -4;
  canvas.width = width;
  canvas.height = height;
  return 0;
}
```

3.4.2 Тестування на віддаленому сервері

Для тестування на віддаленому сервері потрібно зареєструватися на хостингу та отримати доступ до серверу. Після чого необхідно створити архів, завантажити віддалений сервер та розпакувати. Для такого тесту був обрано

сервіс <https://ru.000webhost.com>. Результат швидкості завантаження на віддаленому сервері представлено на рисунку 3.42.

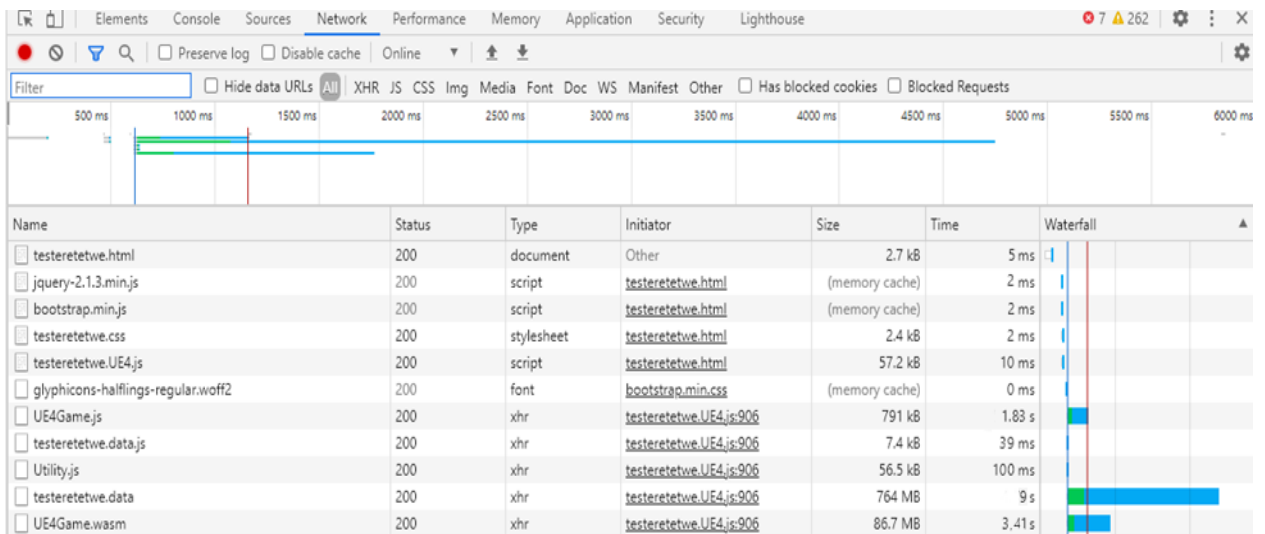


Рисунок 3.42 – Загальна тривалість завантаження на віддаленому сервері

В ході тестування швидкодії на віддаленому сервері, були отримані наступні дані: файл основного контенту завантажувався 9 секунд, на відміну від локального сервера, який завантажив контент за усього за 4 секунди. Було проведено декілька тестів, результати яких фактично не відрізняються. В середньому, швидкість відображення зменшилася у 3.5 рази. Також зменшилася швидкість виконання функцій у файлах javascript (рис 3.43).

