

Міністерство освіти і науки України  
Кам'янець-Подільський національний університет імені Івана Огієнка  
Фізико-математичний факультет  
Кафедра комп'ютерних наук

Дипломна робота  
магістра

з теми: **«РОЗРОБКА РЕДАКТОРА ГЕНЕРУВАННЯ ІНТЕРАКТИВНИХ  
ТРИВИМІРНИХ СЦЕН З ВИКОРИСТАННЯМ СТВОРЕНИХ  
ГРАФІЧНИХ МОДЕЛЕЙ»**

Виконав: студент 2 курсу,  
групи KN1-M21  
спеціальності 122 Комп'ютерні науки  
**Дуцик Андрій Андрійович**

Керівник: **Смалько О. А.**,  
кандидат педагогічних наук, доцент,  
доцент кафедри комп'ютерних наук

Рецензент: **Кух А.М.**,  
доктор педагогічних наук, доцент,  
професор кафедри фізики

Кам'янець-Подільський – 2022

## ЗМІСТ

ВСТУП.....	3
РОЗДІЛ I. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	5
1.1. Поняття і класифікація тривимірних моделей, тривимірної графіки та її створення.....	5
1.2. Огляд програмних платформ створення застосунків для роботи з інтерактивними тривимірними сценами .....	10
РОЗДІЛ II. АНАЛІЗ ЗАСОБІВ ДЛЯ СТВОРЕННЯ РЕДАКТОРА.....	24
2.1. Процес створення тривимірних моделей.....	24
2.2. Інтерфейс та архітектура ігрового рушія Unreal Engine .....	27
РОЗДІЛ III. РОЗРОБКА РЕДАКТОРА.....	32
3.1. Проектування редактора та тривимірних моделей для нього.....	32
3.2. Створення тривимірних моделей .....	36
3.3. Реалізація інтерфейсу користувача та функціоналу генерування моделей.....	59
ВИСНОВКИ.....	67
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	71
ДОДАТКИ.....	74
Додаток А. Порівняння функціональних можливостей поширених редакторів тривимірної графіки .....	75
Додаток Б. Визначення застосунку для створення тривимірних моделей та рушія для реалізації редактора.....	88
Додаток В. Приклади створених елементів інтерфейсу та фрагменти програмного коду до них.....	96
Додаток Г. Опис реалізації функціоналу програми мовою програмування Blueprints .....	107
Додаток Д. Робоче вікно застосунку та приклади генерування сцени.....	118

## ВСТУП

**Актуальність теми дослідження.** Сьогодні висуває підвищені вимоги до якості медіаконтенту, стрімкими темпами розвиваються технології відтворення цифрової тривимірної графіки, безперервно покращуються функціональні можливості застосунків, за допомогою яких вона створюється, а тому зростає потреба у спеціалістах, що вміють фахово працювати з нею. Робота з 3D-графікою – справа непроста. І хоч зацікавлених людей працювати у цій популярній та високооплачуваній сфері чимало, все ж вміння та навички формуються досить довго, спираючись на прикладання значних зусиль, застосування геометричних знань і розвиненого просторового мислення. Задля програмної підтримки користувачів, що бажають опанувати ази розробки та ефективного використання мультимедійних об'єктів, останнім часом створюються різноманітні редактори та сервіси їх генерування та опрацювання. Розробка подібного застосунку, орієнтованого на роботу з тривимірними об'єктами, проводилась впродовж даного дослідження.

**Предмет дослідження:** методи та засоби розробки інтерактивних 3D-застосунків.

**Об'єкт дослідження:** технології тривимірної комп'ютерної графіки.

**Мета роботи:** створення редактора для генерування інтерактивних тривимірних сцен, який може бути корисним для всіх, хто опановує основи 3D-моделювання.

Для реалізації мети виконувались наступні **завдання:**

- Аналіз і порівняння функціональних можливостей поширених редакторів тривимірної графіки.
- Дослідження доступних програмних платформ, придатних для реалізації інтерактивного тривимірного редактора.

- Проектування редактора та вибір візуального спрямування тривимірних моделей для подальшого використання у генерації сцени.
- Створення тематичних тривимірних моделей.
- Розробка зручного та інтуїтивно зрозумілого інтерфейсу користувача.
- Реалізація ефективного функціоналу програмного застосунку для генерування тривимірних сцен на базі створених 3D-моделей.

**Методи дослідження:** аналіз, синтез, порівняння, перевірка і контроль проєктних рішень, валідація дизайн-ідей та програмних застосунків, експеримент.

**Практичне значення:** результати проведеного теоретичного дослідження можуть бути корисними для всіх, хто починає працювати у сфері тривимірної графіки; реалізований проєкт редактора генерування тривимірних сцен може знадобитися майбутнім дизайнерам і розробникам комп'ютерних ігор для формування початкових навичок створення ігрового контенту; робота з розробленим авторським програмним застосунком сприятиме формуванню у всіх бажаючих початкового досвіду 3D-моделювання, базових вмінь конструювання тривимірних об'єктів, навичок моделювання поверхонь складних форм.

**Апробація результатів.** За темою магістерського дослідження подано статтю «Використання редактора генерування інтерактивних тривимірних сцен у процесі формування навичок роботи з графічними моделями» до Вісника Кам'янець-Подільського національного університету імені Івана Огієнка. Фізико-математичні науки.

**Структура роботи.** Дипломна робота складається з вступу, трьох розділів, висновків, списку використаних джерел та додатків.

## РОЗДІЛ І. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1. Поняття і класифікація тривимірних моделей, тривимірної графіки та її створення

Тривимірна графіка надійно увійшла в сфери людської діяльності, починаючи від архітектурних візуалізацій і закінчуючи комп'ютерними іграми. На 3D-зображеннях можна побачити об'ємні фігури, а процес їх створення - це 3D-моделювання за допомогою комп'ютерних програм. Тривимірна графіка стає дедалі популярнішою завдяки тому, що проекти, які виконуються таким методом, стають набагато наочнішими та реалістичнішими, ніж звичайні креслення і двовірні зображення. 3D-модель - це об'ємна фігура в просторі, що створюється в спеціальному редакторі. За основу, як правило, беруть креслення, фотографії, малюнки і докладні описи, спираючись на які і створюють віртуальну модель.

Створення 3D-моделі об'єкта здійснюється за допомогою процесу 3D-моделювання. На першому етапі 3D-моделювання проводиться збір інформації: ескізи, креслення, фотографії та відеоролики, малюнки, часто навіть використовують готовий зразок виробу - загалом, все, що допоможе зрозуміти зовнішній вигляд і структуру об'єкта. На підставі отриманої інформації 3D-моделлер або 3D-дизайнер створює тривимірну модель у спеціальній комп'ютерній програмі. Після того як модель буде виконана, на неї можна буде подивитися з будь-якого ракурсу, наблизити, віддалити, внести необхідні коригування. Сама по собі модель вже готова для подальшого використання - друку на 3D-принтері, 3D-фрезерування на верстатах з ЧПУ або будь-якого іншого методу прототипування. Якщо ж необхідно отримати зображення для наочності клієнтам і споживачам, для рекламних роликів або для презентації, то виконується візуалізація - створення віртуальної комп'ютерної моделі в кінцевому вигляді. Візуалізація передбачає вибір і налаштування матеріалів, текстури,

освітлення, тіні, зовнішні умови навколишнього середовища. Після всіх налаштувань здійснюється фінальний прорахунок, рендеринг.

Існує кілька видів тривимірних моделей:

- полігональна модель;
- NURBS поверхні.

NURBS мають вищий рівень точності, тож їх найчастіше використовують інженери, машинобудівники та архітектори. Полігональні моделі частіше використовують для створення 3D-зображень у мультиплікації, кінематографі та комп'ютерних іграх. Вони складаються з численних найпростіших геометричних фігур, які також називають примітивами.

Є три види 3D-моделювання:

- каркасне моделювання;
- поверхневе моделювання;
- твердотільне моделювання.

Перший із них, найпростіший вид - це каркасне моделювання. Моделі, одержувані при створенні цього типу відтворення, будуть називатися дротяними або каркасними. Складаються вони з ліній, дуг і сегментів. Зображення такого типу не передають повну інформацію про об'єкт: ні про об'єм, ні про структуру поверхні з такої моделі дізнатися неможливо, зате можна вивчити його будову і функціональність. Головною перевагою каркасного моделювання є те, що на зберігання тривимірних моделей, створених цим способом, не потрібно багато оперативної пам'яті комп'ютера. Найчастіше каркасну візуалізацію застосовують у спеціалізованих програмах для побудови передбачуваної траєкторії руху пристрою або інструменту.

Другий вид 3D-моделювання - це поверхневе моделювання. На відміну від каркасного, тут є не тільки сегменти, лінії та дуги, а й поверхні, що утворюють контур відображуваного об'єкта.

Останній, найточніший і най достовірніший тип 3D-моделювання, називається твердотільним моделювання. У результаті його використання можна отримати справжній зразок готового об'єкта, який передає всі дані про нього. Модель, створена завдяки цьому способу візуального відтворення, містить лінії, грані, текстуру та дані про об'єм і масу тіла. Хоча зображення і займають найбільший об'єм пам'яті комп'ютера порівняно з іншими, але він повністю описує готовий об'єкт. Твердотільне моделювання використовується всюди: при створенні техніки, промислових деталей, меблів, ювелірних виробів, кіно та комп'ютерних ігор. Через те, що 3D-моделі використовуються практично у всіх сферах, можна виділити чотири рівні складності об'ємних зображень:

- Перший - найпростіший - не містить інформації про структуру і дрібні деталі об'єкта, наприклад келихи і прості рамки.
- Другий, трохи складніший рівень, містить детальнішу інформацію про модель. До такого рівня можна віднести тумби, столи та інші нескладні предмети.
- До третього рівня можна віднести гарнітури меблів і техніку для дому через численні дрібні деталі та складну неоднорідну структуру.
- Ну а четвертий найчастіше використовується інженерами, прикладами тривимірних моделей цього рівня можуть бути моделі верстатів, автомобілів та іншої складної техніки.

Усі рівні моделі співвідносяться і з видами 3D-моделювання, так перші два - це каркасний, третій - поверхневий, а четвертий - твердотільний спосіб візуалізації. Тривимірна модель складається з безлічі точок, які з'єднуються між собою гранями й утворюють полігони. Вершина - це точка, яка має свої координати в тривимірній системі, тобто X, Y, Z. Свою назву вона отримала через те, що є крайньою точкою плоского багатокутника, або полігона. Грань, або ребро - відрізок, який з'єднує дві вершини, поняття, взяте з геометрії. У тривимірній графіці гранню називають обмежувач полігонів.

Основною складовою в тривимірній графіці вважається полігон - плоский багатокутник, безліч яких і утворює тривимірну фігуру. Абсолютно будь-яка фігура буде будуватися з численних простих фігур (до того ж більшість редакторів використовує трикутники і чотирикутники). Що більше буде простих фігур у складі складної, то більш гладкою здаватиметься поверхня 3D-моделі (так зване високополігональне моделювання).

Сукупність полігонів несе інформацію про розмір і форму 3D-моделі, а обрана текстура дає змогу передати достовірну інформацію про зовнішній вигляд об'єкта і являє собою зображення на поверхні фігури. Методи візуалізації в 3D-моделюванні:

- Scanline
- Raytrace (метод трасування променів)
- Raycasting (метод кидання променів)
- Radiosity

**Scanline** рендер завдяки своїй швидкості застосовується у відеоіграх та інтерактивних сценах. За наявності потужного відеоадаптера з його допомогою можна отримати чітке зображення з частотою понад 30 кадрів на секунду. Дія рендера заснована на реалізації принципу "ряд за рядом". Спочатку необхідні полігони розташовуються за найбільшою вертикальною координатою. Після цього кожен ряд зображення формується за допомогою перетину з найближчим до віртуальної камери полігоном. У процесі переходу між рядами відбувається видалення полігонів, які зникають з поля зору.

**Raytrace** - метою цього методу є отримання зображення з максимальною роздільною здатністю та детальною деталізацією. При цьому рендеринг займає багато часу і не підходить для створення анімованої графіки в реальному часі. Під час використання рейтрейс-методу від віртуальної камери для кожного пікселя на уявному екрані проводяться промені до найближчого тривимірного об'єкта. Колір точки визначається виходячи з того, з якими об'єктами стикається уявний промінь.

**Raycasting** - за цього способу відбувається те ж саме, що і в попередньому випадку, але тут розраховується тільки перша поверхня, на яку впаде промінь. Залежно від характеристик об'єкта й освітленості визначається колір пікселя зображення. Подальша обробка відбитих від об'єкта променів у такому разі не відбувається.

**Radiosity** - сенс такого рендерінгу полягає в тому, що поверхня об'єкта висвітлюється не тільки прямими джерелами світла, а й іншими поверхнями, що відбивають світло. При цьому точка огляду не має значення, що збільшує навантаження, але отримане зображення характеризується високою якістю і реалістичністю. Найкращого результату можна досягти, використовуючи для рендерінгу методи Radiosity і Raytrace одночасно.

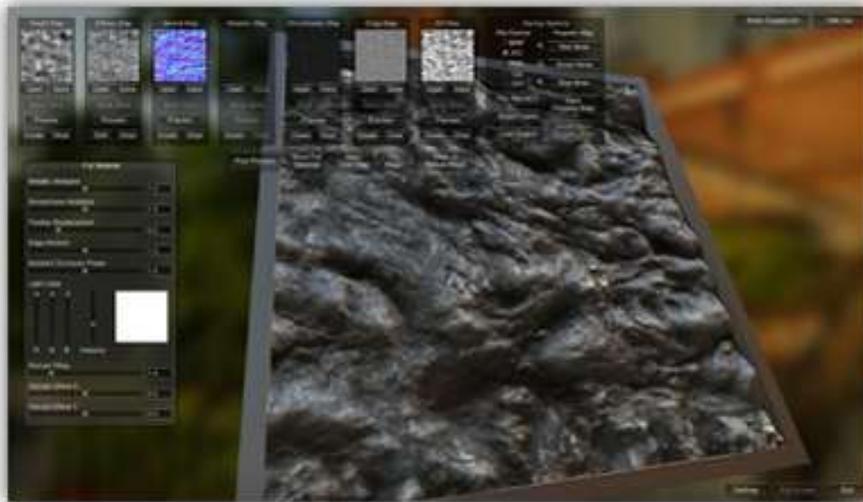
## 1.2. Огляд програмних платформ створення застосунків для роботи з інтерактивними тривимірними сценами

Для створення інтерактивних тривимірних сцен загалом можна використати будь-який тривимірний редактор, але слід враховувати що задля цього слід буде, для більшості з них – вносити власноруч необхідні корективи в програмний код програми для внесення потрібного функціоналу, або використовувати вже готові сторонні рішення. Подібний метод насамперед – не передбачений компанією розробником і ти самим, може мати безліч вразливостей, обмежень та недоліків. Окрім цього, створення власного застосунку на основі чужого продукту є суворим порушенням авторського права та може передбачати порушення ліцензії користування. Слід зауважити, що є одне виключення з цього правила, а саме – редактор тривимірної графіки Blender. Вона безкоштовна і її код є відкритим, створення безкоштовного застосунку на її основі не буде порушенням, але все ще залишаються перелічення проблеми з можливими обмеженнями та незручностями. У випадку мого проекту, з Blender було би необхідно вилучити надмірну кількість функцій і інструментів що були би зайвими, і як результат довелось би прибрати значну більшість її власного функціоналу, що вкрай нераціонально.

Раціональним і зручним рішенням було-би використанням за основу для створення редактора – ігрового рушія. Станом на сьогодні, найпопулярніші ігрові рушії у відкритому доступі або поширюються безкоштовно з певними обмеженнями, або мають безкоштовну версію з правилами користування. На даний момент можна виділити три наступні ігрові рушія що найбільш поширені та знаходяться у відкритому доступі:

- Godot Engine
- Unreal Engine
- Unity 3D

Реалізація застосунку на основі ігрового рушія не тільки найбільш зручне і гнучке рішення але і саме оптимізоване і вірне з точки зору авторського права і закону. Крім цього, створення замість гри – редактора та застосунку на основі ігрового рушія це не нова практика і є чудовий приклад. Materialize від Bounding Box Software. Materialize - це окремий інструмент для створення матеріалів для використання в іграх із зображень. Ви можете створити цілий матеріал з одного зображення або імпортувати наявні у вас текстури і згенерувати потрібні вам текстури.



*Рис. 1.1. Інтерфейс програми Materialize*

**Godot** - крос-платформний, безкоштовний і відкритий ігровий рушій, випущений під ліцензією MIT. Спочатку він був розроблений аргентинськими розробниками програмного забезпечення Хуаном Лінецьким та Аріелем Манзуром для кількох компаній у Латинській Америці до його публічного випуску. Середовище розробки працює на різних операційних системах, включаючи Linux, BSD, macOS та Microsoft Windows. Він призначений для створення як 2D, так і 3D ігор, орієнтованих на ПК, мобільні та веб-платформи. Воно також може використовуватися для створення неігрового програмного забезпечення, включаючи редактори. Godot має на меті запропонувати повністю інтегроване середовище для розробки ігор. Воно дозволяє розробникам створювати гру, не потребуючи інших інструментів, окрім тих, що використовуються для створення контенту

(візуальних ресурсів, музики тощо). Архітектура рушія побудована навколо концепції дерева "вузлів". Вузли організовані всередині "сцен", які є багаторазовими, екземплярами, успадкованими та вкладеними групами вузлів. Всі ігрові ресурси, включаючи сценарії та графічні ресурси, зберігаються як частина файлової системи комп'ютера (а не в базі даних). Таке рішення для зберігання призначене для полегшення співпраці між командами розробників ігор з використанням систем контролю версій програмного забезпечення.

Рушій підтримує розгортання на декількох платформах і дозволяє задавати параметри стиснення текстур та роздільної здатності для кожної платформи. На сайті надаються бінарні файли тільки для платформ редактора, а експорт проектів на інші платформи здійснюється всередині редактора Godot. Редактор Godot, який використовується для створення ігор Godot, підтримує наступні платформи:

- Desktopні платформи Linux, macOS та Windows, що розповсюджуються на сайті, в Steam та Itch. BSD також підтримується, але має бути скомпільована вручну.
- Веб-платформа HTML5, WebAssembly з веб-редактором.
- Телефони та планшети Android (доступні починаючи з версії Godot 3.5).

Рушій підтримує експорт проектів на багато інших платформ, включаючи всі платформи редакторів. Наразі підтримувані платформи станом на Godot 3.5.1 такі:

- Мобільні платформи Android, iOS
- Desktopні платформи Linux, macOS, Microsoft Windows (Universal Windows Platform), BSD (має бути скомпільована вручну)
- Веб-платформи HTML5, WebAssembly.

- Платформи віртуальної/розширеної реальності HTC Vive, Valve Index, Oculus Rift, Oculus Go, Oculus Quest, всі гарнітури Microsoft MR, Apple ARKit та багато інших.

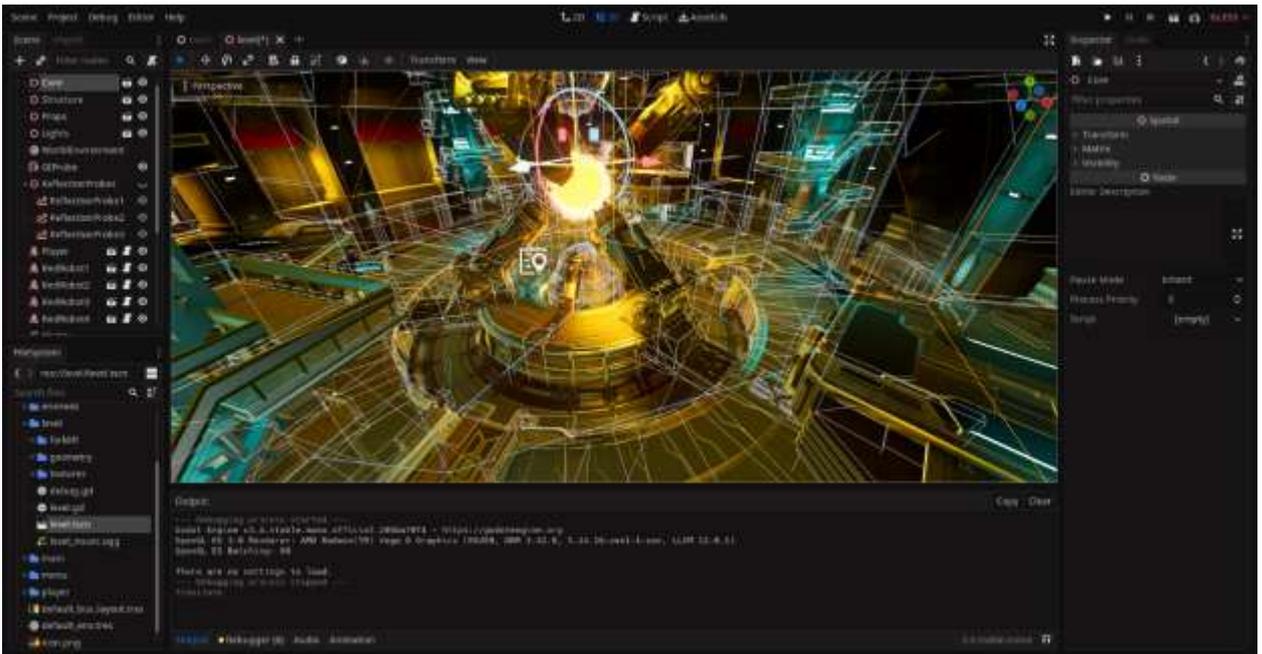
Незважаючи на те, що рушій Godot може працювати на консолях, компанія Godot не підтримує його офіційно, оскільки є проектом з відкритим вихідним кодом, а не ліцензованою компанією, і вони не можуть публікувати код для конкретної платформи під ліцензією з відкритим вихідним кодом. Однак, портування ігор на консолі все ще можливе завдяки послугам, що надаються сторонніми компаніями. Що стосується архітектур процесорів, Godot офіційно підтримує x86 на всіх настільних платформах (як 32-розрядних, так і 64-розрядних, де це можливо) і має офіційну підтримку ARM на macOS, мобільних платформах і автономних платформах Oculus (як 32-розрядних, так і 64-розрядних, де це можливо). Веб-платформа використовує 32-розрядний WebAssembly. Підтримка ARM, RISC-V та PowerPC Linux є неофіційною та експериментальною.

Godot підтримує різноманітні мови програмування для створення ігор, включаючи інтегровану мову GDScript, C++ та C#. Крім того, рушій включає GDNative, засіб для створення зв'язків з іншими мовами. Офіційно підтримувані мови GDNative включають C та C++. Підтримувані спільнотою мови включають Rust, Nim, JavaScript, Haskell, Clojure, Swift та D. Також підтримується візуальне кодування за допомогою вбудованої мови VisualScript, розробленої як візуальний еквівалент GDScript. Візуальне програмування буде вилучено з ядра рушія в Godot 4.0. Редактор Godot включає текстовий редактор з автоматичним відступом, підсвічуванням синтаксису та завершенням коду. Він також має налагоджувач з можливістю встановлення точок зупинки та крокування програми.

Godot має власну вбудовану мову сценаріїв, GDScript, мову програмування високого рівня з динамічною типізацією, яка синтаксично схожа на Python. На відміну від Python, GDScript оптимізована для сценічної архітектури Godot і може визначати сувору типізацію змінних. Розробники

Godot заявили, що було протестовано багато альтернативних сторонніх скриптових мов, таких як Lua, Python та Squirrel, перш ніж вирішили, що використання власної мови дозволить досягти кращої оптимізації та інтеграції з редакторами. У версії 4.0 в GDScript була реалізована нова функція під назвою Typed array. Це дозволяє користувачам легко змінювати звичайний масив на типізований і навпаки, не змінюючи багато коду.

Графічний рушій Godot використовує OpenGL ES 3.0 для всіх підтримуваних платформ; в іншому випадку використовується OpenGL ES 2.0. Розробляється майбутня підтримка Vulkan, що також включає можливість підтримки Metal за допомогою MoltenVK. Рушій підтримує нормальне відображення, дзеркальність, динамічні тіні за допомогою карт тіней, запечене та динамічне глобальне освітлення, а також повноекранні ефекти постобробки, такі як bloom, DOF, HDR та гамма-корекція. Також включена спрощена мова шейдерів, подібна до GLSL. Шейдери можна використовувати для матеріалів та постобробки. Крім того, вони можуть бути створені шляхом маніпулювання вузлами у візуальному редакторі. Godot також включає окремий 2D-графічний рушій, який може працювати незалежно від 3D-рушія. 2D-рушій підтримує такі функції, як світло, тіні, шейдери, набори плиток, паралаксу прокрутку, полігони, анімацію, фізику і частинки. Також є можливість змішувати 2D та 3D за допомогою "вузла видошукача". Godot вийшов з Консерваторії свободи програмного забезпечення 1 листопада 2022 року. [1]



*Рис. 1.2. Інтерфейс рушія Godot*

**Unity** - кросплатформний ігровий рушій, розроблений компанією Unity Technologies, вперше анонсований та випущений у червні 2005 року на конференції Apple Worldwide Developers Conference як ігровий рушій для Mac OS X. З того часу рушій поступово розширювався для підтримки різноманітних настільних, мобільних, консольних платформ та платформ віртуальної реальності. Він особливо популярний для розробки мобільних ігор для iOS та Android, вважається простим у використанні для розробників-початківців, а також популярний для розробки інді-ігор.

Рушій може використовуватися для створення тривимірних (3D) і двовимірних (2D) ігор, а також інтерактивних симуляторів та інших додатків. Рушій був прийнятий на озброєння галузями, що не відносяться до відеоігор, такими як кіно, автомобілебудування, архітектура, машинобудування, будівництво та Збройні сили США. Ігровий рушій Unity був запущений у 2005 році з метою "демократизувати" розробку ігор, зробивши її доступною для більшої кількості розробників. Наступного року Unity посів друге місце в категорії "Найкраще використання графіки Mac OS X" на конкурсі Apple Design Awards від Apple Inc. Спочатку Unity був випущений для Mac OS X, пізніше була додана підтримка Microsoft Windows і веб-браузерів.

Unity дає користувачам можливість створювати ігри та досвід як у 2D, так і в 3D, і рушій пропонує основний API сценаріїв на C# з використанням Mono, як для редактора Unity у вигляді плагінів, так і для самих ігор, а також функціональність перетягування. До того, як C# стала основною мовою програмування, що використовується для рушія, він раніше підтримував Boo, який був видалений з виходом Unity 5, і засновану на Boo реалізацію JavaScript під назвою UnityScript, яка була застаріла в серпні 2017 року, після виходу Unity 2017.1, на користь C#. У 2D-іграх Unity дозволяє імпортувати спрайти та вдосконалений 2D-візуалізатор світу. Для 3D-ігор Unity дозволяє вказувати стиснення текстур, міпмапи та налаштування роздільної здатності для кожної платформи, яку підтримує ігровий рушій, і забезпечує підтримку відображення нерівностей, відображення відображень, відображення паралаксу, оклюзії екранного простору навколишнього середовища (SSAO), динамічних тіней за допомогою карт тіней, рендерингу в текстуру та повноекранних ефектів постобробки.

Unity є кросплатформним рушієм. Редактор Unity підтримується на Windows, macOS та платформі Linux, а сам рушій наразі підтримує створення ігор для більш ніж 19 різних платформ, включаючи мобільні, настільні, консолі та віртуальну реальність. Unity 2020 LTS офіційно підтримує наступні платформи:

- Мобільні платформи iOS, Android (Android TV), tvOS;
- Десктопні платформи Windows (Universal Windows Platform), Mac, Linux;
- Веб-платформа WebGL;
- Консольні платформи PlayStation (PS4, PS5), Xbox (Xbox One, Xbox Series X/S), Nintendo Switch, Stadia;
- Платформи віртуальної/доповненої реальності Oculus, PlayStation VR, Google ARCore, Apple ARKit, Windows Mixed Reality (HoloLens), Magic Leap, та через Unity XR SDK Steam VR, Google Cardboard.

Протягом перших десяти років існування продукту платні версії Unity продавалися повністю; у 2016 році корпорація перейшла на модель підписки. Unity має безкоштовні та платні варіанти ліцензування. Безкоштовна ліцензія призначена для особистого користування або невеликих компаній, що приносять менше 100 000 доларів на рік, пізніше піднята до 200 000 доларів, а підписка базується на доходах, отриманих від ігор з використанням Unity. Платний варіант, Unity Pro, був необхідний для розробників, які мали понад 200 000 доларів річного доходу, але це також могло бути надано розробникам консолей за допомогою ліцензії на привілейовану платформу від виробника консолей. Ключі Unity Pro були б частиною іншого SDK від виробника консолі, за який розробник платив.

2010-х роках Unity Technologies використала свій ігровий рушій для переходу в інші галузі, що використовують 3D-платформу в реальному часі, включаючи кінематограф та автомобілебудування. Unity вперше експериментувала в кінематографі з Adam, короткометражним фільмом про робота, який тікає з в'язниці. Пізніше Unity співпрацювала з режисером Нілом Бломкампом, чия студія Oats Studios використовувала інструменти рушія, включаючи рендеринг у реальному часі та Cinemachine, для створення двох комп'ютерних короткометражних фільмів, Адам: Дзеркало та Адам: Пророк. На конференції Unite Europe 2017 року в Амстердамі Unity зосередилася на кінематографі за допомогою Unity 2017. 1's new Cinemachine tool. У 2018 році Disney Television Animation випустила три короткометражки під назвою Baymax Dreams, які були створені за допомогою рушія Unity. Рушій Unity також використовувався Disney для створення фонів для фільму "Король Лев" 2019 року. Автовиробники використовують технологію Unity для створення повномасштабних моделей нових автомобілів у віртуальній реальності, побудови віртуальних складальних ліній та навчання робітників. Рушій Unity використовується DeepMind, компанією Alphabet Inc. для навчання штучного інтелекту. Інші сфери використання Unity Technologies включають архітектуру, інженерію та будівництво. [2]



*Рис. 1.3. Інтерфейс рушія Unity 3D*

**Unreal Engine** (UE) - ігровий рушій для комп'ютерної 3D-графіки, розроблений компанією Epic Games, вперше продемонстрований у грі-шутері від першої особи Unreal 1998 року. Спочатку розроблений для комп'ютерних шутерів від першої особи, він з тих пір використовується в різних жанрах ігор і був прийнятий іншими галузями, особливо в кіно- і телеіндустрії. Написаний на мові C++, Unreal Engine відрізняється високим ступенем портативності, підтримуючи широкий спектр настільних, мобільних, консольних платформ та платформ віртуальної реальності. Останнє покоління, Unreal Engine 5, було запущено у квітні 2022 року. Його вихідний код доступний на GitHub після реєстрації облікового запису, а комерційне використання надається на основі моделі роялті. Epic відмовляється від своєї маржі роялті за ігри до тих пір, поки розробники не отримають дохід у розмірі 1 млн доларів США, а також скасовує плату, якщо розробники публікують ігри в Epic Games Store. Epic включила функції від придбаних компаній, таких як Quixel, у свій рушій, що, як вважається, сприяло зростанню доходів Fortnite.

У серпні 2005 року Марк Рейн, віце-президент Epic Games, показав, що Unreal Engine знаходиться в розробці вже два роки. "Люди не усвідомлюють цього, але ми вже два роки розробляємо Unreal Engine. Звичайно, у нас ще немає повної команди, це лише один хлопець, і ви, напевно, можете

здогадатися, хто це", - сказав він в інтерв'ю C&VG. Виступаючи в інтерв'ю на початку 2008 року, Свіні заявив, що він був фактично єдиною людиною, яка працювала над рушієм, хоча він підтвердив, що його відділ досліджень і розробок почне розширюватися пізніше того ж року, розробляючи рушій паралельно з розробкою Unreal Engine 3. "В деякому роді ми схожі на компанію, що займається виробництвом комп'ютерного обладнання, завдяки нашому поколінню в області розвитку технологій. У нас буде команда, яка розроблятиме Unreal Engine 3 протягом наступних років, і команда, яка нарощуватиме темпи роботи над Unreal Engine. А потім, коли почнеться перехід на наступне покоління, ми переведемо всіх на нього. Ми фактично ведемо паралельну розробку для декількох поколінь одночасно", - сказав він. У лютому 2012 року Рейн заявив, що "люди будуть шоковані пізніше цього року, коли побачать Unreal Engine"; Еріс представила UE4 обмеженому колу відвідувачів на Конференції розробників ігор 2012 року, а відео з демонстрацією рушія технічним художником Аланом Віллардом було випущено для громадськості 7 червня 2012 року через GameTrailers TV. Однією з головних особливостей, запланованих для UE4, було глобальне освітлення в реальному часі з використанням воксельної трасування конусів, що усуває заздалегідь розраховане освітлення. Однак ця функція, що отримала назву Sparse Voxel Octree Global Illumination (SVOGI) і була продемонстрована в демонстраційній версії Elemental, була замінена на схожий, але менш обчислювально дорогий алгоритм через проблеми з продуктивністю. UE4 також включає нову систему візуальних сценаріїв "Blueprints" (наступник "Kismet" з UE3), яка дозволяє швидко розробляти ігрову логіку без використання коду, що призводить до меншого розриву між технічними художниками, дизайнерами та програмістами.

19 березня 2014 року на Конференції розробників ігор (GDC) компанія Epic Games випустила рушій Unreal Engine за новою моделлю ліцензування. За щомісячну підписку у розмірі 19 доларів США розробники отримували доступ до повної версії рушія, включаючи вихідний код на мові C++, який

можна було завантажити через GitHub. З будь-якого випущеного продукту стягувалося роялті у розмірі 5% від валового доходу. Першою грою, випущеною з використанням Unreal Engine, стала Daylight, розроблена з раннім доступом до рушія і випущена 29 квітня 2014 року.

4 вересня 2014 року Еріс випустила Unreal Engine для шкіл та університетів безкоштовно, включаючи персональні копії для студентів, які навчаються на акредитованих програмах з розробки відеоігор, комп'ютерних наук, мистецтва, архітектури, симуляції та візуалізації. Еріс відкрила Unreal Engine Marketplace для придбання ігрових активів. 19 лютого 2015 року Еріс запустила Unreal Dev Grants - фонд розвитку з бюджетом у 5 мільйонів доларів, метою якого є надання грантів творчим проектам з використанням Unreal Engine.

У березні 2015 року Еріс випустила Unreal Engine, разом з усіма майбутніми оновленнями, безкоштовно для всіх користувачів. В обмін на це Еріс встановила вибірковий графік роялті, вимагаючи 5% від доходу для продуктів, які приносять більше 3000 доларів на квартал. Суїні заявив, що коли вони перейшли на модель підписки в 2014 році, використання Unreal зросло в 10 разів і за рахунок багатьох менших розробників, і вважав, що вони залучать ще більше користувачів за допомогою цієї нової схеми ціноутворення.

Unreal Engine офіційно підтримує наступні платформи станом на 5.1:  
Windows, macOS, Linux

- iOS, Android
  - Nintendo Switch, PlayStation 4, Xbox One, PlayStation 5, Xbox Series X/S, та Stadia
  - Magic Leap, HTC Vive, Oculus, PlayStation VR, OSVR, Samsung Gear VR, і HoloLens 2
  - Раніше він офіційно підтримував Google Daydream та HTML5
- UnrealScript (часто скорочено UScript) - рідна скриптова мова Unreal.

Engine, що використовувалася для написання ігрового коду та подій ігрового процесу до виходу Unreal Engine. Мова була розроблена для простого, високорівневого ігрового програмування. Інтерпретатор UnrealScript був запрограмований Суїні, який також створив більш ранню мову ігрових сценаріїв ZZT-оор. Провідний програміст Deus Ex Кріс Норден описав його як "супер гнучкий", але відзначив його низьку швидкість виконання. Подібно до Java, UnrealScript був об'єктно-орієнтованим без множинного успадкування (всі класи успадковуються від загального класу Object), а класи визначалися в окремих файлах, названих для класу, який вони визначають. На відміну від Java, UnrealScript не мала обгортки об'єктів для примітивних типів. Інтерфейси підтримувалися лише в Unreal Engine 3-го покоління та деяких іграх на Unreal Engine 2. UnrealScript підтримував перевантаження операторів, але не перевантаження методів, за винятком необов'язкових параметрів. На конференції розробників ігор 2012 року Epic оголосила, що UnrealScript видаляється з Unreal Engine на користь C++. Візуальне написання сценаріїв буде підтримуватися системою візуальних сценаріїв Blueprints, заміною попередньої системи візуальних сценаріїв Kismet.

Рушій Unreal Engine спочатку був розроблений для використання в якості базової технології для відеоігор. Рушій використовується в ряді гучних ігрових проєктів з високими графічними можливостями, включаючи PlayerUnknown's Battlegrounds, Final Fantasy VII Remake, Valorant і Yoshi's Crafted World, а також в іграх, розроблених компанією Epic, в тому числі Gears of War і Fortnite.

Рушій Unreal Engine знайшов застосування у кінематографі для створення віртуальних декорацій, які можуть відслідковувати рух камери навколо акторів та об'єктів і виводитися в реальному часі на великі світлодіодні екрани та системи атмосферного освітлення. Це дозволяє в реальному часі компонувати кадри, негайно редагувати віртуальні декорації за потреби і знімати декілька сцен протягом короткого періоду, просто

змінюючи віртуальний світ позаду акторів. Загальний вигляд був визнаний більш природним, ніж типові хромакейні ефекти.

Серед постановок, де були використані ці технології, були телевізійні серіали "Мандалорієць" та "Західний світ". Джон Фавро та підрозділ Lucasfilm Industrial Light & Magic працювали з Epic над розробкою технології StageCraft для "Мандалорця" на основі аналогічного підходу, який Фавро використовував у фільмі "Король Лев". Потім Фавро поділився цим технологічним підходом з Джонатаном Ноланом та Лізою Джой, продюсерами "Західного світу". Шоу вже розглядало використання віртуальних декорацій раніше і мало певні напрацьовані технології, але інтегрувало використання Unreal Engine, як і в StageCraft, для третього сезону. Іспанська компанія Orca Studios співпрацює з Epic над створенням декількох студій для віртуальних зйомок, подібних до підходу StageCraft з використанням Unreal Engine, що забезпечує віртуальні декорації, особливо під час пандемії COVID-19, яка обмежила подорожі. У січні 2021 року Deadline Hollywood оголосив, що Epic використовує частину своїх Epic MegaGrants, щоб вперше підтримати анімаційний художній фільм "Гільгамеш", який буде повністю вироблений в Unreal Engine анімаційними студіями Hook Up, DuermeVela та FilmSharks. В рамках розширення своїх MegaGrants, Epic також профінансував 45 додаткових проектів приблизно з 2020 року для створення фільмів та короткометражних фільмів на Unreal Engine. Unreal Engine також використовувався в нетворчих сферах завдяки своїй доступності та набору функцій. Він використовувався як основа для інструменту віртуальної реальності для дослідження молекул фармацевтичних препаратів у співпраці з іншими дослідниками, як віртуальне середовище для дослідження та проектування нових будівель та автомобілів, а також використовувався для кабельних новинних мереж для підтримки графіки в реальному часі. [3]



*Рис. 1.4. Інтерфейс рушія Unreal Engine*

## РОЗДІЛ II. АНАЛІЗ ЗАСОБІВ ДЛЯ СТВОРЕННЯ РЕДАКТОРА

### 2.1. Процес створення тривимірних моделей

3D-моделювання або тривимірна графіка - створення об'ємної моделі того чи іншого об'єкта. Візуалізація 3D-моделей дає змогу продемонструвати виріб цілком, а також його окремі частини. За допомогою такої моделі можна показати як видимі елементи, так і приховані частини виробу. Процес створення 3D-моделі складається з таких кроків:

1. Розробка форми та геометрії предмета. На цьому етапі створюється форма виробу, при цьому його фізичні параметри не враховуються. Для цього застосовуються такі методи створення 3D-моделей, як видавлювання, полігональне моделювання, обертання.
2. Створення окремих елементів виробу та їх подальше складання. Інженери розробляють деталі, виходячи з умов їхньої експлуатації та призначення. Далі йде процес складання, де всі вузли виробу стають на свої робочі місця.
3. Текстурування. Накладення текстур дає змогу домогтися реалістичного зображення виробу.
4. Вибір точки спостереження та налаштування освітлення. Це досить складний крок, від якого залежить реалістичність тривимірної моделі. Потрібно виставити контрастність, яскравість, глибину тіней та інші характеристики світла.
5. Тривимірна візуалізація та рендеринг. Фахівці підвищують деталізацію і додають ефекти: туман, світлові відблиски, сяйво тощо. Також на цьому кроці уточнюються деталі тривимірної візуалізації та налаштування моделювання.

Заключний етап - постобробка. На цьому кроці додають візуальні ефекти, які привертають увагу споживача. Перш ніж створити прототип за

3D-моделлю, потрібно здійснити її адаптацію. Модель підлаштовують під 3D-обладнання, а також перевіряють її геометрію і технічні параметри. У проєкт вносяться коригування з урахуванням можливого просідання матеріалу і вимог обладнання. Дотримання технології моделювання дає змогу домогтися максимально високої відповідності та якості прототипу.