```

7
0 function UTF8ToString(ptr, maxBytesToRead) {
1 2.4 ms return ptr ? UTF8ArrayToString(HEAPU8, ptr, maxBytesToRead) : "";
2 }
3
4 function stringToUTF8Array(str, outU8Array, outIdx, maxBytesToWrite) {
5 1.1 ms if (!(maxBytesToWrite > 0)) return 0;
6 var startIdx = outIdx;
7 var endIdx = outIdx + maxBytesToWrite - 1;
8 1.5 ms for (var i = 0; i < str.length; ++i) {
9 2.2 ms var u = str.charCodeAt(i);
0 if (u >= 55296 && u <= 57343) {
1 var u1 = str.charCodeAt(++i);
2 u = 65536 + ((u & 1023) << 10) | u1 & 1023;
3 }
4 if (u <= 127) {
5 if (outIdx >= endIdx) break;
6 0.9 ms outU8Array[outIdx++] = u;
7 } else if (u <= 2047) {
8 if (outIdx + 1 >= endIdx) break;
9 0.5 ms outU8Array[outIdx++] = 192 | u >> 6;
0 0.2 ms outU8Array[outIdx++] = 128 | u & 63;
1 } else if (u <= 65535) {
2 if (outIdx + 2 >= endIdx) break;
3 outU8Array[outIdx++] = 224 | u >> 12;
4 outU8Array[outIdx++] = 128 | u >> 6 & 63;
5 outU8Array[outIdx++] = 128 | u & 63;
6 } else {
7 if (outIdx + 3 >= endIdx) break;
8 if (u >= 2097152) warnOnce("Invalid Unicode code point 0x" + u.toString(16) + " encountered when
9 outU8Array[outIdx++] = 240 | u >> 18;
0 outU8Array[outIdx++] = 128 | u >> 12 & 63;
1 outU8Array[outIdx++] = 128 | u >> 6 & 63;
2 outU8Array[outIdx++] = 128 | u & 63;
3 }
4 }
5 0.1 ms outU8Array[outIdx] = 0;
6 1.2 ms return outIdx - startIdx;
7 }
8
9 function stringToUTF8(str, outPtr, maxBytesToWrite) {
0 0.6 ms assert(typeof maxBytesToWrite == "number", "stringToUTF8(str, outPtr, maxBytesToWrite) is missing 1
1 1.9 ms return stringToUTF8Array(str, HEAPU8, outPtr, maxBytesToWrite);
2 }
3
4 function lengthBytesUTF8(str) {
5 var len = 0;
6 1.4 ms for (var i = 0; i < str.length; ++i) {
7 1.8 ms var u = str.charCodeAt(i);
8 if (u >= 55296 && u <= 57343) u = 65536 + ((u & 1023) << 10) | str.charCodeAt(++i) & 1023;
9 if (u <= 127) ++len; else if (u <= 2047) len += 2; else if (u <= 65535) len += 3; else len += 4;
0 }
1 return len;
2 }

```

Рисунок 3.43 – Швидкість виконання скриптів на віддаленому сервері

Для того, щоб пришвидшити відображення та створення контенту, у файлі `htaccess` необхідно додати кешування та зменшення файлів, увесь код буде розміщуватися в один рядок.

`Mod_expires`

`<ifModule mod_expires.c>`

`ExpiresActive On`

```
ExpiresByType image/x-icon "access plus 7 days"  
ExpiresByType image/jpeg "access plus 7 days"  
ExpiresByType image/png "access plus 7 days"  
ExpiresByType image/gif "access plus 7 days"  
ExpiresByType application/x-shockwave-flash "access plus 7 days"  
  
ExpiresByType text/css "access plus 7 days"  
ExpiresByType text/javascript "access plus 7 days"  
ExpiresByType application/javascript "access plus 7 days"  
ExpiresByType application/x-javascript "access plus 7 days"  
ExpiresByType text/html "access plus 1 day"  
ExpiresByType application/xhtml+xml "access plus 10 minutes"  
</ifModule>
```

В результаті кешування пришвидшило завантаження контенту, однак для більш якісного та швидкого результату потрібен більш потужний сервер.

4 ОХОРОНА ПРАЦІ

Розробка архітектурної візуалізації здебільш проводиться у офісних приміщеннях. Електрична мережа такого приміщення має такі характеристики: трифазна чотирипровідна мережа напругою 380/220 В змінного струму, частота 50 Гц, з глухозаземленою нейтраллю. Такі приміщення відносяться до приміщень без підвищеної небезпеки враження людей електричним струмом згідно з НПАОП 40.1-1.21-98. Усі розетки мають напис «220 В». З кожним робітником проводяться інструктажі з охорони праці відповідно до НПАОП 0.00-4.12-05 які включають у себе первинний повторний та позаплановий інструктаж

Функціональна схема робочого місця представлена на рисунку 4.1.