Створення тривимірних моделей саме для медіа, відеогри, кіно та в цілому художньої сфери та сфера розваг. Деякі моменти з процесу можуть мінятися місцями, оскільки для створення персонажа і, наприклад, танка, пайплайн (тобто процес розроблення) відрізнятиметься:

Концепт-арт. Концепт-арт - це ідея, втілена художником, за коротким її описом. Призначенням концепт-арту є створення начерку об'єкта, який буде запущений у виробництво. За короткий термін необхідно підібрати найкращу концепцію, інакше можна припуститися помилок на пізніх етапах створення моделі.[5]

Скульптинг Для початку художнику належить зробити скульпт моделі. На цьому етапі потрібно зануритись у творчість і моделювати засобами скульптингу, не замислюючись про полігони. Після всього процесу скульптування модель вже має мати максимально закінчений вигляд, оскільки саме її треба використовувати для подальших етапів виробництва. Таку модель не можна буде додати в гру, і зробити наступні етапи буде проблематично, оскільки на ній занадто багато полігонів.[5]

Ретопологія. На цьому етапі доведеться зменшити кількість полігонів. Ретопологія - це перебудова полігональної сітки, створення нової геометрії поверх старої для подальшого її використання. Низькополігональна необхідна насамперед для оптимізації, а також для створення чистої сітки під час подальшої анімації та створення UV карт. Після етапу ретопології, модель може здатися надто простою, плоскою, але це виправляється після.[5]

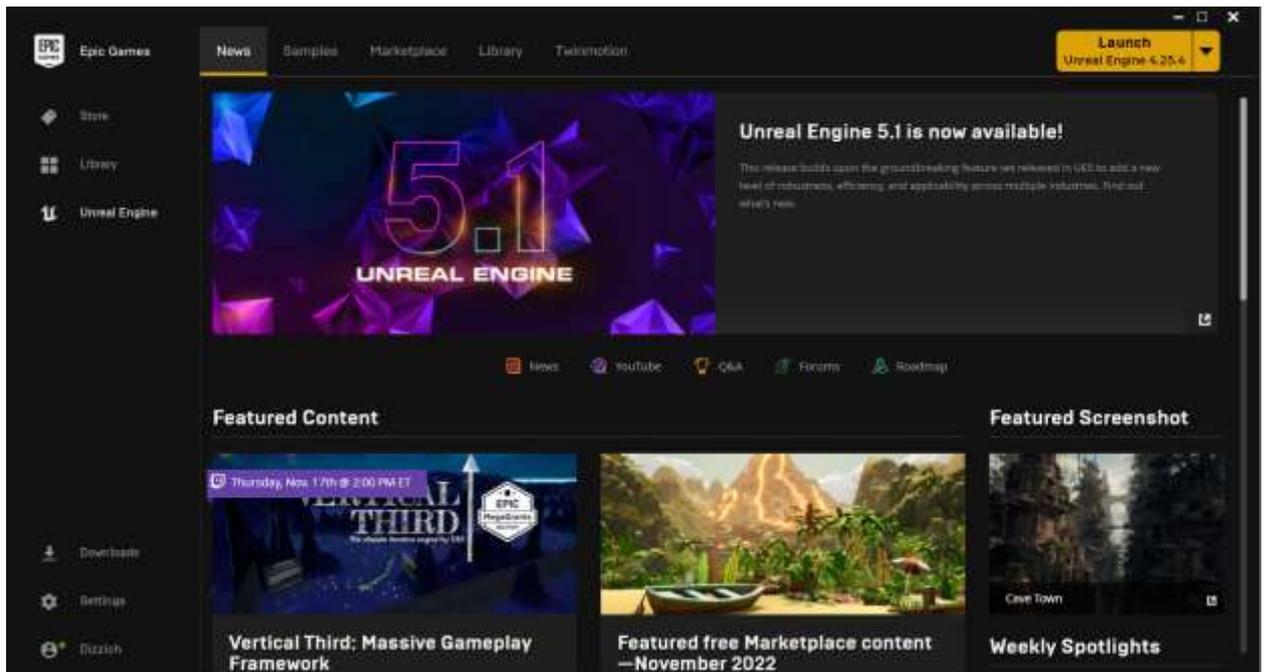
UV-розгортка. Цей етап потрібен для того, щоб текстури поводитися коректно. Процес розгортки нагадує ліпку фігур з паперу. UV карта створюється шляхом розрізання граней на моделі.[5]

Запікання мап. Запікання необхідне для перенесення деталізації з високополігональної моделі (high poly) на низькополігональну (low poly). Насамперед це потрібно для того, щоб модель мала не плоский вигляд і коректно відображала освітлення, яке перебуває в сцені. До прикладу інформація про основні використовувані мапи. Normal Map - якраз таки ця мапа дає змогу додати деталізацію, не витрачаючи на це зайві полігони. Вектори, які використовуються для визначення того, як світло відбивається від поверхні. Їх можна використовувати для контролю над переходом між гранями, але також їхній напрямок можна змінювати, щоб lowpoly-модель відбивала світло так само, як складніша модель. В такому випадку, низькополігональна модель починає відбивати поверхню так само як і високополігональна. За рахунок цього здається, що модель стала більш деталізованою. Ambient Occlusion – ця мапа додає тіні на моделі, де це необхідно. Тим самим модель стає більш реалістичною. Curvature - це карта прораховує нерівності на поверхні моделі, так само виділяє всі грані, після чого ці нерівності можна буде використовувати під час текстурування.[5]

Текстурування. Практично останній етап в процесі розробки. Етап текстурування досить важливий, адже саме текстури сильно впливають на сприйняття моделі. Текстура - це масив кольірних точок, що утворюють зображення. Це не тільки розфарбування об'єкта. Фактично, термін текстура означає шорсткість або гладкість поверхні об'єкта. Це ті властивості поверхні, які можна відчувати на дотик. PBR матеріали - це матеріали, які коректно відображають усі текстури, тобто шорсткості, відображення, затінення.[5]

## 2.2. Інтерфейс та архітектура ігрового рушія Unreal Engine

Unreal Engine - це набір інструментів для розробки ігор, що має широкі можливості: від створення двомірних ігор на мобільні до AAA-проектів для консолей. Цей рушій використовувався під час розроблення таких ігор, як ARK: Survival Evolved, Tekken 7 та Kingdom Hearts III. Розробка в Unreal Engine дуже проста для початківців. За допомогою системи візуального створення скриптів Blueprints Visual Scripting можна створювати готові ігри, не написавши жодного рядка коду! У поєднанні зі зручним інтерфейсом це дає змогу швидко виготовляти робочі прототипи. Для встановлення Unreal Engine використовується Epic Games Launcher. Перейдіть на сайт Unreal Engine і натисніть на кнопку Get Unreal у правому верхньому куті. Перед завантаженням програми запуску необхідно буде створити обліковий запис. Після його створення скачайте програму запуску, що відповідає вашій операційній системі, встановіть лаунчер а в ньому вже можна встановити сам рушій та компоненти до нього. [6]



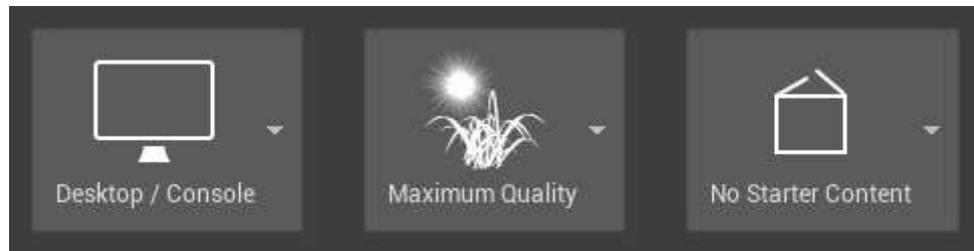
*Рис.2.1. Інтерфейс застосунку Epic Store*

Слід натиснути на одну з кнопок Launch, щоб відкрити браузер проектів (Project Browser). Після його відкриття натисніть на вкладку New Project. Потрібно вибрати Blueprint.



*Рис.2.2. Вікно вибору шаблону в Unreal Engine*

Далі можна вибрати один із шаблонів. Однак, оскільки проект створюється з нуля, то потрібен шаблон Blank. Нижче перераховані додаткові параметри.[6]

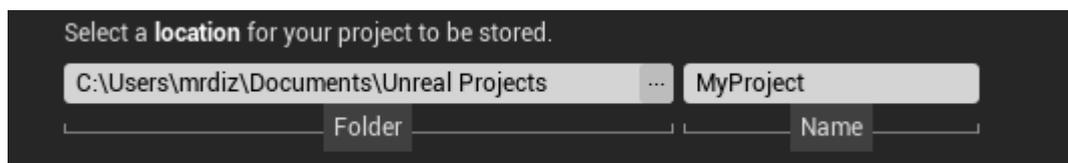


*Рис.2.3. Вибір стартових параметрів проекту в вікні створення проекту в Unreal Engine*

Ось, за що відповідає кожна опція:

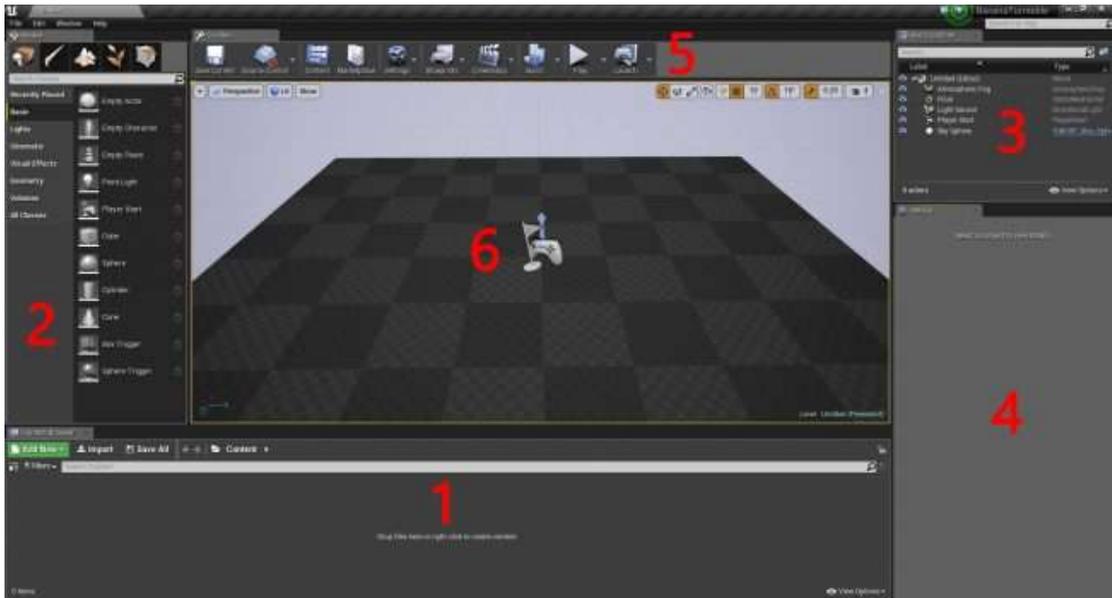
- Target Hardware: при виборі Mobile/Tablet будуть відключені деякі ефекти постобробки. Також можна буде використовувати мишу для сенсорного введення. Оптимально використовувати Desktop/Console, це стандарт.
- Graphical Target: у разі вибору Scalable 3D or 2D буде вимкнено деякі ефекти постоброблення.
- Starter Content: можна ввімкнути цю опцію, щоб додати базовий контент (Starter Content).

Крім того, є розділ для вибору місця розташування папки проекту та імені проекту.



*Рис.2.4. Панель вибору теки для зберігання проекту та файлів проекту в Unreal Engine*

Після створення проекту відкриється редактор. Він розділений на кілька панелей:



*Рис. 2.5. Головне вікно редактору Unreal Engine*

1. Content Browser: у цій панелі відображаються всі файли проєкту. Її можна використовувати для створення папок і впорядкування файлів. Тут також можна виконувати пошук за файлами за допомогою пошукового рядка або фільтрів.
2. Modes: у цій панелі можна перемикатися між інструментами, наприклад Landscape Tool і Foliage Tool. Інструментом за замовчуванням є Place Tool. Він дає змогу розташовувати на рівні різні типи об'єктів, як-от джерела освітлення і камери.
3. World Outliner: відображає всі об'єкти на поточному рівні. Можна впорядкувати список, розподіливши пов'язані об'єкти по папках, а також шукати і фільтрувати їх за типами.
4. Details: тут відображаються всі властивості обраного об'єкта. Ця панель використовується для зміни параметрів об'єкта. Внесені зміни вплинуть тільки на обраний екземпляр об'єкта. Наприклад, якщо в сцені є дві сфери, то під час зміни розміру однієї зміни торкнуться тільки її.
5. Toolbar: містить безліч різних функцій. Найчастіше ми будемо користуватися Play.

6. Viewport: це огляд рівня. Оглядатися на всі боки можна, утримуючи праву клавiшу миші та переміщаючи її. Для переміщення рівнем потрібно затиснути праву клавiшу миші та затискати клавiші WASD.

## РОЗДІЛ ІІІ. РОЗРОБКА РЕДАКТОРА

### 3.1. Проектування редактора та тривимірних моделей для нього

Перед створенням власне, самого застосунку редактору, потрібно мати більш чітке і конкретне уявлення того, як має виглядати і що саме має робити застосунок. По задуму, редактор має бути інтерактивним тривимірним застосунком з сучасним рендер-рушієм для тривимірної візуалізації у реальному часу, щоб відображувати результат генерування сцен з застосуванням готових тривимірних моделей. Для розробки необхідно мати уявлення про майбутній дизайн інтерфейсу програми, опис її функціоналу, поняття про те, які саме тривимірні сцени повинні бути згенеровані і які для цього, власне, необхідно створити тривимірні моделі.

Обов'язковий функціонал редактору – генерування інтерактивної тривимірної сцени з використанням готових моделей, а якщо конкретніше – наявність тривимірної сцени з візуалізацією у реальному часі та з генеруванням сцени з набору створених моделей. Тобто, це тривимірну сцену у якій користувач має можливість впливати на неї та генерувати з набору моделей - сцену. За моїм уявленням, базовий функціонал має передбачати можливість користувачу самому обирати моделі з яких буде генеруватись сцена, фінальний тривимірний об'єкт.

За візуальним та художнім спрямуванням я обрав тему середньовічної холодної зброї, але без чіткої прив'язки за напрямом, без історичної відповідності і тощо. Зробити генерацію саме холодної зброї конкретного історичного періоду і з чітким слідуванням історичної відповідності можна при такій необхідності і постановці такої задачі, в рамках першої, базової версії редактора – така необхідність відсутня, просто генерації зброї відповідної до тематики середньовічних мечів буде більш ніж достатнім. Для подальшої реалізації необхідні приклади того що я хочу створити, тому в мережі у вільному просторі було обрано невеликий набір зображень мечів,

відповідно до яких, схожі складові меча і будуть змодельовані в якості тривимірних моделей.



*Рис. 3.1. Приклад гарди та помелу для середньовічного меча*



*Рис. 3.2. Ілюстрація з мечами в грі The Witcher 3*



*Рис. 3.3. Приклади середньовічних мечів*

Задум передбачає що з набору складових комплектуючих середньовічного меча, користувач інтерактивно, обираючи на свій погляд ці частини – генеруватиме сам сцену з тривимірним, згенерованим мечом.

З огляду на сучасні тенденції розробки застосунків – дизайн інтерфейсу користувача повинен бути мінімалістичним, легко зрозумілим та сучасно виглядати. На розгляді було декілька ідей майбутнього дизайну інтерфейсу з яких була обрана одна. Перший варіант інтерфейсу – список. Користувач обирає модель на кожну позицію у форматі списку.

Другий варіант інтерфейсу – комірки, майже як перший, але на екрані розміщені комірки в котрих користувач обирає частину відповідно кожної складової меча.

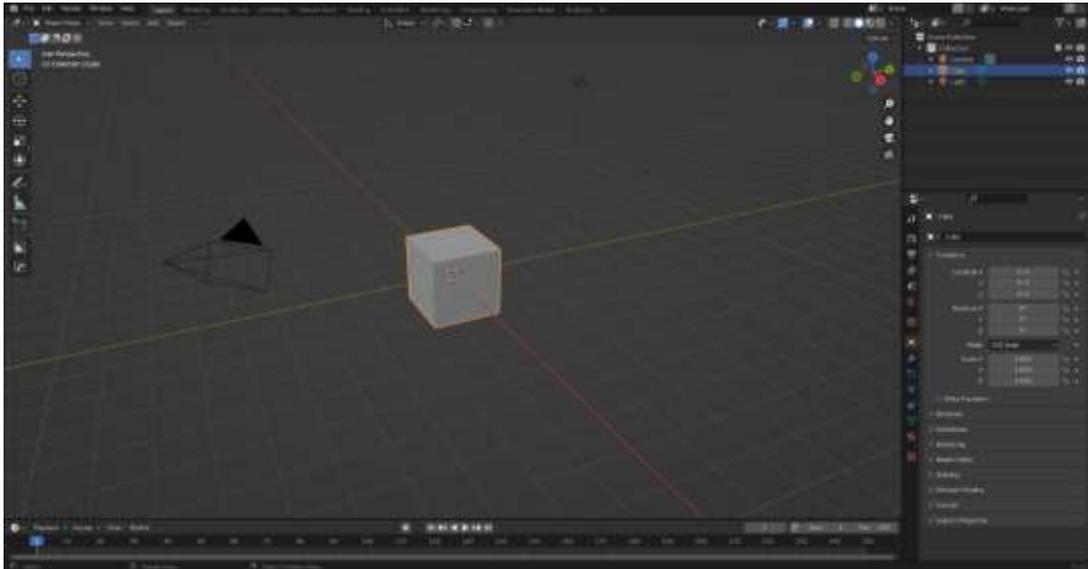
Третій варіант – інтерактивні динамічні комірки, відповідні кожній частині. Обраний і найкращий варіант реалізації, передбачає сучасний, гарний і очевидно працюючий дизайн інтерфейсу. Поява нових комірок буде залежати від можливості приєднання відповідної частини меча, відповідність комірки до частини меча буде наглядно проілюстрована через дизайн.

Дизайн інших елементів інтерфейсу користувача буде притримуватись мінімалістичного і спрощеного зовнішнього вигляду.

### 3.2. Створення тривимірних моделей

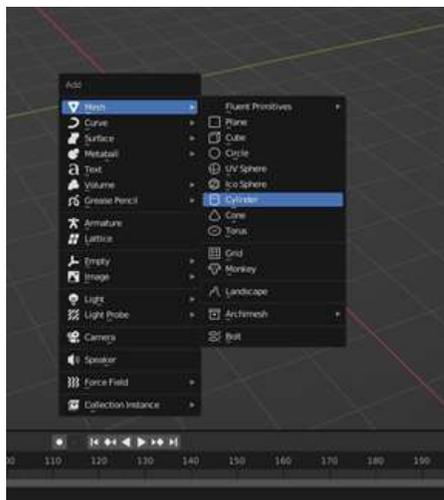
Для всього процесу створення та редагування тривимірних моделей в проекті використовується безкоштовний редактор Blender версії 3.2.0. Редактор обрано базуючись на результатах дослідження поширених тривимірних редакторів і їх порівнянні. Детально про дослідження 3D-редакторів у Додатках. (додаток А). Створюватись моделі будуть за обраними прикладами та замальовками. Технічним форматом моделей обрано “mid-poly”, це значно спростить процес інтерактивної взаємодії рушія з моделями, так як для створення матеріалів візуалізації, не будуть створюватись та використовуватись технічні текстурні матеріали, типу “normal-map”, “ambient occlusion map”, “object map”. Формат “high-poly” не буде використано, так як він неоптимізований і в ньому немає нагальної потреби, середньої кількості полігонів чи трикутників більш ніж достатньо для привабливої візуалізації. Формат полігональної сітки моделей що буде створена, передбачає формат полігонів виключно “quads”, так як це самий оптимізований та привабливий візуально формат сітки для тривимірного рендерингу. В такому форматі, рушій автоматично підрозділяє квадратні полігони на трикутні, а також, при використанні будь якого згладжування – квадратна сітка також краще підрозділяє полігони та згладжує їх.

Далі на прикладі одного набору моделей для генерації, буде розглянуто крок за кроком процес створення тривимірних моделей для проекту, з врахуванням технічних потреб. Першим кроком є, насамперед відчинення програми Blender та створення нової, пустої сцени.



*Рис. 3.4. Інтерфейс нової сцени в Blender*

Далі комбінацією клавіш “A + 8” виділяються всі об’єкти у сцені та натискаючи “Del” – видаляються. Далі, в порожній сцені, комбінацією клавіш “Shift + A” викликається контекстне меню створення нових об’єктів. Зі списку потрібно обрати примітив, найближчий за формою до створюваного об’єкта. Перша модель буде рукоятка меча, для її створення найзручніше буде почати з примітива циліндричної форми.



*Рис. 3.5. Створення нової параметричної фігури в Blender*

Появою примітива у сцені його створення не обмежено. У панелі в нижньому лівому куті інтерфейсу, до початку полігонального редагування об’єкта можна змінити деякі з його параметрів. Для моделювання з використанням автоматичного підрозділу та згладжуванням, краще буде

почати з як можна меншої кількості полігонів, наприклад змінивши значення з 32 полігонів на стінках циліндра до 16.



*Рис. 3.6. Налаштування параметричного циліндра в Blender*

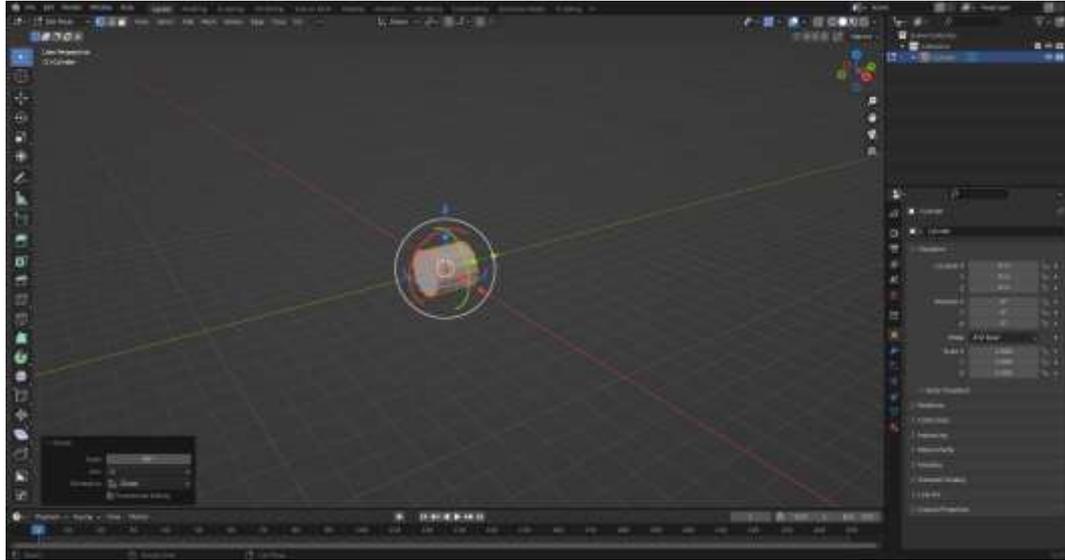
Для зручності, у верхньому лівому куті, в другій від центру вкладені, під назвою “Gizmos”, можна увімкнути відображення всіх трьох “гізмо”. Це інтерактивний елемент інтерфейсу для відображення осі та координат з можливістю взаємодіяти з ними для трансформацій.



*Рис. 3.7. Увімкнення додаткових елементів інтерфейсу в Blender*

Для більш швидкої роботи, все-ще раціональніше використовувати комбінації клавіш. Натискання клавіші “R” викликає функцію повороту, “Rotate”, далі, натискаючи клавішу “X”, обирається відповідна ось-

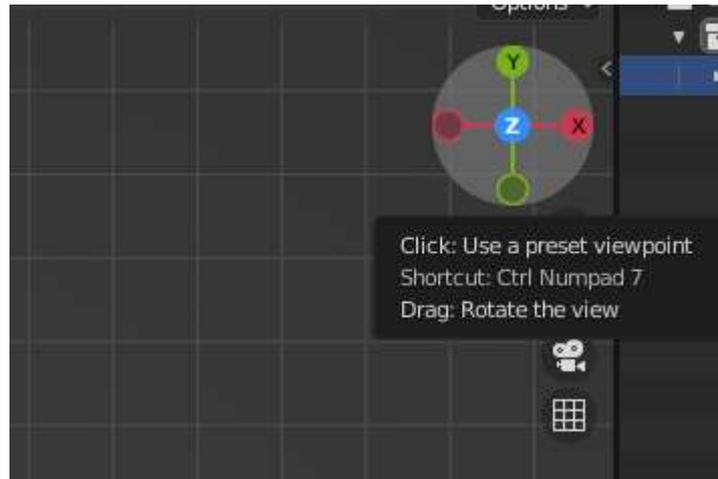
координата X, і тепер, рухаючи курсором миші - виконується обертання по обраній осі-координаті, але якщо набрати на клавіатурі цифри, задається конкретне значення. Натискаючи “9” та “0” задається значення 90 для повороту по осі-координаті X. Лівим натисканням миші, або клавішею “Enter” відбудеться підтвердження дій.



*Рис. 3.8. Приклад обертання фігури в Blender*

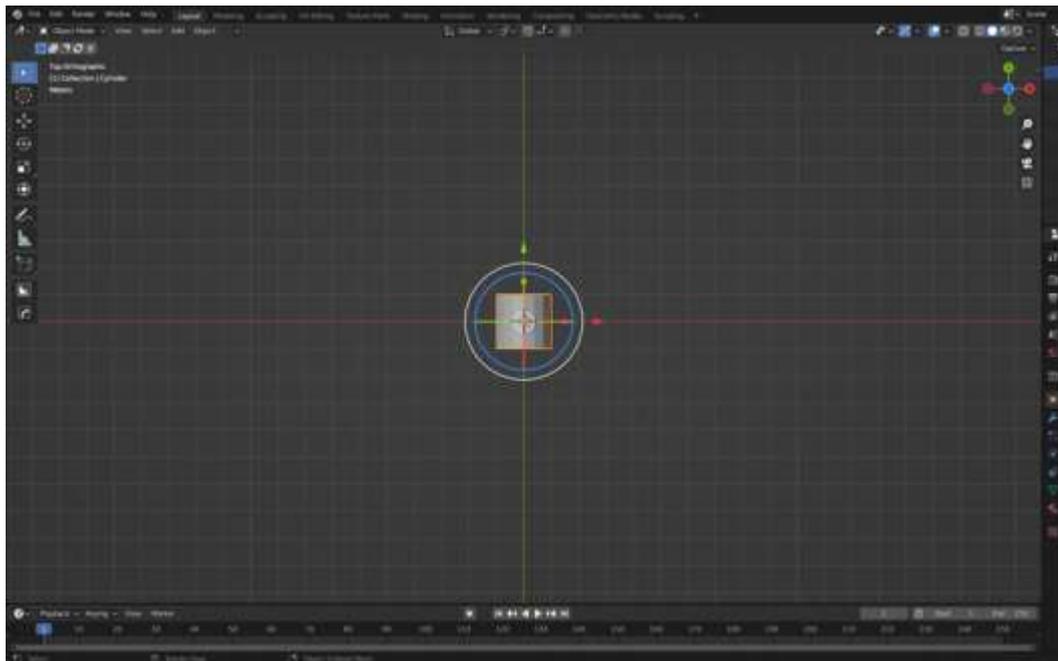
Для зручності роботи і зрозумілості підсумкового результату, потрібно працювати за наглядними прикладами, для цього їх можна додати прямо у тривимірну сцену чи змінити інтерфейс програми під себе та додати вікно з переглядом зображення, куди і завантажити приклади.

Для додавання власних зображень у сцену – достатньо просто перетягнути мишкою зображення з теки комп'ютера у робоче поле вікна Blender. Щоб зображення автоматично розмістилось у необхідній і зручній осі та позиції, потрібно завчасно налаштувати вигляд у робочому вікні програми. Найшвидший засіб задати вид у вікні, це натискання на осі координат на “гізмо” орієнтації у верхньому правому куті робочого віна.



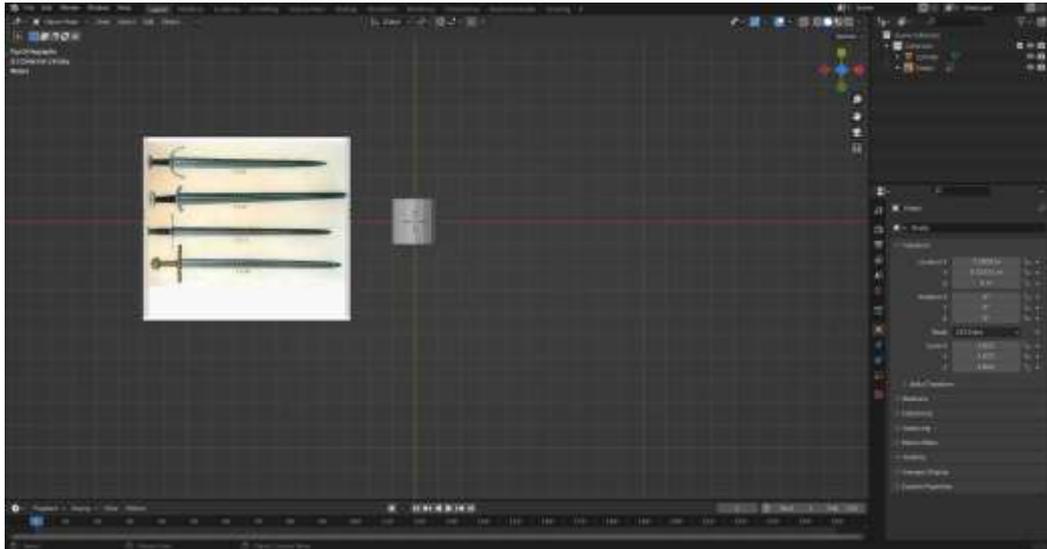
*Рис. 3.9. Взаємодія з гізмо в Blender*

Натискання на осі-координат автоматично переміщую вид камери відповідно до позиції в орієнтації “гізмо”, а ще, автоматично перемикає перспективу камери з “перспективи” на “ортографічну”.



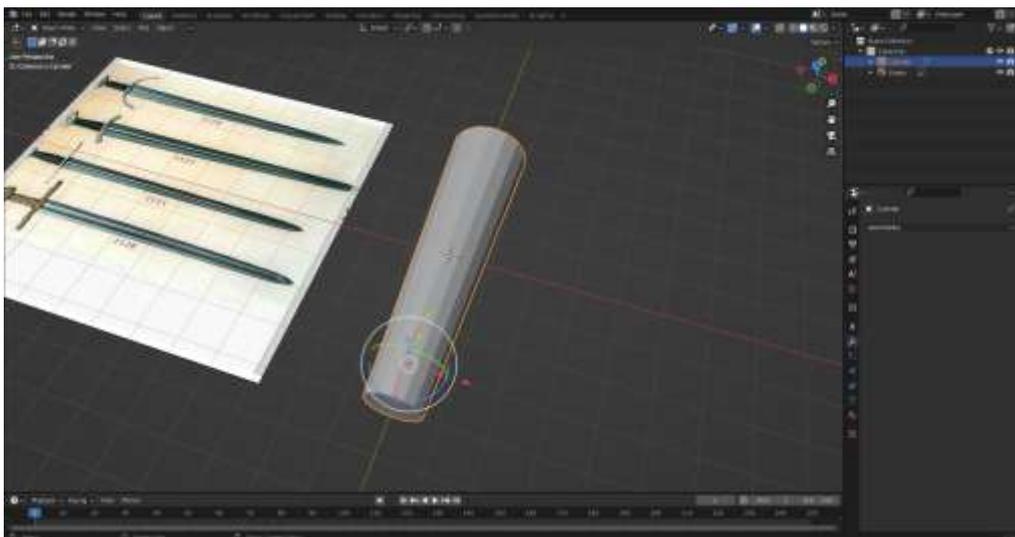
*Рис. 3.10. Приклад ортографічного виду в Blender*

Далі просто перетягую необхідні зображення з прикладами, на які можна орієнтуватися при подальшому моделюванні та заданні форми.



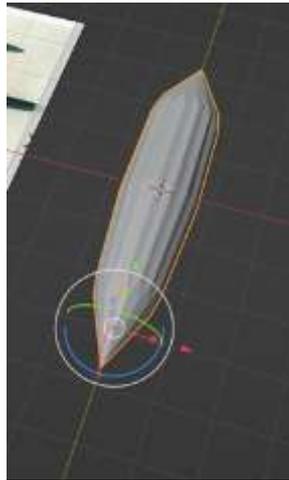
*Рис. 3.11. Приклад додавання зображень у робоче поле в Blender*

Клавішою “Tab” переходячи у режим редагування полігонів з обраним циліндром, рухаючи безпосередньо “вертекси” (вершини), йде етап редагування та задання форми циліндру, а саме – затисканням миші виділення набору вершин з одного боку і витягування його далі, чим відбувається розтягування циліндра по довжині. Для руху можна використовувати як увімкнені стрілки “гіздо”, так і комбінації клавіш, клавіша для роботи з пересуванням – “G”. За таким самим принципом слід від масштабувати один край, звужуючи його та задаючи форму подібну конусу.



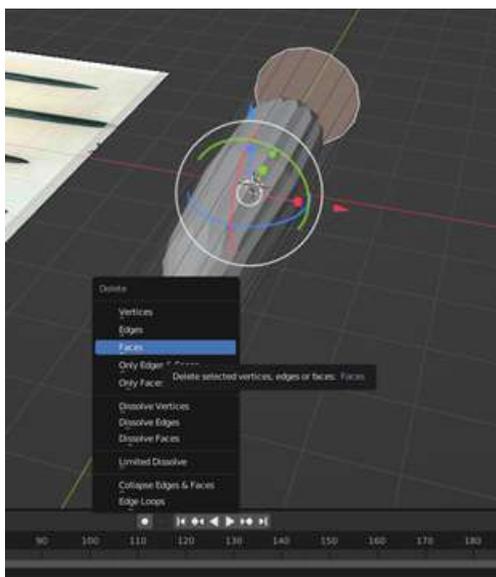
*Рис. 3.12. Редагування вершин шляхом трансформації в Blender*



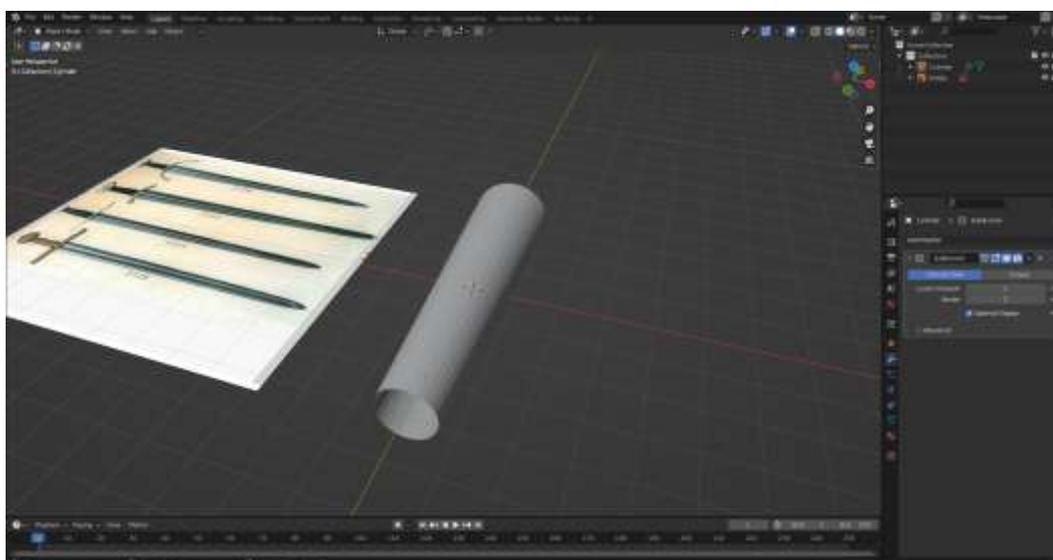


*Рис. 3.15. Утворення кривої форми при роботі модифікатора в Blender*

Це відбувається через неправильну сітку моделі, а правильніше - топологію. На кришках циліндру, які закривають його, по суті – утворились “n-gones”, неправильні полігони в яких більше чотирьох вершин. Такі полігони самі по собі некоректні, ще наочніше це стає під впливом модифікаторів чи інших деформаціях. Поки форма незакінчена, немає сенсу виправляти цей фрагмент, раціональніше – видалити ці “n-гони” для подальшого редагування вже після, якби їх видалення заважало подальшій роботі одразу, то їх виправленням варто зайнятись одразу. Обравши “n-gone” в режимі полігонального редагування достатньо видалити його, і одразу буде помітно значно кращий результат.

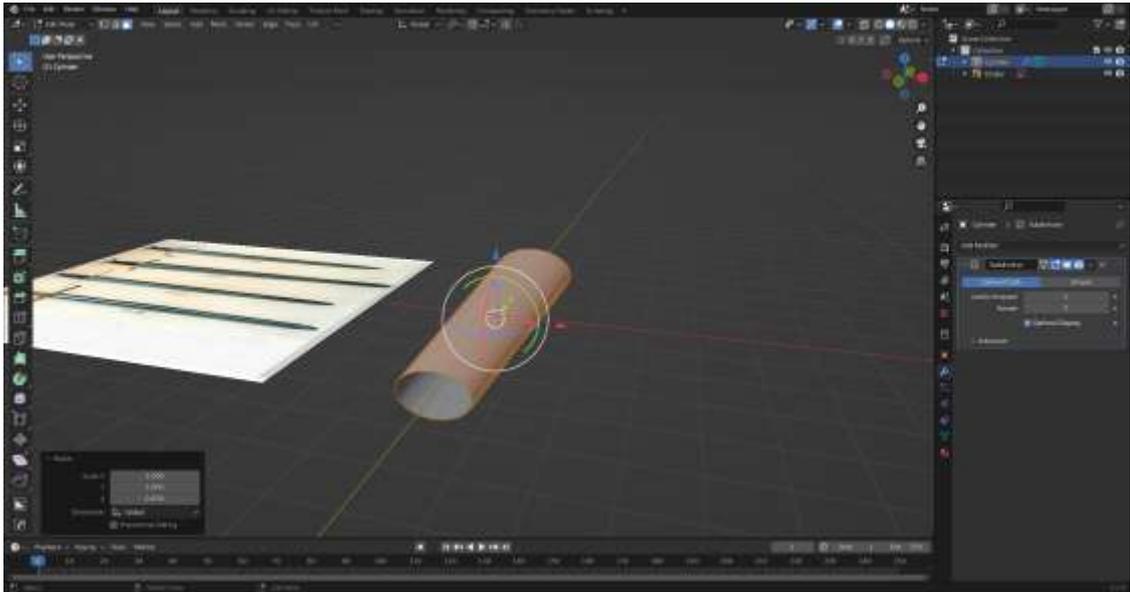


*Рис. 3.16. Видалення полігону в Blender*



*Рис. 3.17. Приклад виправленої форми без N-gone в Blender*

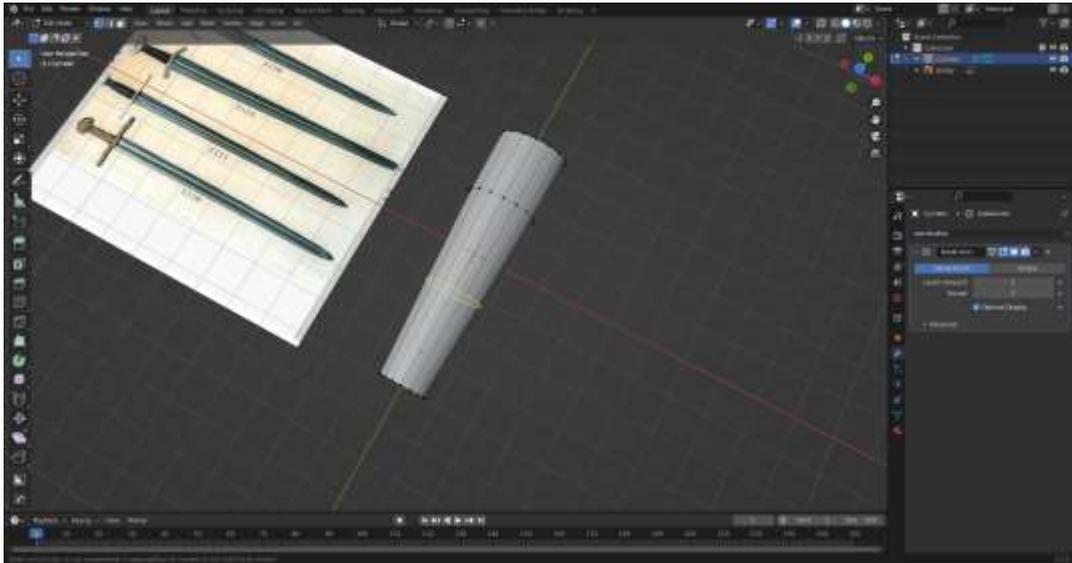
Продовжуючи задавати форму моделі, створюючи рукоятку, візуально помітно що циліндр необхідно сплюснути, з більшості можливостей виконати цю дію, достатньо вибрати наступний алгоритм дій - просто виділити всі вершини та від масштабувати їх по потрібній осі, тим самим сплюснувши фігуру.



*Рис. 3.18. Сплющення геометрії в Blender*

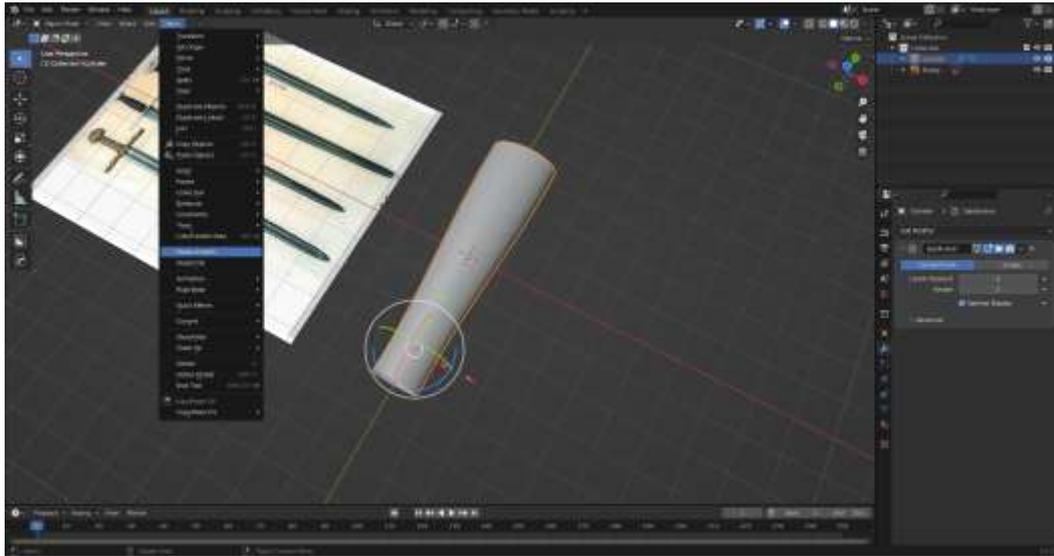
Розтягнувши, для задання потрібної форми рукоятки, точно необхідно більше точок-вершин у відповідних місцях змін, бо неможливо зігнути там, де відсутні необхідні ребра, точки, і полігони.