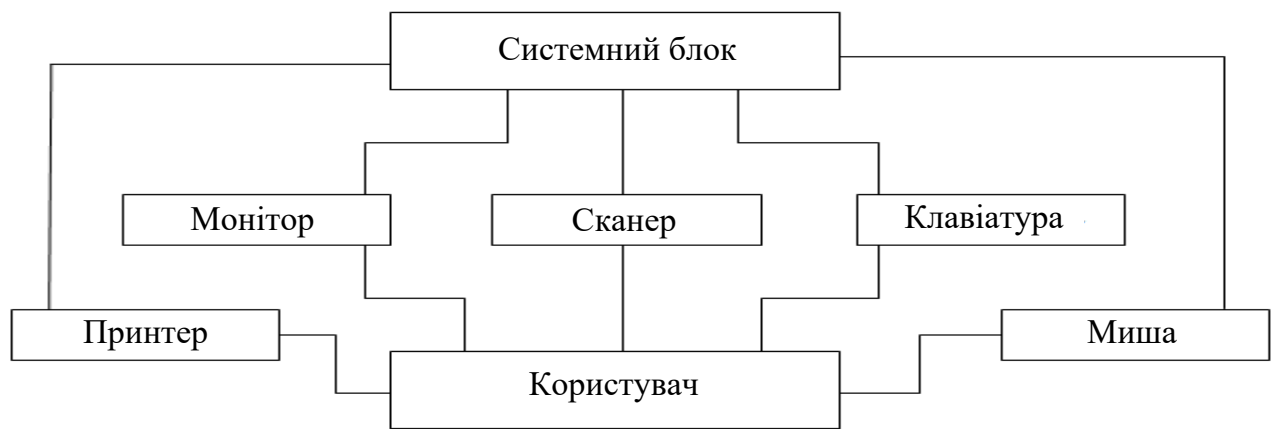


Рисунок 4.1 – Функціональна схема робочого місця

4.1 Вимоги до організації робочих місць

Робоче місце – це зона простору, що оснащена необхідним устаткуванням, де відбувається трудова діяльність одного працівника чи групи працівників.

Гігієнічні вимоги визначають умови життєдіяльності і працездатності людини у процесі взаємодії з технікою і середовищем; показниками є рівень освітлення, температура, вологість, шум, вібрація, токсичність, загазованість тощо.

Фізіологічні та психофізіологічні вимоги визначають відповідність техніки і середовища можливостям працівника щодо сприйняття, переробки інформації, прийняття і реалізації рішень.

Організація робочого місця передбачає:

- правильне розміщення робочого місця у виробничому приміщенні;
- обладнання на робочих місцях;
- урахування особливостей діяльності.

Загальні принципи організації робочого місця:

- на робочому місці повинні бути усі необхідні для роботи предмети, які мають бути поряд із працівником, нічого зайвого на робочому просторі не повинно бути;
- предмети, якими користуються частіше повинні розташовуватися ближче;
- робоче місце має забезпечувати необхідну оглядовість.

Організація робочого місця користувача комп'ютера повинна забезпечувати відповідність усіх елементів робочого місця та їх взаємного розташування ергономічним вимогам [44].

4.2 Вимоги до електробезпеки

Приміщення із робочими місцями користувачів комп'ютерів для забезпечення електробезпеки обладнання, а також для захисту від ураження електричним струмом самих користувачів ПК повинні мати достатні технічні засоби захисту відповідно до НПАОП (Нормативно-правові акти з охорони праці) 40.1-1.07-01 «Правила експлуатації електрозахисних засобів», НПАОП

40.1-1.21-98 «Правила безпечної експлуатації електроустановок споживачів», НПАОП 40.1-1.32-01 «Правила будови електроустановок. Електрообладнання спеціальних установок».

З метою запобігання ушкодженням, що можуть статися через ураження електричним струмом, загоряння, коротке замикання тощо, розроблено загальний стандарт безпеки ІЕС 950. В приміщенні встановлюється аварійний резервний вимикач, який повністю вимкає електричне живлення приміщення, крім освітлення.

ПЕОМ, периферійні пристрої ПЕОМ та устаткування для обслуговування, ремонту та налагодження ЕОМ повинні підключатися до електромережа тільки з допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення. Штепсельні з'єднання та електророзетки крім контактів фазового та нульового робочого провідників повинні мати спеціальні контакти для підключення нульового захисного провідника. Конструкція їх має бути такою, щоб приєднання нульового захисного провідника відбувалося раніше ніж приєднання фазового та нульового робочого провідників. Порядок роз'єднання при відключенні має бути зворотним. Необхідно унеможливити з'єднання контактів фазових провідників з контактами нульового захисного провідника.