Комбінацією клавіш “Ctrl + R” активується функцію “розрізання” для додання нових ребер, спочатку додається нове ребро по центру моделі, наводячи курсор миші, відповідно – до центру. Далі, створене ребро при руху курсору миші, автоматично рухається, наче по рейкам, вздовж відповідних ребер, по своїй суті – рухаючись вздовж “face”, тобто, по поверхні полігону. Далі, нові розрізи вже можна додавати між новими утвореними та трансформованими ребрами, і вже існуючими.



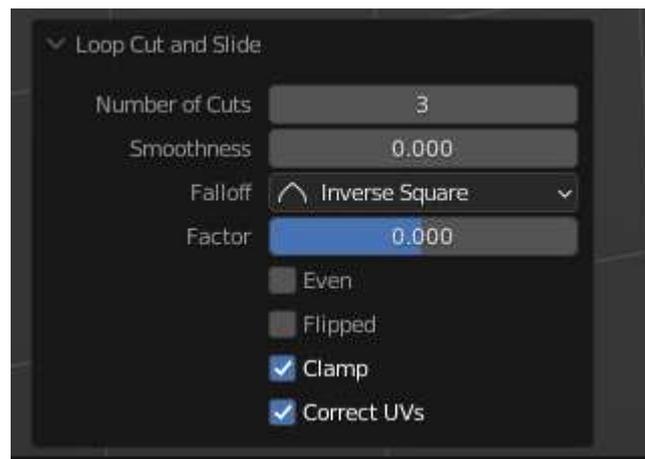
*Рис. 3.18. Додавання нових ребер в Blender*

Досі робота модифікатора “Subdivision Surface” помітна, кількість полігонів, точок та ребр – різна. У режимі відображення об'єктів відображається згладжена і підрозділена версія моделі з режиму полігонального редагування, саме так відображається робота модифікатора, але у вікні програми чітко видно кожне ребро. Все через вимкнений “shading”, відповідно – гладке відображення полігональних моделей яке активно використовується у своїй сфері комп'ютерної графіки. Щоб увімкнути “shading” для наших об'єктів, достатньо у режимі об'єктів зайти в панель “Object” та натиснути “Shade Smooth”.



*Рис. 3.19. Включення згладженого відображення в Blender*

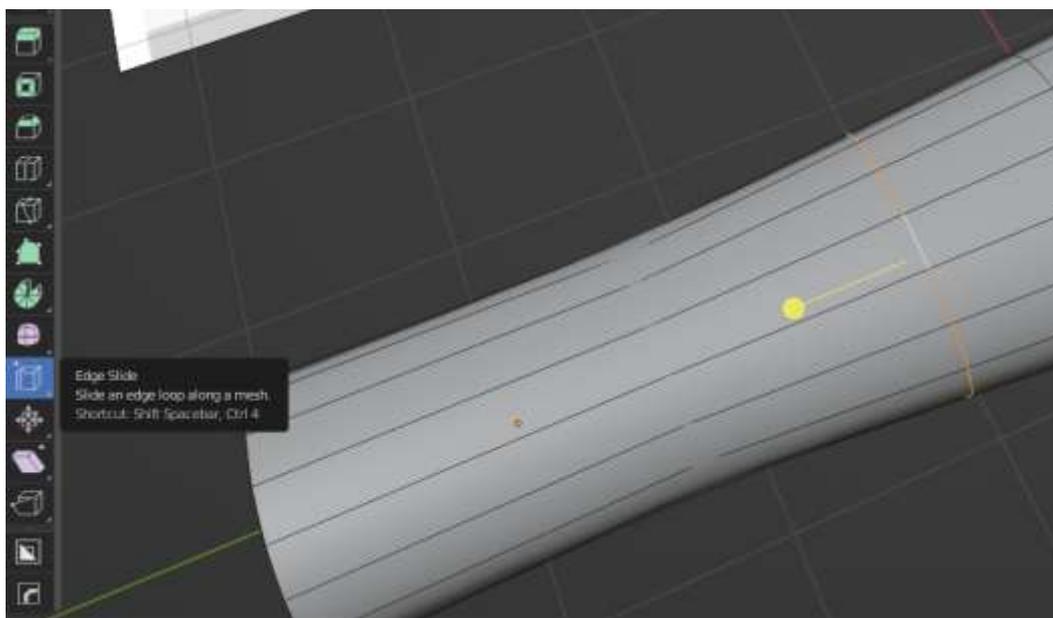
Для деталізації і ускладнення фігури моделі вже пройденими алгоритмами дій і використаними інструментами просто додаються на її поверхню нові деталі чи елементи. Для цього додаються нові ребра по центру комбінацією “Ctrl+R” та трьома короткими рухами колесиком миші вгору для збільшення кількості нових ребер, також це можна зробити в контекстному вікні налаштування останньої дії в нижньому лівому куті, змінюючи параметр кількості розрізів, “Numbers of Cut”.



*Рис. 3.20. Налаштування дії інструменту додання ребер в Blender*

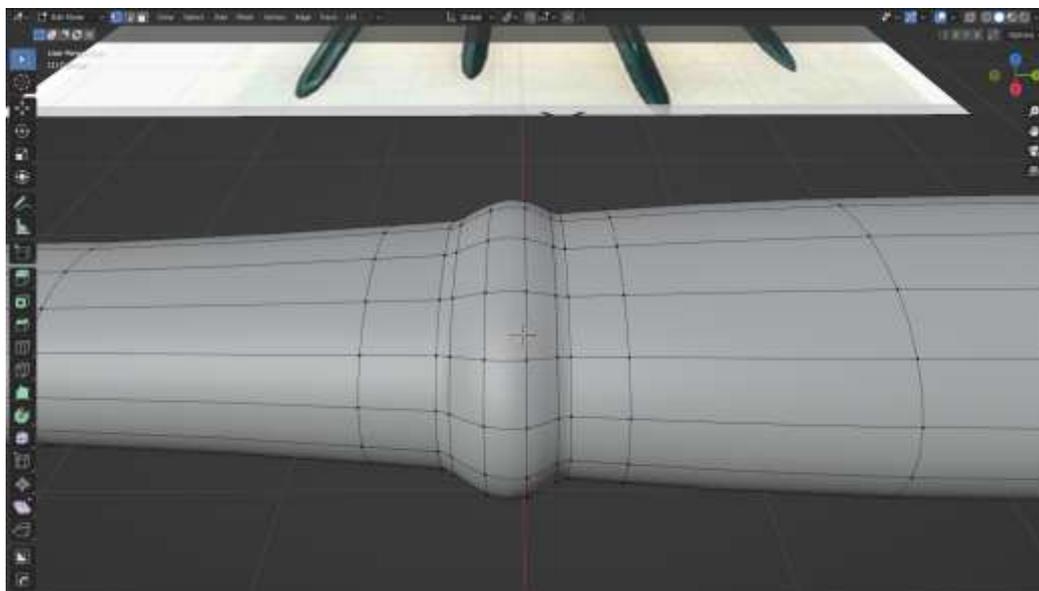
Подвійним натисканням клавіші “G” з обраним кільцем ребер, активується трансформація позиції відносно поверхні полігонів на яких ці ребра розміщені, або вершин вздовж ребер. Таким чином розміщуються

кільця в необхідній позиції одне відносно одного. Також ця функція представлена в панелі інструментів моделювання, під назвою “Edge Slide”.



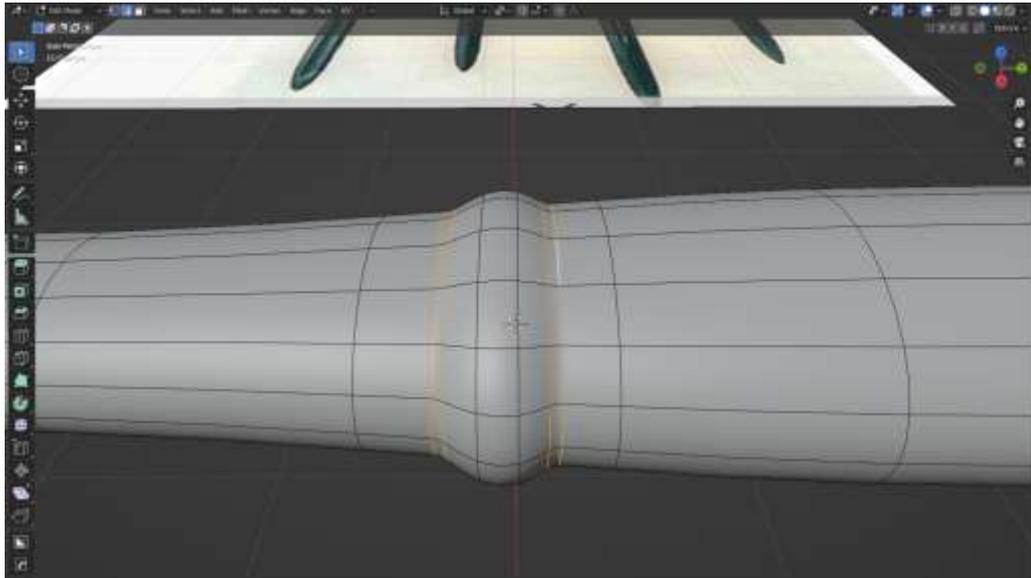
*Рис. 3.21. Демонстрація інструменту "Edge Slide" в Blender*

Додаючи нові ребра та переміщаючи їх і існуючі, задається зручна топологія (сітка) для трансформації фігури.



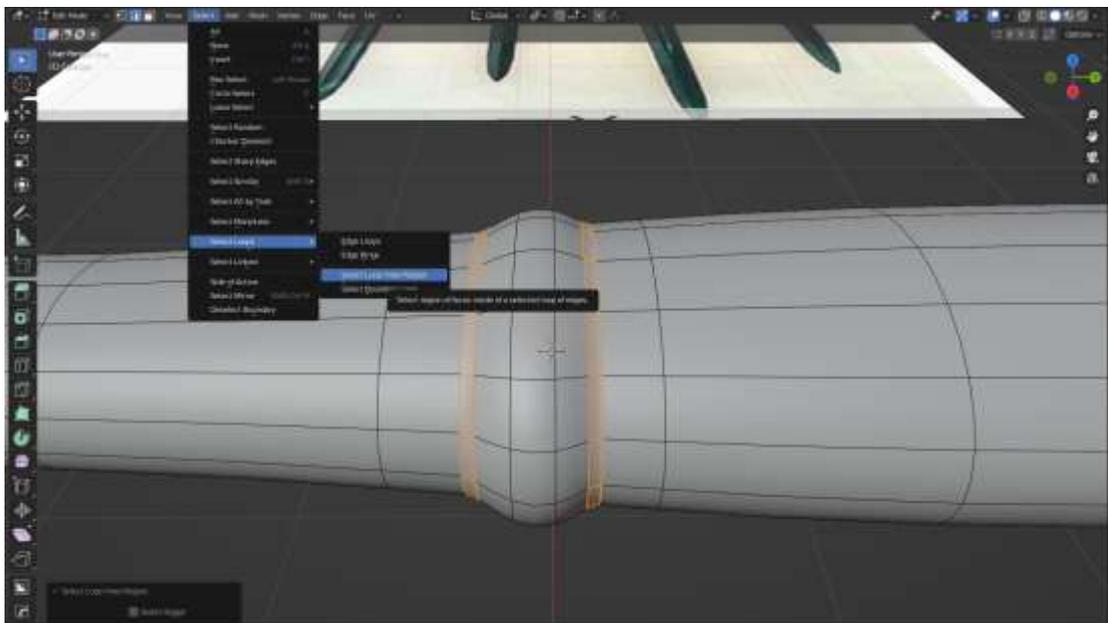
*Рис. 3.22. Огляд змін геометрії після використання дії перед цим*

Користуючись комбінацією клавіш з зажатою “Ctrl” та подвійним лівим натисканням миші, на одному з ребер умовного кільця – виділяються усе кільце, а далі, затиснувши клавішу “Shift” і таким самим принципом – виділяються інші необхідні кільця.



*Рис. 3.23. Виділення ребер як замкнутих кілець в Blender*

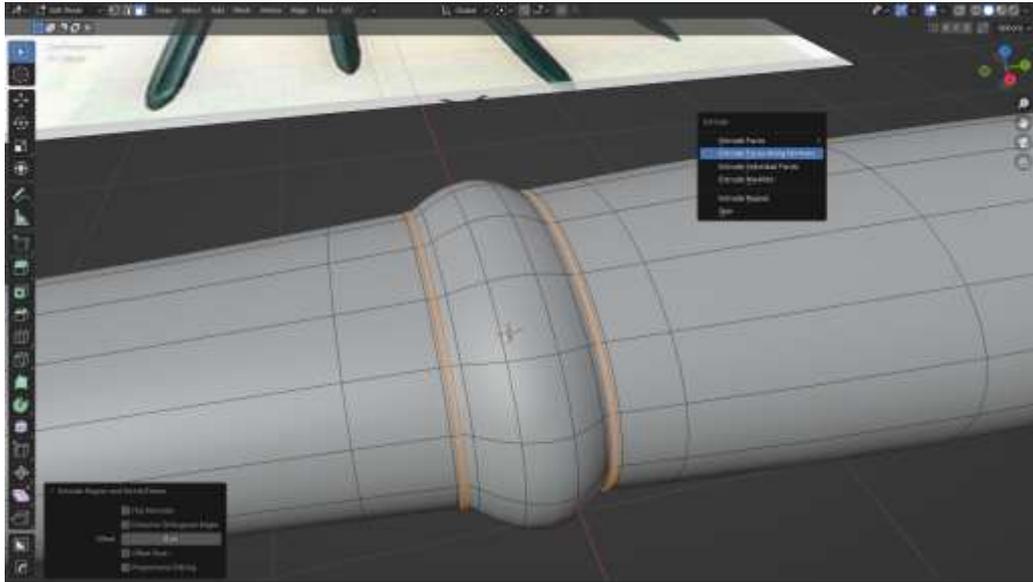
Щоб скористатись функцією “Extrude” та видавити чи вдавнити всередину полігони, які розміщені всередині виділених ребер, необхідно скористатись функцією виділення “Select”, далі “Select Loops” і функція під назвою “Select Loop Inner-Region”.



*Рис. 3.24. Застосування функції "Select Loop Inner-Region" в Blender*

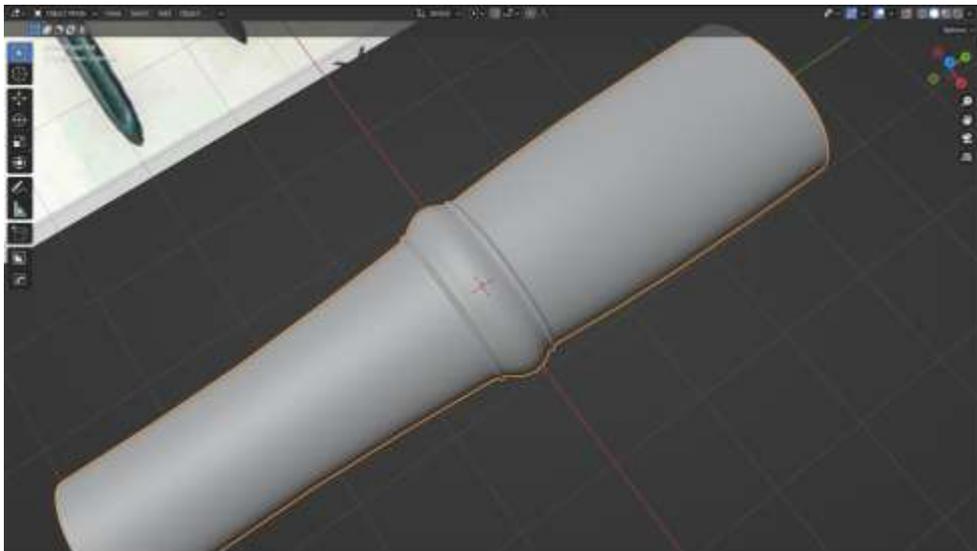
Далі для видавлювання використовується функція “Extrude”, під гарячою клавішею “E”, але, щоб коректно виконати функцію, видавлювання потрібно виконати по напрямленню відносно нормалей самих полігонів. Для

цього, використовується комбінація клавіш “Alt + E”, та обирається варіант “Extrude Faces Along Normals”.



*Рис. 3.25. Використання інструменту "Extrude Faces Along Normals" в Blender*

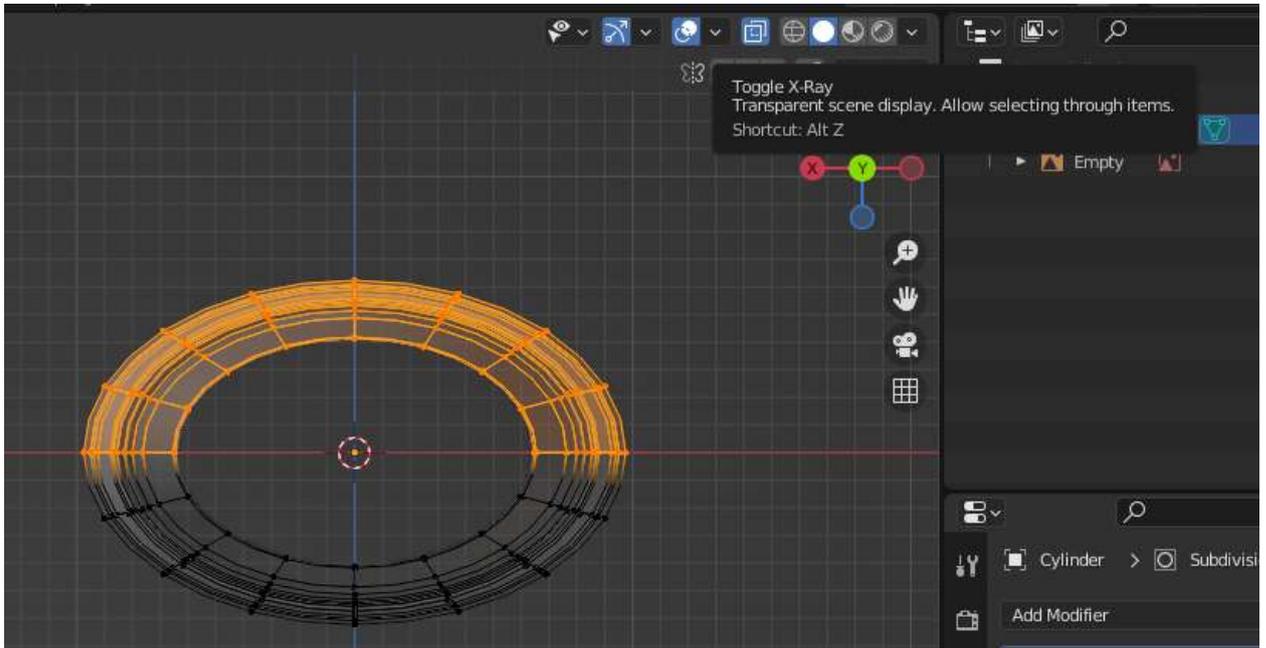
Рухаючи курсор миші – в одну чи іншу сторону виконується зсув від початкової позиції, і з врахуванням модифікатору “Subdivision Surface” формується геометрія.



*Рис. 3.26. Огляд отриманої геометрії з використанням "Subdivision Surface"*

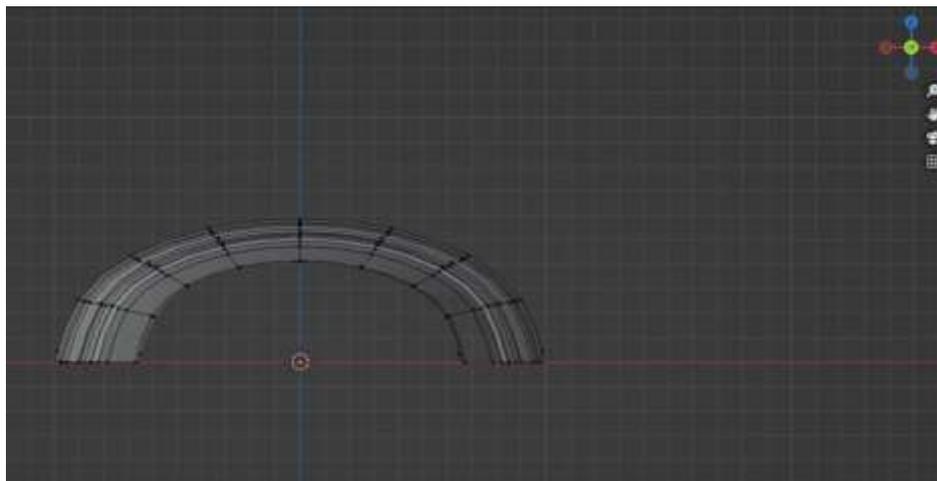
Форма фігури рукоятки занадто округла, для її сплющення і коректності внесення змін у модель, рішенням буде розділ моделі по центру по осі на вибір, та використання модифікатору віддзеркалення. Модифікатор

“Mirror” виконує віддзеркалення моделі відносно її центральної осі, та по осі координат за вибором в його налаштуваннях. Перед застосуванням, бажано для початку видалити одну половину моделі, що можна зробити обравши вид відображення в “Гізмо”, і в ортогональному режимі з режимом рентгену, або коректніше – “X-Ray”.



*Рис. 3.27. Увімкнення відображення "X-Ray" в Blender*

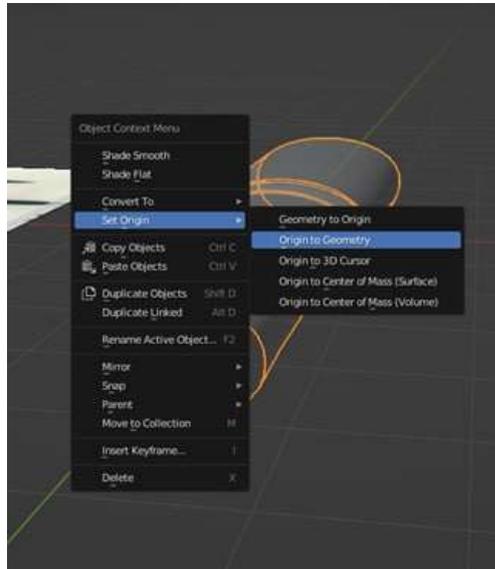
Далі сплюсшуючи верхню половину вершин і полігонів задається менш округла форма, після чого виділяється нижня частина по центральній вершині, для майбутньої симетрії, та видаляється.



*Рис. 3.28. Поділ та вирівнювання геометрії по середині*

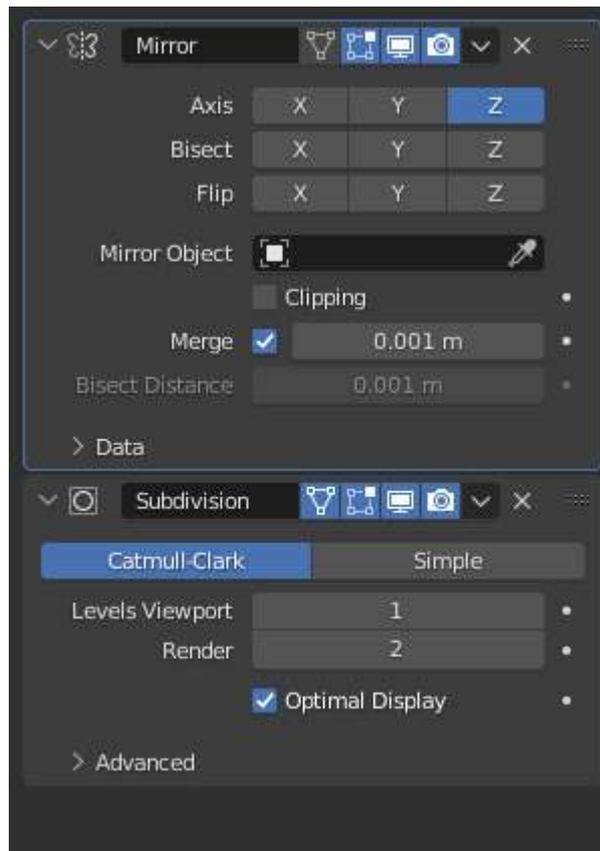
Центральна ось, або “Origin” - мусить знаходитись у геометричному центрі моделі, якщо це не відповідає дійсності, перед видалення одної з

половин, необхідно застосувати розміщення центральної осі моделі функцією “Set Origin” та “Origin to Geometry” в режимі об'єктів, з контекстного меню на правому кліку миші.



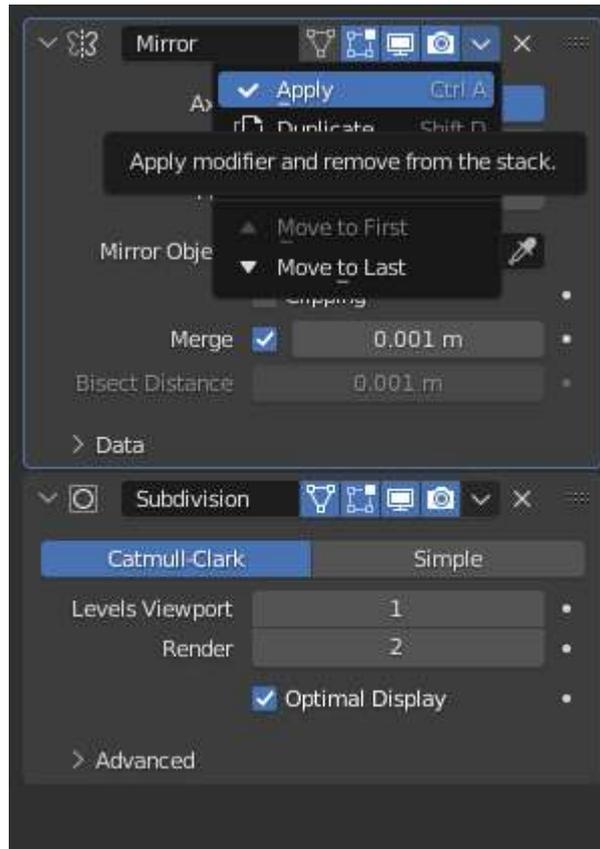
*Рис 3.29. Зміщення центральної осі моделі у геометричний центр в Blender*

Після видалення та задання симетрії використанням модифікатора “Mirror”, його необхідно налаштувати. Так-як нижню частину видалено то віддзеркалення за модифікатором необхідно виконати за відповідною координатою. Окрім того, необхідно слідкувати за ієрархією роботи модифікаторів у списку панелі модифікаторів та перетягнувши, розмістити роботу модифікатора “Mirror” перед роботою модифікатора “Subdivision Surface”.



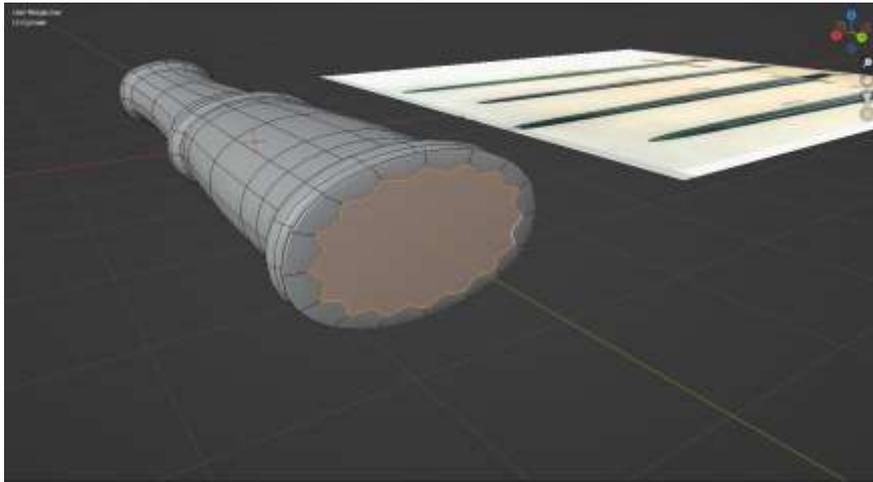
*Рис 3.30. Налаштування модифікаторів "Mirror" та "Subdivision Surface" в Blender*

В режимі редагування об'єктів в панелі модифікаторів в разі необхідності, роботу модифікатора можливо підтвердити, тим самим видаливши модифікатор і результат його роботи перенести в модель. З метою подальшого редагування, слід підтвердити роботу модифікатору і додати нову деталізацію на поверхні моделі.



*Рис. 3.31. Застосування дії модифікатора в модель в Blender*

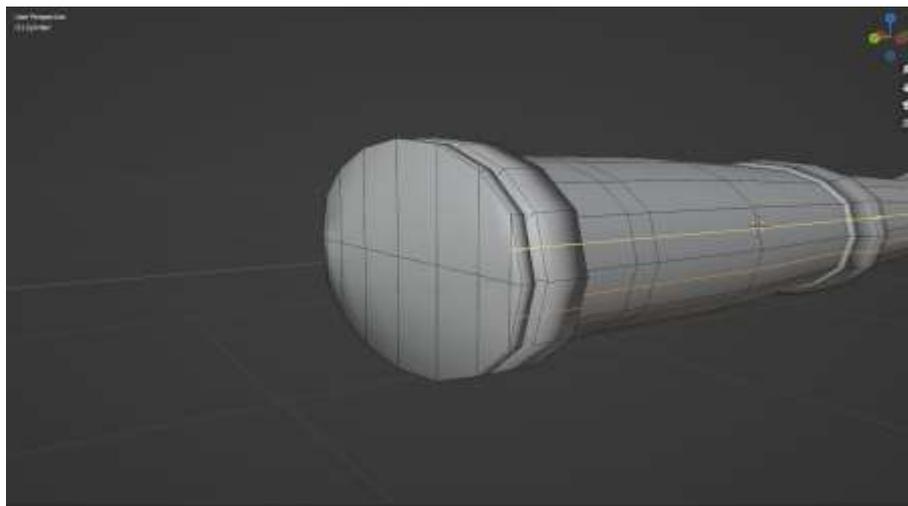
Для “зшиття” дірок у полігональній сітці є цілий ряд інструментів та засобів, як і для протилежного. Якщо необхідно досягнути конкретної і ідеальної топографії у сітці, доведеться зашивати отвори вручну. Для цього, для початку, можна виділити ребра навколо отвору, тобто кільцем, і клавішею “F” – утворити “N-gone” який закриватиме собою отвір.



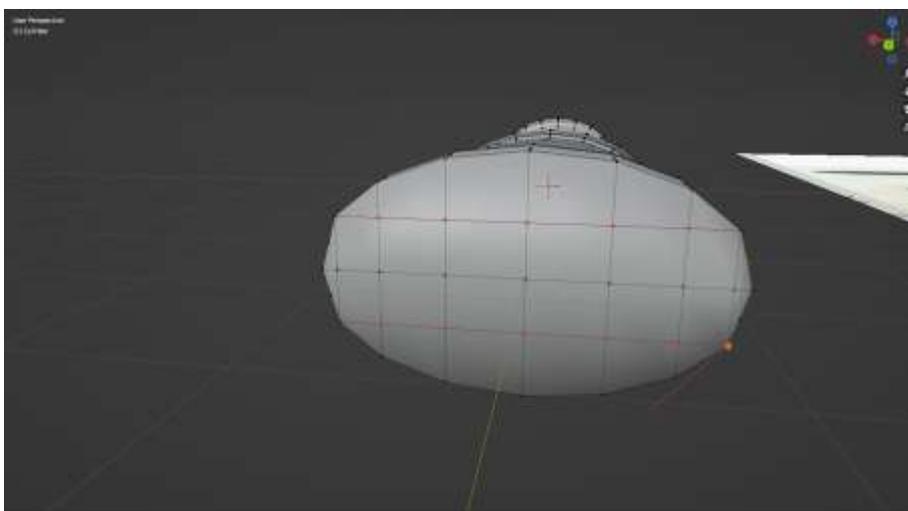
*Рис. 3.32. Утворення "N-gone" від дії модифікатора "Subdivision Surface"*

Далі, з застосуванням інструменту "Knife" на гарячу клавішу "K", обираються вершини країв отвору, відповідно додаючи на "n-gone" ребра. Таким чином, вручну утворюються необхідні полігони з чотирьох кутів для збереження технічної та естетичної правильності топології. Така сітка буде правильно реагувати на дію модифікаторів та правильним чином відобразитись при візуалізації.

Поділ сітки не дозволяє утворити виключно квадратні полігони на кришці що закриватиме отвір, тому вимушено вручну додаються нові ребра там, де це дозволить кожному полігону мати чотири вершини. Використати для цього можна вже відомий інструмент "Loop Cut" з комбінації клавіш "Ctrl + R".

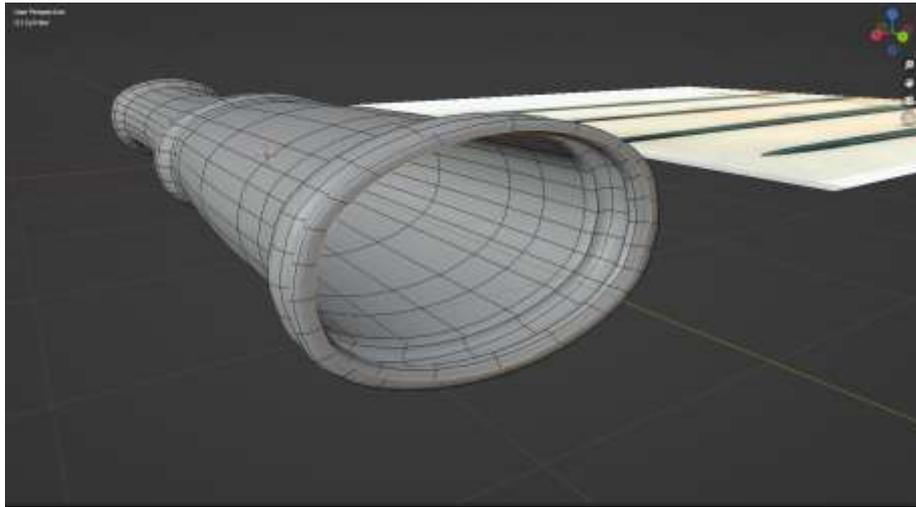


*Рис. 3.33. Розділ полігонів моделі використанням інструменту "Edge Cut"*



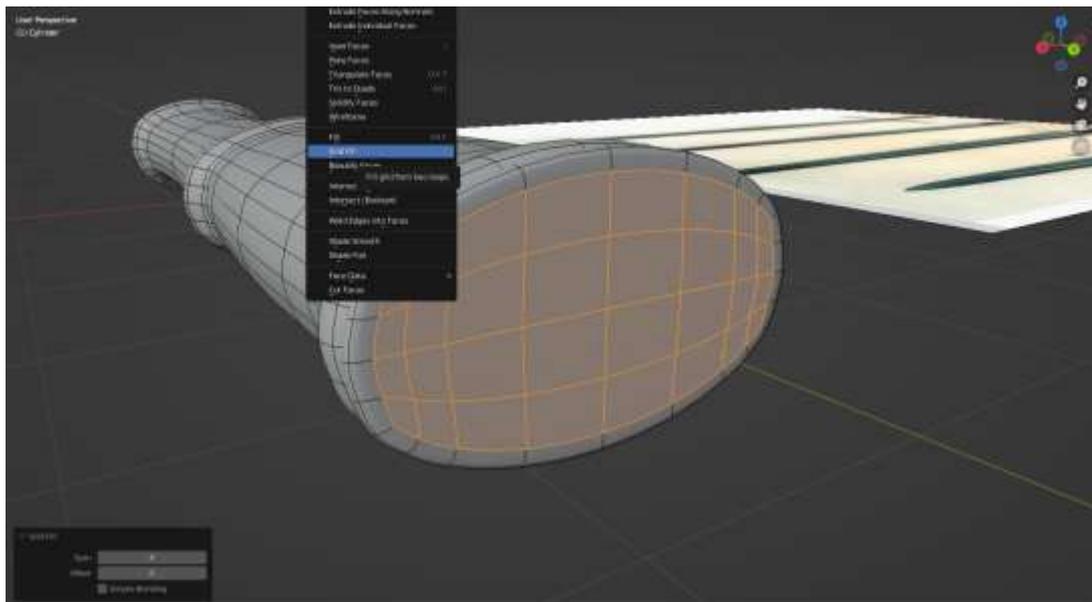
*Рис. 3.34. Використання інструменту "Knife" для розділу полігонів та утворення нових*

Нажаль, подібний алгоритм дій не дає правильного результату і на поверхні згладженої топології виникають артефакти, похибки. Тому, обрано один з інших можливих варіантів оптимізації топології. Натиском клавіші "Г" що викликає функцію "Inset Faces", утворюється з кільця ще одне, внутрішнє кільце.



*Рис. 3.35. Утворення в виділеному кільці нового завдяки інструменту "Inset Faces"*

Далі, при виділеному кільці ребер, вручну активується функція "Grid Fill" з вкладки "Face".



*Рис. 3.36. Застосування функції "Grid Fill"*

З використанням всіх описаних вище дій та принципів моделювання, створюється цілий набір готових моделей для їх подальшого використання у редакторі генерування з складення нової моделі з існуючих, власно створених.



*Рис. 3.37. Огляд всіх створених моделей в Blender*

### 3.3. Реалізація інтерфейсу користувача та функціоналу генерування моделей

Для реалізації застосунку – програмною платформою обрано ігровий рушій Unreal Engine, вибір зроблено на основі результатів дослідження поширених інтерактивних тривимірних платформ, і виборі саме ігрових рушіїв в цьому напрямі, сам аналіз і дослідження поширених рушіїв у Додатках. (додаток Б). Розробники програм використовують графіку і текст для відображення необхідної інформації, це називається інтерфейсом користувача (user interface, UI). UI в Unreal Engine створюється за допомогою Unreal Motion Graphics (UMG). UMG дає змогу зручно вибудовувати UI, перетягуючи елементи UI, як-от кнопки та текстові мітки. UMG UI Designer складається з семи основних елементів:

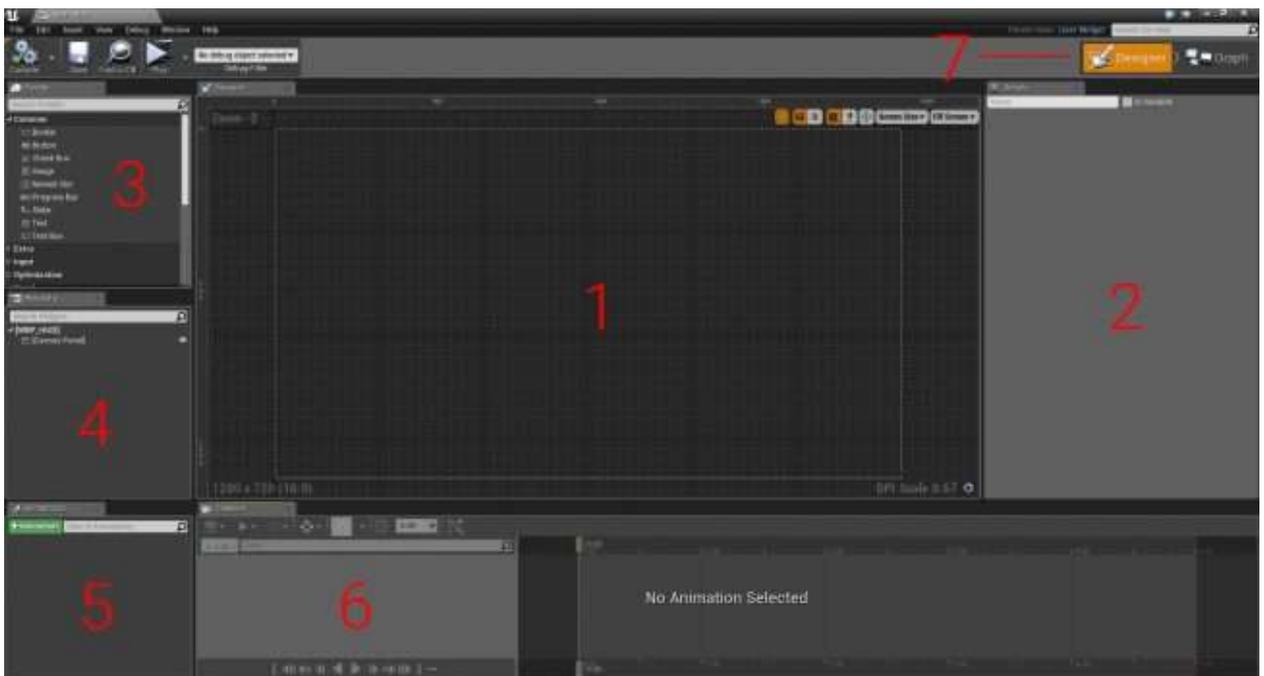


Рис. 3.38. Інтерфейс редактору UMG

1. Designer: у цій області представлено візуальне відображення віджета. Переміщатися по ній можна затиснувши праву клавішу миші та рухаючи мишею. Масштабування виконується прокручуванням коліщатка миші.
2. Details: тут відображаються властивості обраного віджета

3. Palette: список усіх віджетів, які можна використовувати. Усі створені користувачем віджети теж з'являються тут.
4. Hierarchy: список усіх уже використовуваних віджетів
5. Animations: деякі властивості віджетів можуть мати анімацію, наприклад, розташування і розмір. У цій панелі перераховані всі анімації.
6. Timeline: під час вибору анімації на цій панелі показуються анімовані властивості та ключові кадри
7. Editor Mode: тут можна перемикатися між режимами Designer і Graph. Режим Graph майже аналогічний Event Graph у Blueprint.

Щоб створювати елементи інтерфейсу на екрані користувача, потрібно створювати віджети. Віджет (widget) - це елемент UI, що надає UI візуальні функції. Наприклад, віджет Button надає об'єкт, який користувач може бачити і натискати на нього. Сам віджет необов'язково має бути видимим. Наприклад, віджет Grid Panel рівномірно розділяє свій простір між його вмістом. Користувач не може побачити Grid Panel, але бачить його вплив. Крім того, віджети можуть містити інші віджети. Вікно створення і редагування елементів користувацького інтерфейсу поділено на два принципово різних редактора. Візуальний редактор графічних елементів інтерфейсу та редактор візуального програмування Blueprint для програмування цих елементів.