Неприпустимим є підключення ПЕОМ та периферійних пристроїв ПЕОМ до звичайної двопровідної електромережі, в тому числі – з використанням перехідних пристроїв.

Електромережу штепсельних розеток для живлення ПЕОМ, периферійних пристроїв ПЕОМ при розташуванні їх уздовж стін приміщення прокладають по підлозі поряд зі стінами приміщення, як правило, в металевих трубах і гнучких металевих рукавах з відводами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання.

При розташуванні в приміщенні за його периметром до 5 ПЕОМ, використанні трипровідникового захищеного проводу або кабелю в оболонці

з негорючого або важкозаймистого матеріалу дозволяється прокладання їх без металевих труб та гнучких металевих рукавів.

Електромережу штепсельних розеток для живлення ПЕОМ при розташуванні їх у центрі приміщення, прокладають у каналах або під знімною підлогою в металевих трубах або гнучких металевих рукавах. При цьому не дозволяється застосовувати провід і кабель в ізоляції з вулканізованої гуми та інші матеріали, що містять сірку. Відкрита прокладка кабелів під підлогою забороняється. Металеві труби та гнучкі металеві рукави повинні бути заземлені. Заземлення повинно відповідати вимогам НПАОП 40.1-1.21-98.

Для підключення переносної електроапаратури застосовують гнучкі проводи в надійній ізоляції.

Тимчасова електропроводка від переносних приладів до джерел живлення виконується найкоротшим шляхом без заплутування проводів у конструкціях машин, приладів та меблях. Доточувати проводи можна тільки шляхом паяння з наступним старанним ізолюванням місць з'єднання.

Є неприпустимими:

- експлуатація кабелів та проводів з пошкодженою або такою, що втратила захисні властивості за час експлуатації, ізоляцією;
- застосування саморобних подовжувачів, які не відповідають вимогам ПВЕ до переносних електропроводок;
- застосування для опалення приміщення нестандартного (саморобного) електронагрівального обладнання або ламп розжарювання;
- користування пошкодженими розетками, вимикачами та іншими електровиробами, а також лампами, скло яких має сліди затемнення або випинання;
- підвішування світильників безпосередньо на струмопровідних проводах, обгортання електроламп і світильників папером, тканиною та іншими горючими матеріалами;
- використання електроапаратури та приладів в умовах, що не відповідають рекомендаціям підприємств-виготовлювачів [45].

4.3 Охорона праці в системі "людина-комп'ютер".

Надійність системи "людина-комп'ютер" значною мірою визначається функціональним станом людини. Психофізіологічні та емоційні перенапруження, втома людини-оператора можуть призвести в комп'ютеризованих системах керування до помилок і як наслідок – до значних економічних втрат.

Згідно зі статистичними даними від 40 до 75 % аварій літаків зумовлено людським фактором. Відмови комп'ютеризованої системи керування рухом залізничного транспорту, на гірничо-збагачувальних комбінатах з вини операторів становлять понад 50 % їх загальної кількості, причому значна їх частина спричинена невідповідністю функціонального стану оператора складності виконуваної роботи.

Помилки працівників, що працюють з комп'ютером в адміністративно-управлінській сфері, викликають, звісно, менші за масштабами наслідки. Проте незадовільний функціональний стан користувачів комп'ютерів може викликати небажані наслідки (професійні та професійно зумовлені захворювання), що також пов'язано зі значними соціальними та економічними втратами враховуючи стрімке зростання кількості комп'ютеризованих робочих місць.

Визначення та вивчення факторів, що впливають на функціональний стан користувачів комп'ютерів дозволить виділити основні причини виникнення станів напруженості, стомлення, стресу і здійснити відповідні профілактичні заходи.

Трудова діяльність користувачів комп'ютерів відбувається у певному виробничому середовищі, яке впливає на їх функціональний стан. Найбільш значні – фізичні фактори виробничого середовища, до яких належать електромагнітні хвилі різних частотних діапазонів, електростатичні поля, шум, параметри мікроклімату та ціла низка світлотехнічних показників.