Далі наочно розглянуто виконання елементів інтерфейсу. Головне меню з кнопками переходу з вікна головного меню у вікно генерування, та кнопкою завантаження збереженої сцени. Більше прикладів у додатках (додаток В).

Інтерфейс вікна генерування тривимірної сцени з компонентів:



*Рис. 3.39. Огляд програмного коду інтерфейсу головного меню мовою "Blue Print" в редакторі "Unreal Engine"*

Комірка вибору компоненту для генерування зброї:

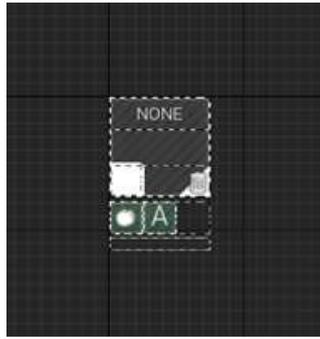


Рис. 3.40. Дизайн комірки вибору компоненту сцени в редакторі "Unreal Engine"

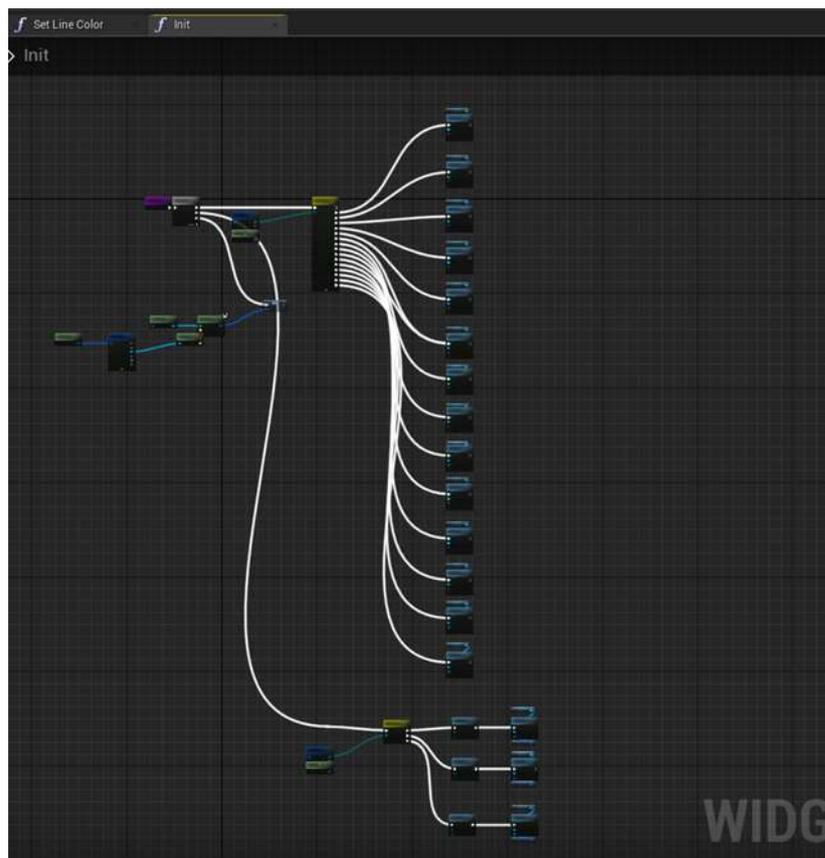


Рис. 3.41. Програмний код комірки вибору компоненту сцени мовою "Blue Print" в редакторі "Unreal Engine"

Далі розглянемо реалізацію самого функціоналу застосунку, логіку його роботи, принцип створення. Функціонал створюватиметься завдяки візуальній мові програмування "Blueprint", як і робота інтерфейсу.

**Blueprints** - це система візуального скриптингу Unreal Engine. Вона є швидким способом створення прототипів ігор. Замість порядкового написання коду все можна робити візуально: перетягувати ноди (вузли), задавати їхні властивості в інтерфейсі та з'єднувати їхні "дроти". У

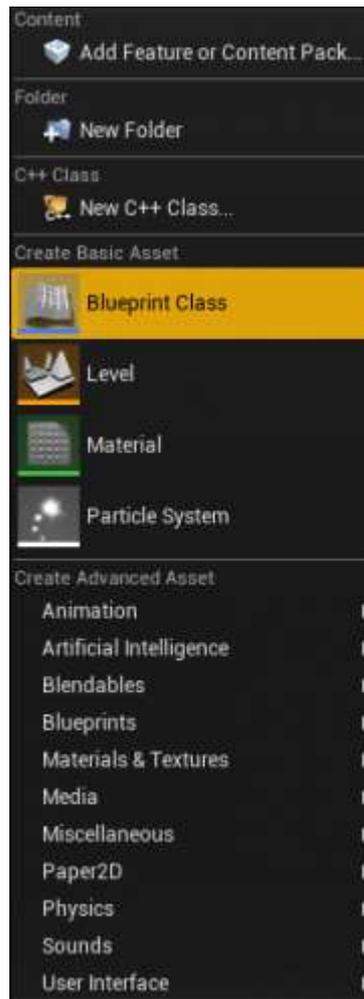
найпростішому випадку Blueprint являє собою "річ". Blueprints дають змогу створювати свої поведінки для об'єктів. Об'єкт може бути чимось фізичним (типу поворотного столу) або чимось абстрактним, наприклад, системою здоров'я. Сама система програмування представлена у вигляді логістичних ланок які складаються з набору вузлів або нодів, кожен з яких виконує якусь свою дію чи функцію, отримуючи інформацію від попереднього чи попередніх та передаючи її далі.

Blueprints прості у використанні, проте не такі швидкі, як код на C++. Тобто якщо вам потрібно створити щось "важке" з точки зору обчислень, наприклад, складний алгоритм, то краще скористатися C++.

Але навіть якщо ви віддаєте перевагу C++, то бувають випадки, коли оптимальніше використовувати Blueprints. Ось деякі з переваг Blueprints:

- Зазвичай розробка на Blueprints швидша, ніж на C++
- Простота впорядкування. Можна розділяти ноди на різні області, наприклад, на функції та графи
- Якщо ви працюєте з людьми, які не знають програмування, то зміна Blueprint простіша завдяки їхній наочності та інтуїтивній зрозумілості

Для їх створення перейдіть у Content Browser і натисніть на Add New. Виберіть у списку Blueprint Class.



*Рис. 3.42. Контекстне меню створення в Content Browser*

Відкриється вікно із запитом вибору батьківського класу. Blueprint успадковуватиме всі змінні, функції та компоненти з обраного батьківського класу.

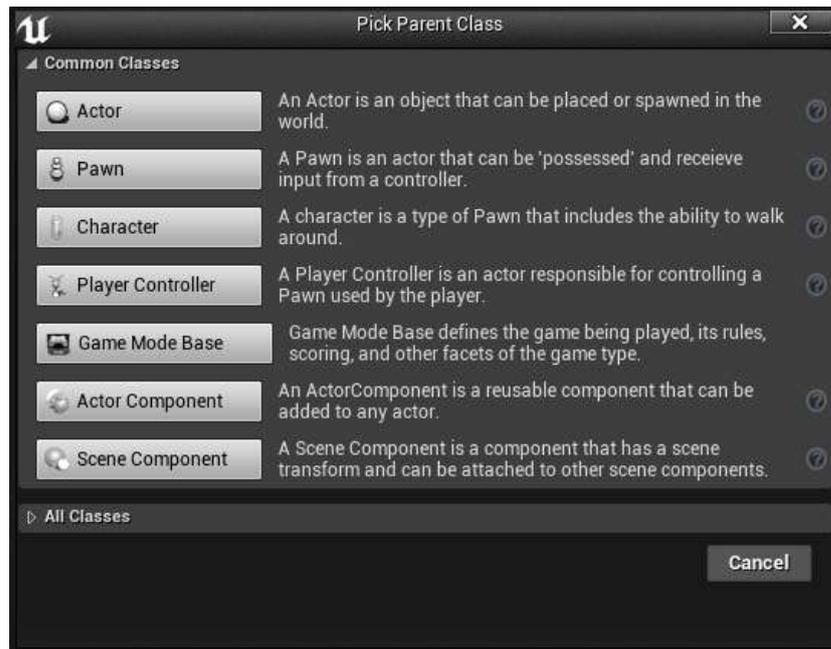


Рис. 3.43. Меню створення нового Blueprint

Редактор Blueprint. Blueprint editor складається з чотирьох основних панелей:

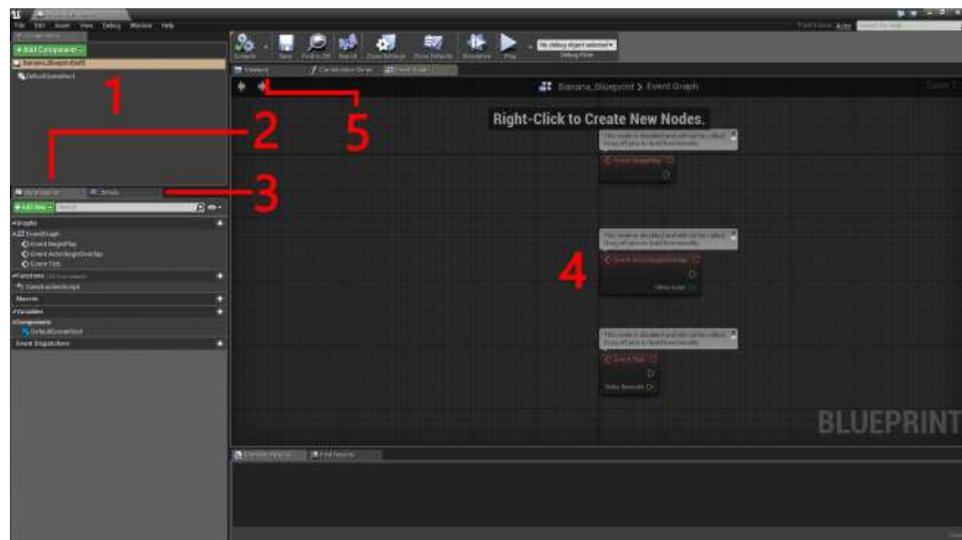


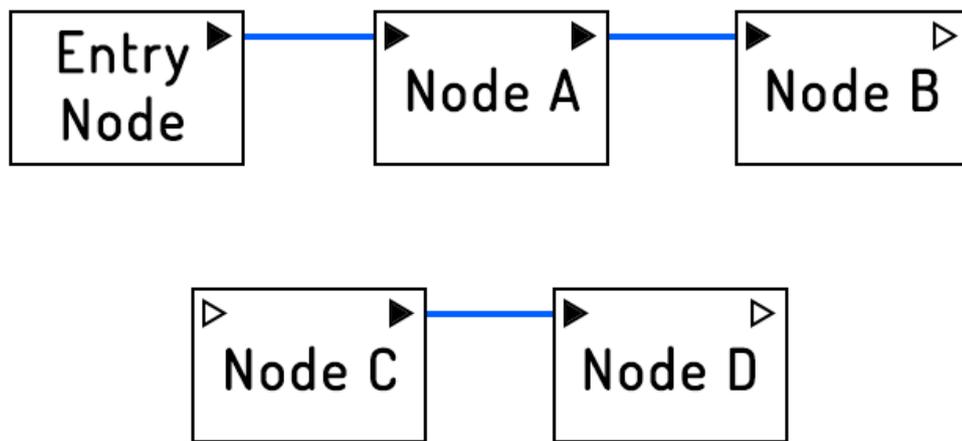
Рис. 3.44. Інтерфейс редактора Blueprint Editor

1. Components: містить список поточних компонентів.
2. My Blueprint: цей розділ зазвичай використовується для управління графами, функціями і змінними.
3. Details: тут відображаються властивості поточного обраного елемента.
4. Graph: саме тут відбувається магія. Усі ноди та логіка знаходяться тут. Переміщатися панеллю можна, затиснувши праву клавішу миші та

переміщаючи мишу. Масштабування виконується прокручуванням коліщатка миші.

5. Viewport: тут відображаються всі компоненти, що мають візуальні елементи. Переміщення та огляд виконуються так само, як і у Viewport основного редактора.

На відміну від своїх близьких родичів - нодів матеріалів - ноди Blueprint мають особливі контакти, звані контактами Execution. Контакт ліворуч - це вхід, контакт праворуч - вихід. У всіх нодів є хоча б по одному входу і виходу. Якщо нод має контакт входу, то його потрібно підключити, щоб він запрацював. Якщо нод не під'єднаний, усі наступні ноди не будуть виконуватися. Ось приклад:



*Рис. 3.45. Приклад роботи та взаємодії вузлів (нод)*

Node A і Node B будуть виконуватися, тому що у їхніх вхідних контактів є підключення. Node C і Node D ніколи не виконуються, тому що вхідний контакт Node C не має підключення.

Демонстрація реалізації програмного коду розглянуто в прикладах, розміщених у додатках (додаток Г).

## ВИСНОВКИ

В результаті проведення даного дипломного дослідження було виконано всі поставлені завдання. Зокрема:

- Проаналізовано і на основі функціональних можливостей з всіх поширених - обрано один редактор тривимірної графіки. Програма була обрана відповідно критеріїв, а саме – доступність та наявність інструментів що відповідають рівню сучасного 3D-редактора. Досліджено процес створення тривимірних моделей для подальшого використання досліджених методів і інструментів на практиці.
- Досліджено доступні програмні платформи, що відповідають можливості гнучкої та ефективної реалізації інтерактивного тривимірного застосунку на їх основі. На практиці досліджено можливість використання ігрового рушія як програмної платформи не для гри, а для редактора з інтерактивним тривимірним простором і сучасною візуалізацією (3D-рендерингом).
- Спроектовано редактор, а саме його інтерфейс та функціонал, та обрано візуальне спрямування для створюваних 3D-моделей, проведено роботу з пошуком прикладів для орієнтування при тривимірному моделюванні складних форм.
- Створено набір тематичних, складних тривимірних моделей за обраними технічними характеристиками та візуальними прикладами.
- Розроблено простий, гнучкий та інтуїтивний користувацький інтерфейс для програми з використанням інструментів інтегрованого редактору Unreal Motion Graphics.
- Реалізовано запланований функціонал програмного застосунку (з використанням мови візуального програмування Blurprints),

призначений для генерування тривимірних сцен, використовуючи створені тривимірні моделі.

Під час написання магістерської роботи я дослідив ринок тривимірних редакторів, детально ознайомившись з можливостями, інструментами, перевагами і недоліками кожного і в результаті обрав програму що відповідала моїм технічним потребам та зручному користувацькому досвіду. Користуючись інструментами 3D-редактора, створив набір власних тривимірних моделей в одному обраному художньому стилі, для чого було проведено роботу з пошуком прикладів і дослідженням фігур та структури об'єктів за візуальним прикладом яких і було створено тривимірні моделі. Досліджено ринок програмних платформ для створення інтерактивних застосунків, які підтримують роботу з тривимірним простором та рендерингом у реальному часі. Перевірено у використанні спроможності ігрового рушія Unreal Engine для створення власної програми-редактора, тобто досліджено використання ігрового рушія не за його прямим призначенням. На практиці використано для програмування інтегровану в рушій, візуальну мову програмування Blueprints, можливості ті інструменти якої цілком відповіли запиту реалізувати функціонал програми.

Функціонал і можливості створеного редактора можна розширювати до безкінечності, не порушуючи початкової структури та задуму програми. Є можливість додавати необмежену кількість нових тривимірних моделей для примноження кількості ітерацій фінальної сцени. Можливо повністю змінити художній напрям і замінити всі моделі на відповідні, отримати абсолютно новий застосунок для генерації нових сцен. Якщо додати в застосунок функціонал експорту сцени в поширених тривимірних форматах (OBJ; FBX), це автоматично примножить його практичні можливості, зробить його придатним та корисним у всіх сферах комп'ютерної графіки. Використання згенерованих сцен-моделей з редактору буде достатньо корисним в ігровій індустрії, індустрії візуальних ефектів та графіки, в художній сфері чи для тривимірного друку. Функціонал програми також можна розширити

інтеграцією нових можливостей внесення змін в сцену, наприклад інструменти базової трансформації чи корегування певних фрагментів моделі, наприклад - збільшення довжини леза, якщо говорити про створений застосунок. Також не слід ігнорувати можливість додання інструментів взаємодії з візуалізацією, це можуть бути такі нові функції, як: зміна кольору та відтінків текстурних мап, зміна масштабу текстурних мап, зміна насиченості текстурних мап типу Normal Map та Ambient Occlusion, зміна параметрів шейдерів, типу, параметру віддзеркалення для металевих поверхностей. Також можна додати можливість змінювати фон чи мапу оточення - HDRI.

Створення власного інтерактивного 3D-редактора на основі ігрового рушія відкриває двері для подальшого розвитку у цьому напрямі. Ніша допоміжних інструментів у комп'ютерній графіці на базі ігрових рушіїв замість створення, власне, ігрових застосунків – це перспективний напрям, подібних прикладів поки що мало, а подальше заглиблення в цей напрямок має дійсно чималий потенціал завдяки, як мінімум - простоті та зручності такого засобу реалізації своїх тривимірних застосунків. Такий нестандартний засіб використання ігрових рушіїв значно спрощує досягнення мети у розробці власних інструментів, які натомість вже можна використовувати за будь-яким спрямуванням в комп'ютерній графіці.

Згодом, для примноження можливостей програми, можна інтегрувати функціонал імпорту власних моделей для потенційного користувача. Таким чином, програма стає для користувача його власним, потужним інструментом для генерування будь-яких сцен за власною тематикою, залежно тільки від наявних 3D-моделей. Також, за задумом, функціонал поступово доповнюватиметься різними засобами деформування компонентів сцени, корегування їх візуалізації, типу вибору колірному тону для матеріалів з попередньо підготовлених палітр, вибір і налаштування текстур та текстурних мав, налаштування матеріалів, і в подальшому, створення

інструментів моделювання прямо в програмі, наприклад, інструментами тривимірного скульптингу.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сайт рушія Godot [Електронний ресурс]. Режим доступу до ресурсу: <https://godotengine.org//>.
2. Сайт рушія Unity3D [Електронний ресурс]. Режим доступу до ресурсу: <https://unity.com//>.
3. Загальний опис Unreal Engine [Електронний ресурс]. Режим доступу до ресурсу: <https://www.unrealengine.com//> .
4. Игорь Трефилов, Создание 3D моделей для игр — как это происходит? [Електронний ресурс]. Режим доступу до ресурсу: <https://dtf.ru/gamedev/651865-sozдание-3d-modeley-dlya-igr-kak-eto-proishodit//> .
5. PatientZero. Тьюторіал по Unreal Engine. Часть 1: знакомство с движком [Електронний ресурс]. Режим доступу до ресурсу: <https://habr.com/ru/post/344394//>.
6. OpenGL Wiki [Електронний ресурс]. Режим доступу до ресурсу: <https://www.khronos.org//>.
7. Загально про Blender [Електронний ресурс]. Режим доступу до ресурсу: <https://www.blender.org //>.
8. Допоміжна документація Blender [Електронний ресурс]. Режим доступу до ресурсу: <https://docs.blender.org//>
9. Допоміжна документація Godot [Електронний ресурс]. Режим доступу до ресурсу: <https://docs.godotengine.org/en/stable//>.
10. Сайт компанії розробника Сінема 4D [Електронний ресурс]. Режим доступу до ресурсу: <https://www.maxon.net//>.
11. Бібліотека Unreal Engine [Електронний ресурс]. Режим доступу до ресурсу: <https://dev.epicgames.com//>.
12. Документація Blender для розробників [Електронний ресурс]. Режим доступу до ресурсу: <https://developer.blender.org//>.

13. Орещенко А. В. Тривимірна графіка для кіно: виникнення і розвиток. Часопис картографії. 2013. Вип. 8. С. 181.
14. В. Є. Михайленко, В. В. Ванін, С. М. Ковальов. Інженерна та комп'ютерна графіка : підруч. для студ. ВНЗ, 5-те вид., К. : Каравела, 2010., 360 с.
15. Д. Херн, М. П. Бейкер. Комп'ютерна графіка й стандарт OpenGL. 3-е вид., М. : Вільямс, 2005., 1168 с.
16. Дж. Лі, Б. Уер. Тривимірна графіка та анімація., 2-е вид., М. : Вільямс, 2002., 640 с.
17. Е. Енджел. Інтерактивна комп'ютерна графіка. Вступний курс на базі OpenGL., 2-е вид., М. : Вільямс, 2001., 592 с.
18. Г. Снук. 3D-ландшафти в реальному часі на C++ і DirectX 9., 2-е вид., М. : Кудиц-прес, 2007., 368 с.
19. Уроки по Unreal Engine [Електронний ресурс] Режим доступу до ресурсу: <https://itproger.com/course/unreal-engine/>.
20. Веселовська Г. В. Комп'ютерна графіка: Навчальний посібник для вузів., Херсон: ОЛДІ-плюс, 2004., 582 с.
21. Програмування комп'ютерної графіки та мультимедійні засоби : навч. посіб. / Л. М. Журавчак, О. М. Левченко. Львів : Львівська політехніка, 2019., 276 с
22. Основи комп'ютерної графіки: курс лекцій / О. Я. Різник ; М-во освіти і науки, молоді та спорту України, Нац. ун-т «Львів. політехніка»., Л. : Вид-во Львів. політехніки, 2012., 220 с.
23. Pharr, Matt; Humphreys, Greg (2004). Physically based rendering from theory to implementation. Amsterdam: Elsevier/Morgan Kaufmann.
24. James D. Foley, Andries van Dam, Steven K. Feiner, John Hughes. Computer Graphics: Principles and Practice 2nd Edition in C / James D. Foley, Andries van Dam, Steven K. Feiner, John Hughes., 1995. – 1175 с.
25. Gambetta G. Computer Graphics from Scratch: A Programmer's Introduction to 3D Rendering / Gabriel Gambetta., 2021. 248 с.