Фактори, що впливають на функціональний стан операторів ПК представлено на рис. 4.2.

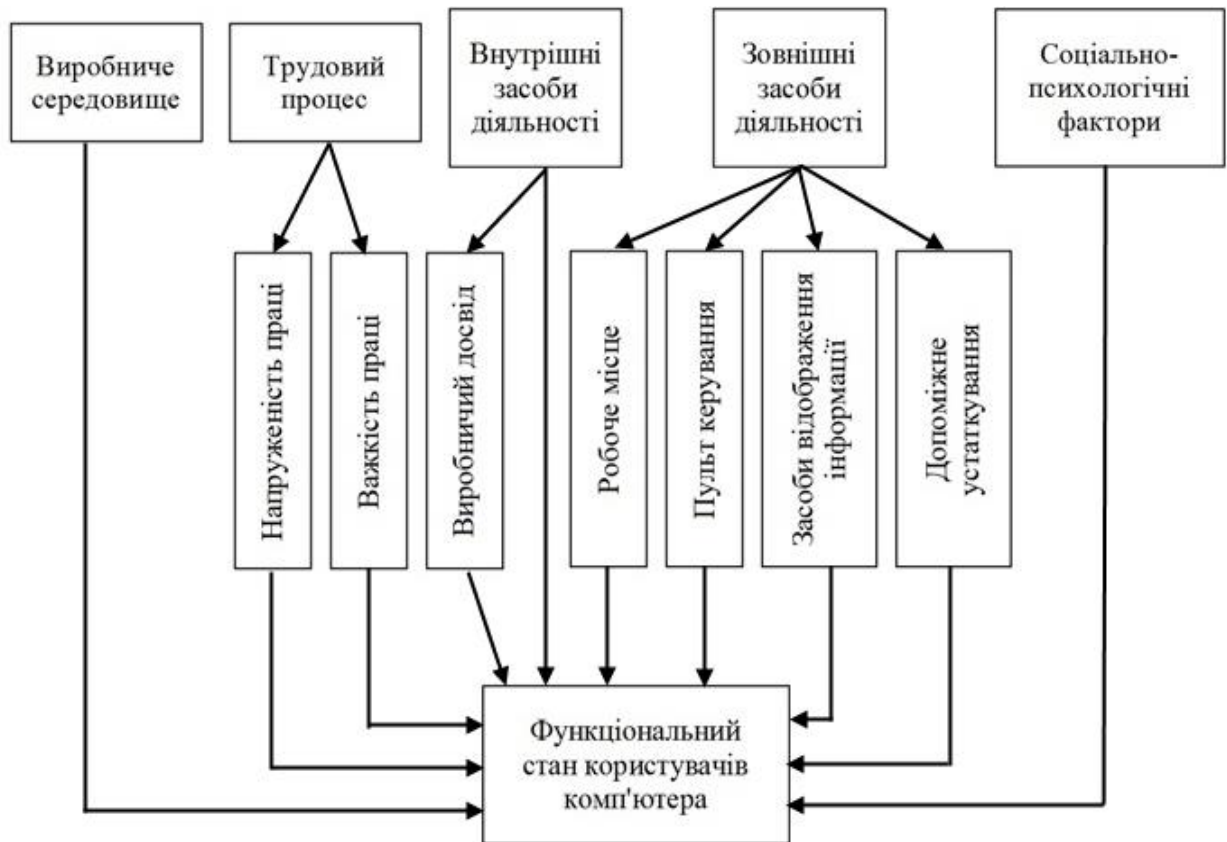


Рисунок. 4.2 – Схема факторів які впливають на функціональний стан операторів ПК [46]

ВИСНОВКИ

В ході реалізації магістерської атестаційної роботи були проаналізовані можливості використання ігрової технології Unreal, яка застосовується у архітектурній візуалізації для розв'язання завдання 3D моделювання виробничого приміщення виконання. Для даної роботи необхідно мати знання 3D-моделювання:

- моделювання об'єктів з правильно створеною геометричною сіткою;
- текстурування;
- освітлення.

Також проаналізувати методи взаємодії між ігровим двигуном та 3D редактором. Основною проблемою було правильно імпортувати геометрію до Unreal, тому, що усі об'єкти повинні мати розгортку для запікання освітлення та також повинні бути цільною геометрією. Також слід зазначити, що розробка проводилась у версії 4.23 яка на даний момент дозволяє конвертувати сцену у web додаток, у наступних версіях Unreal цей функціонал було видалено, однак був розроблено плагін Datasmith, який дозволяє легко імпортувати усю сцену незалежно від обраної системи рендерінгу.

Дане функціональне рішення дозволить проводити віртуальні екскурсії з повною свободою пересування в сцені на відміну від рішень панорамних 360° фотографій, користувач має повний контроль пересування та має цілковите уявлення про приміщення. Також дане рішення надає можливість створення футуристичних музеїв або інших приміщень.

Наукова новизна полягає в розробці інформаційної моделі процесу проектування віртуальних моделей виробничих приміщень та окремих моделей віртуальної реальності.

Подальшим розвитком отриманих результатів може бути додавання VR функціоналу, щоб користувач мав повний ефект занурення та відчуття

присутності. Однак не кожен сервер може обробляти таку кількість даних при цьому працюючи без втрати кадрів.

Таким чином в результаті виконання магістерської роботи були досліджені сучасні методи формування доповненої та віртуальної реальності, розроблена віртуальна модель приміщення у 3D редакторі 3D Max, проведена оптимізація моделей за допомогою зменшення кількості полігонів при побудові моделі, була сформована віртуальна 3D модель у ігровому двигуні Unreal Engine, згенеровано освітлення та текстури. На базі отриманої моделі був розроблено та протестовано Web додаток для формування доповненої реальності на сайті. Дана атестаційна робота оформлена відповідно усім стандартам ДСТУ 3008-2015 та методичних вказівок.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008: 2015 Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлення. - К.: ДП «УкрНДНЦ», 2016. - 31 с.

2. Невлюдов, І.Ш. Дипломне проектування для студентів усіх форм навчання спеціальностей 151 «Автоматизація та комп'ютерно-інтегровані технології» [Текст]: навч. посіб. / І.Ш. Невлюдов, А.О. Андрусевич, О.В. Токарева, Г.В. Пономарьова. – Київ – 58, пр. Космонавта Комарова, 1, 2016. – 320 с.

3. Методичні вказівки з «Розробки й оформлення магістерської атестаційної роботи» для студентів другого (магістерського) рівня вищої освіти галузі знань 15 Автоматизація та приладобудування за спеціальністю 151 Автоматизація та комп'ютерно-інтегровані технології освітні програми: «Автоматизоване управління технологічними процесами», «Комп'ютерно-інтегровані технологічні процеси і виробництва», «Комп'ютеризовані та робототехнічні системи» / Упоряд. І.Ш. Невлюдов, В.В. Косенко, В.В. Євсєєв. – Харків: ХНУРЕ, 2019. – 55 с.

4. Положення про протидію академічному плагіату в ХНУРЕ [Електронний ресурс] / Режим доступу: [www/ URL: https://nure.ua/wp-content/uploads/Main_Docs_NURE/Polozhennya-pro-protidiyu-akademichnomu-plagiatu-v-HNURE----290-vid-28.04.2017.pdf](http://www.nure.ua/wp-content/uploads/Main_Docs_NURE/Polozhennya-pro-protidiyu-akademichnomu-plagiatu-v-HNURE----290-vid-28.04.2017.pdf) - 29.08.2019р. - Загл.с екрана

5. АВТОМАТИЗАЦІЯ ТА ПРИЛАДОБУДУВАННЯ («Automation and Development of Electronic Devices» ADED-2020) [Електронний ресурс] : збірник студентських наукових статей / Харківський національний університет радіоелектроніки ; редкол.: І.Ш. Невлюдов та ін. – Харків : ХНУРЕ, 2020. – Вип. 1. – 232.

6. Комп'ютерна анімація : навчальний посібник для студентів напряму підготовки 6.051501 "Видавничо-поліграфічна справа" / О. С. Євсєєв. – Х.: Вид. ХНЕУ ім. С. Кузнеця, 2014. – 152 с. (Укр. мов.).

7. Розвиток VR технологій [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.avclub.pro/articles/3d-tehnologii/kratkaya-istoriya-razvitiya-tehnologii-virtualnoy-realnosti> – 26.11.2020
8. Опис віртуальної, доповненої, змішаної реальності [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.forbes.com/sites/quora/2018/02/02/the-difference-between-virtual-reality-augmented-reality-and-mixed-reality/> – 26.11.2020
9. Віртуальна реальність у освіті. [Електронний ресурс]. Доступно: https://vrgeek.ru/2016/07/21/2467_obrazovanie-v-vr. – 26.11.2020
10. Юхвид А. В. Философские проблемы виртуальной реальности в творчестве, искусстве и образовании. Правовые аспекты использования виртуальных технологий [Електронний ресурс]. Доступно: http://www.yukhvid.narod.ru/Doklad_Ekaterinburg.htm.
11. Форми освіти у віртуальній реальності [Електронний ресурс]. Доступно: <http://www.edutainme.ru/post/vr-formats>. – 26.11.2020
12. Різниця доповненої та змішаної реальності [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.forbes.com/sites/quora/2018/02/02/the-difference-between-virtual-reality-augmented-reality-and-mixed-reality/?sh=431614952d07> – 26.11.2020.
13. Системи віртуальної реальності (Virtual Environment & Virtual Reality). [Електронний ресурс]. Доступно: <http://ve-group.ru/vr-systems>.
14. Полігональне моделювання [Електронний ресурс]. – 2016. – Режим доступу до ресурса: <https://klona.ua/blog/3d-modelirovanie/poligonalnoe-modelirovanie-znachenie-osobennosti-rekomendacii-v-rabote>. – 26.11.2020.
15. Застосування 3D сканерів [Електронний ресурс]. – Режим доступу до ресурсу: <https://blog.iqb.ru/3d-scanning-kazan-2017/> – 27.11.2020.
16. Методи сканування 3D сканерів [Електронний ресурс]. – Режим доступу до ресурсу: <https://vektor.us.ru/blog/3d-skaner.html> – 27.11.2020.

17. Принцип дії 3D сканерів [Електронний ресурс]. – Режим доступу до ресурсу:<https://make-3d.ru/articles/chto-takoe-3d-skaner-i-kak-on-rabotaet/> – 9.12.2020.
18. Програмні рішення для 3D сканерів [Електронний ресурс]. – Режим доступу до ресурсу:<https://blog.iqb.ru/3d-software/> – 27.11.2020.
19. Опис ігрового двигуна Unity [Електронний ресурс]. – Режим доступу до ресурсу:<https://cubiq.ru/dvizhok-unity/> – 27.11.2020.
20. Реалізація доповненої реальності у Unity [Електронний ресурс]. – Режим доступу до ресурсу:<https://learn.unity.com/tutorial/setting-up-ar-foundation#5da50119edbc2a0acad677d9> – 28.11.2020.
21. Офіційна документація Unreal Engine про Blueprints [Електронний ресурс]. – Режим доступу до ресурсу:<https://docs.unrealengine.com/en-US/Engine/Blueprints/index.html> – 28.11.2020.
22. Офіційна документація Unreal Engine про Blueprints [Електронний ресурс] – Режим доступу до ресурсу <https://docs.unrealengine.com/enUS/Engine/Blueprints/Scripting/index.html> – 28.11.2020.
23. Unreal Engine VR для розробників Книга, Митч Макеффри
24. Шолом віртуальної реальності Samsung Gear VR [Електронний ресурс]. – Режим доступу до ресурсу:<https://www.ixbt.com/mobile/samsung-gear-vr.shtml> – 28.11.2020.
25. Шолом віртуальної реальності HTC [Електронний ресурс]. – Режим доступу до ресурсу:<https://www.ixbt.com/mobile/htc-vive-preview.shtml> – 29.11.2020.
26. Сучасна розробка віртуальних екскурсій Google [Електронний ресурс]. – Режим доступу до ресурсу:<https://www.buro247.ru/news/culture/1-oct-2019-google-versailles-vr-tour.html> – 29.11.2020.
27. Проект vAcademia [Електронний ресурс]. – Режим доступу до ресурсу:<https://www.startbase.ru/projects/28/view/> – 29.11.2020.

28. Перенос Вінниці у 3D[Електронний ресурс]. – Режим доступу до ресурсу:http://impuls.vntu.edu.ua/index.php?option=com_content&view=article&id=2804%3A3-d-&catid=9%3A2014-02-09-08-50-23&lang=en – 29.11.2020.

29. Система освіти Lecture VR[Електронний ресурс]. – Режим доступу до ресурсу:<https://www.linkedin.com/pulse/lecture-vr-virtual-reality-learning-platform-oculus-david-whelan> – 29.11.2020.

30. Напрямки розвитку платформи VR [Електронний ресурс]. – Режим доступу до ресурсу:<https://habr.com/ru/company/mailru/blog/339686/> – 29.11.2020.

31. VR додаток Discovery[Електронний ресурс]. – Режим доступу до ресурсу:<http://www.virtualrealityreviewer.com/virtual-reality-network-launched-by-discovery/> – 29.11.2020.

32. Конспект лекцій з дисципліни «Теорія 3D моделювання». Для студентів спеціальностей: 131 Прикладна механіка - освітня програма «Технології машинобудування»; 133 Галузеве машинобудування - освітня програма «Металорізальні верстати та системи» усіх форм навчання / Укл.: М.В.Фролов – ЗНТУ, 2018. – 94 с.

33. Опис програми 3D моделювання 3DSMAX [Електронний ресурс]. – Режим доступу до ресурсу:<https://www.autodesk.ru/products/3ds-max/features?plc=3DSMAX&term=1-YEAR&support=ADVANCED&quantity=1> – 30.11.2020.

34. Опис програми 3D моделювання 3DSMAX [Електронний ресурс]. – Режим доступу до ресурсу:<https://3ddevice.com.ua/blog/3d-printer-obzor/obzor-3ds-max/> – 30.11.2020.

35. Опис модифікаторів та способів моделювання у 3DSMAX [Електронний ресурс]. – Режим доступу до ресурсу:<https://3dmax-online.ru/selfeducation/3d-max-modifikatory> – 30.11.2020.

36. Опис програми 3D моделювання Blender[Електронний ресурс]. – Режим доступу до ресурсу:<https://3ddevice.com.ua/blog/3d-printer-obzor/3d-redaktor-blender-obzor> – 02.12.2020.

37. Опис можливостей Blender [Електронний ресурс]. – Режим доступу до ресурсу:<https://docs.blender.org/manual/ru/dev/> – 03.12.2020.

38. Система рендерінгу Cycles [Електронний ресурс]. – Режим доступу до ресурсу:<https://docs.blender.org/manual/ru/dev/render/cycles/index.html> – 01.12.2020.

39. Система рендерінгу Eevee [Електронний ресурс]. – Режим доступу до ресурсу:<https://blender3d.com.ua/blender-2-8-eevee> – 03.12.2020.

40. Система рендерінгу Corona Render [Електронний ресурс]. – Режим доступу до ресурсу:<https://repetitor3d.ru/corona/sozдание-i-nastrojka-materialov> – 9.12.2020.

41. Система рендерінгу V-ray [Електронний ресурс]. – Режим доступу до ресурсу:<http://vraydoc.narod.ru/vray220/index.htm> – 03.12.2020.

42. Маккоу, А. Веб-приложения на JavaScript / Алекс Маккоу - СПб.: Питер, 2012. - 288 с.

43. НПАОП 0.00-1.31-99. Правила охорони праці під час експлуатації електронно-обчислювальних машин // ДНАОП: електронна версія. URL: https://dnaop.com/html/245_3.html – 9.12.2020.

44. Ергономічні вимоги до організації робочих місць. [Електронний ресурс]. – Режим доступу до ресурсу: https://pidruchniki.com/14821111/bzhd/ergonomichni_vimogi_organizatsiyi_robichih_mists – 9.12.2020.

45. Гандзюк М.П., Желібо Є.П., Халімовський М.О. Основи охорони праці. К., 2010. – 324с

46. . Невлюдов И.Ш., Цымбал А.М., Милютин С.С. Интеллектуальное проектирование технологических процессов роботизированной сборки. – Харьков: НТМТ, 2010. – 206 с.