26. Вольфф Д. OpenGL 4.0 Shading Language Cookbook / Девід Вольфф., 2011. 340 с.
27. Guha S. Computer Graphics Through OpenGL: From Theory to Experiments / Sumanta Guha., 2014. 951 с.
28. Luna F. Introduction to 3D Game Programming with DirectX 11 / Frank Luna., 2012. 600 с.
29. Gaboury J. Image Objects: An Archaeology of Computer Graphics / Jacob Gaboury., 2021. 312 с.
30. Kosarevsky S. 3D Graphics Rendering Cookbook: A comprehensive guide to exploring rendering algorithms in modern OpenGL and Vulkan / S. Kosarevsky, V. Latypov., 2021. 312 с.
31. Satheesh PV. Unreal Engine Game Development Essentials / Satheesh PV., 2016. 266 с.
32. LeE J. Learning Unreal Engine Game Development / Joanna LeE., 2016. 274 с.
33. Game Development Projects with Unreal Engine: Learn to build your first games and bring your ideas to life using UE4 and C++ / H.Fozi, G. Marques, D. Pereira, D. Sherry., 2020. 822 с.
34. Romero M. Blueprints Visual Scripting for Unreal Engine 5: Unleash the true power of Blueprints to create impressive games and applications in UE5, 3rd Edition / M. Romero, B. Sewell, L. Cataldi., 2022. 568 с. (3).
35. Nixon D. Beginning Unreal Game Development: Foundation for Simple to Complex Games Using Unreal Engine / David Nixon., 2020. 413 с. (1).
36. Unreal Engine Community Wiki [Электронный ресурс]. Режим доступа до ресурсу: <https://unrealcommunity.wiki/> / .
37. Blender Developer Documentation [Электронный ресурс]. Режим доступа до ресурсу: <https://wiki.blender.org/> / .

## ДОДАТКИ

## Додаток А.

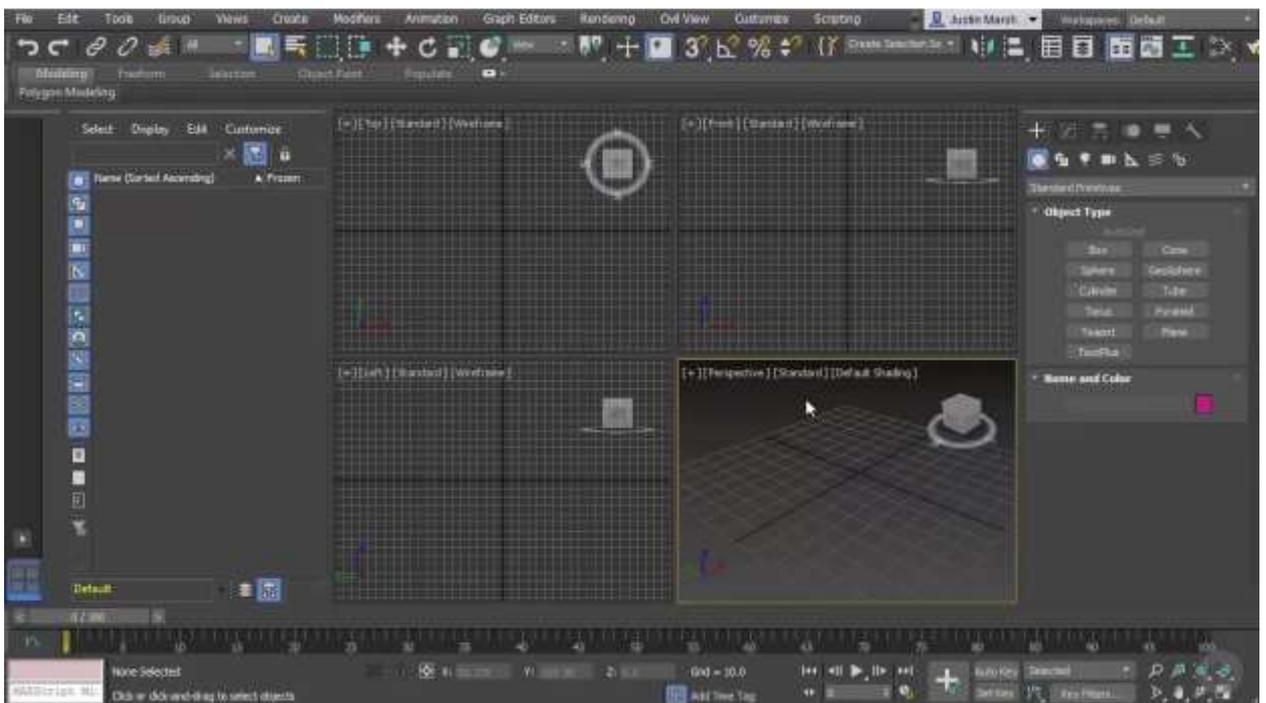
### Порівняння функціональних можливостей поширених редакторів тривимірної графіки

Зараз застосовуються різноманітні програми для 3D-моделювання. З кожним роком їх стає дедалі більше, оскільки виробники програмного забезпечення прагнуть задовольнити потреби широкого кола користувачів, тому при виникненні нових запитів вони випускають нові продукти. Зустрічаються як платні, так і безкоштовні програми для 3D-моделювання. Вивчимо детальніше найпопулярніші з них:

- Autodesk 3dsMax
- Maya
- AutoCad
- Cinema 4D
- Компас 3D
- Rhinoceros
- Blender
- Wings 3D
- Google SketchUp

**Autodesk 3dsMax** (раніше 3DStudio MAX) - професійне програмне забезпечення для 3D-моделювання, анімації та візуалізації при створенні ігор і проектуванні. Наразі розробляється і видається компанією Autodesk. Програма доступна за передплатою від одного місяця до трьох років для комерційних цілей. Для студентів і викладачів підписка на один рік безкоштовна, але з такою ліцензією програму можна використовувати тільки для навчання. 3ds Max - широко поширена програма, підходить для професійної роботи і надає значний функціонал. З її допомогою можна створювати анімацію і тривимірну графіку, є можливості для малювання складних моделей. Отримувані в результаті об'єкти детально опрацьовані і

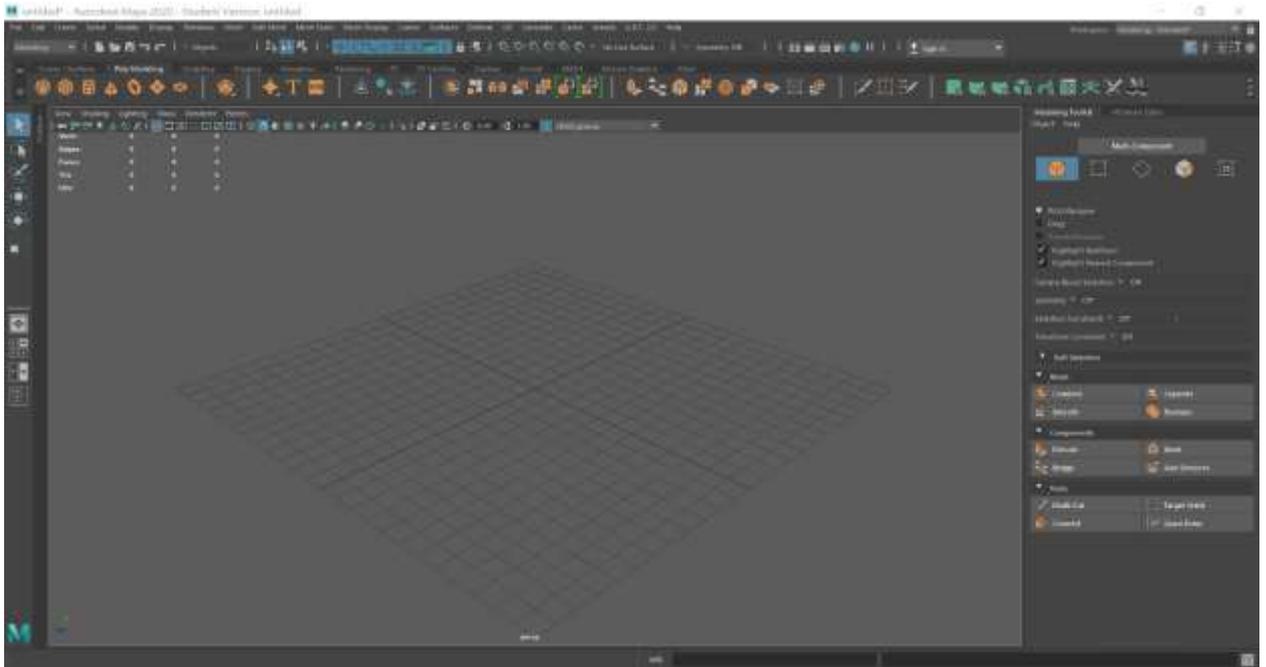
потім можуть анімуватися. Крім платної версії програми є безкоштовна версія для студентів. 3Ds Max пропонує користувачам різні способи та надійні інструменти для створення та редагування анімацій. Вони зможуть створювати тривимірні комп'ютерні анімації та ефекти, які можна застосовувати до комп'ютерних ігор, фільмів, трансляцій, медичних ілюстрацій або судово-медичних презентацій. Програмне забезпечення має кілька контролерів анімації, які використовуються для зберігання значень ключів і процедурних налаштувань, обробляючи все, що користувачі анімують з його допомогою. Крім того, 3Ds Max дає змогу пов'язувати об'єкти разом. У результаті чого можна формувати ієрархії або ланцюжки, за допомогою яких можна одночасно анімувати набори об'єктів, спрощуючи процес.



*Рис. А.1. Інтерфейс програми 3ds Max*

**Maya** - професійна програма, яку використовують під час зйомок фільмів і створення ігор. Вона має широкі можливості для створення реалістичних моделей високої якості. Autodesk Maya - редактор тривимірної графіки, доступний на Windows, macOS і Linux. Maya має широку функціональність 3D-анімації, моделювання та візуалізації. Програму використовують для створення анімації, середовищ, графіки руху,

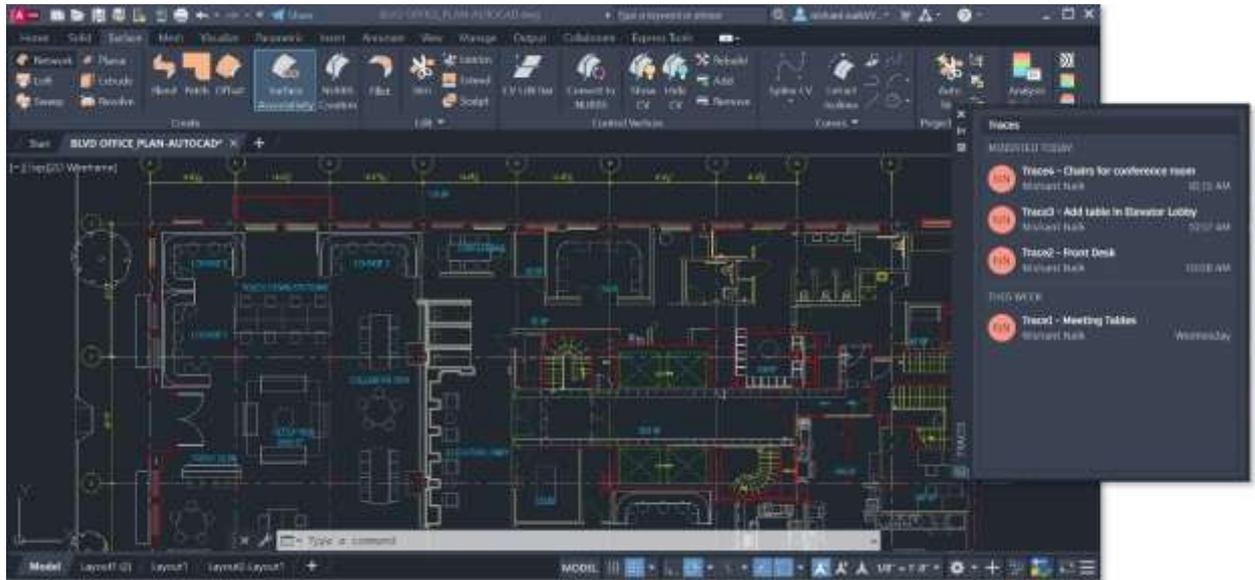
віртуальної реальності та персонажів. Широко застосовується в кінематографії, телебаченні та ігровій індустрії. Спочатку розроблено Alias Systems Corporation, а потім викуплено і підтримується в даний час Autodesk, Inc.



*Рис. А.2. Інтерфейс програми Maya*

**AutoCad** - інструмент для ефектного 2D і 3D-моделювання, має багатомовний інтерфейс, розібратися в якому можна навіть без спеціальних навичок. З його допомогою можна спочатку намалювати двомірну модель, а потім перевести її в тривимірне зображення. Також можна створювати окремі об'єкти і великі конструкції, а також ігрові текстури. AutoCAD - дво- і тривимірна система автоматизованого проектування і креслення, розроблена компанією Autodesk. Першу версію системи було випущено 1982 року. AutoCAD і спеціалізовані застосунки на його основі знайшли широке застосування в машинобудуванні, будівництві, архітектурі та інших галузях промисловості. AutoCAD містить повний набір інструментів для комплексного тривимірного моделювання (підтримується твердотільне, поверхневе і полігональне моделювання). AutoCAD дає змогу отримати високоякісну візуалізацію моделей за допомогою системи рендерингу mental ray. Також у програмі реалізовано управління тривимірним друком

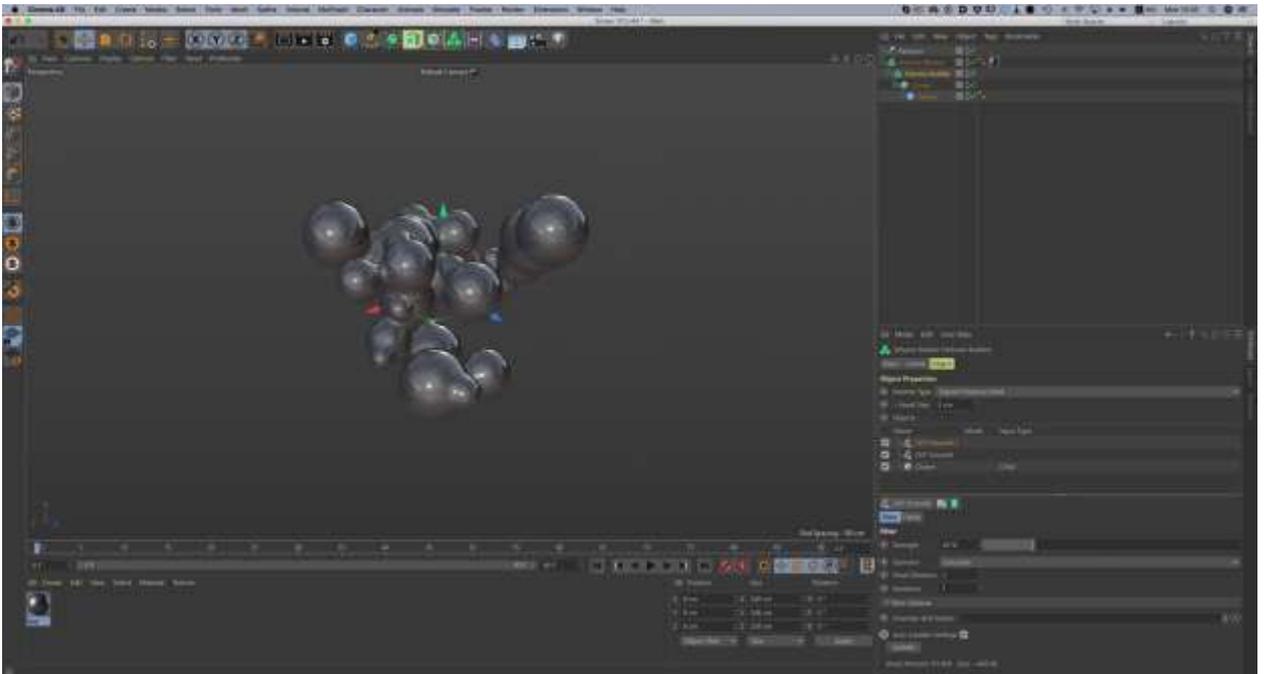
(результат моделювання можна відправити на 3D-принтер) і підтримка хмар точок (дає змогу працювати з результатами 3D-сканування).



*Рис. А.3. Інтерфейс програми AutoCad*

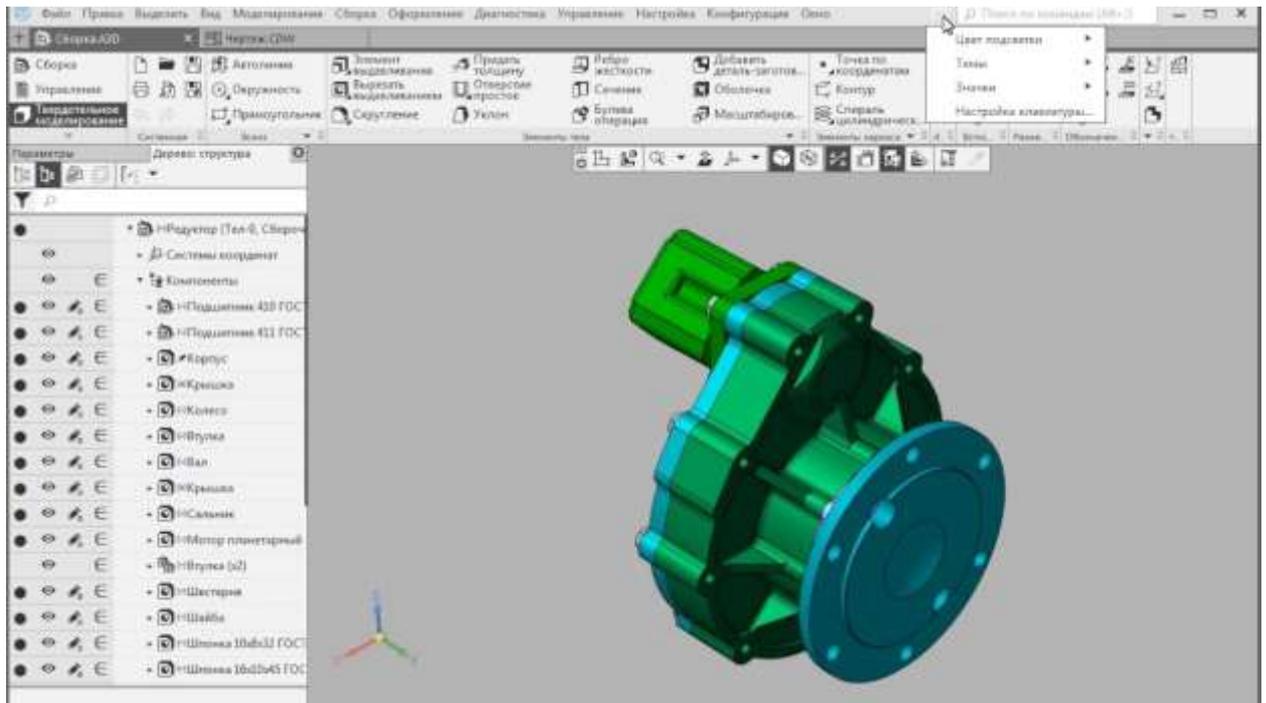
**Cinema 4D** - багатофункціональна програма для тривимірного моделювання та анімації. Відрізняється зрозумілим інтерфейсом. Починаючи з R21, доступна лише одна версія Cinema 4D. Вона замінює всі попередні варіанти, включаючи BodyPaint 3D, і включає в себе всі функції минулого варіанту "Studio". З R21 всі бінарні файли були уніфіковані. Немає ніякої технічної різниці між комерційною, навчальною або демонстраційною версіями. Різниця тепер тільки в ліцензуванні. У 2014 році вийшла версія Cinema 4D Lite, яка поставлялася в комплекті з Adobe After Effects Creative Cloud 2014. "Lite" виступає в якості ознайомчої версії, з багатьма функціями утриманими. Це частина партнерства між двома компаніями, де плагін виробництва Maxon під назвою Cineware дозволяє в будь-якому варіанті створити безперебійний робочий процес з After Effects. Варіант "Lite" залежить від After Effects CC, для його запуску потрібно, щоб остання була запущена, і продається тільки як компонент пакету, що входить до складу AE CS через Adobe. Спочатку Cinema 4D був розроблений для комп'ютерів Amiga на початку 1990-х років, і перші три версії програми були доступні виключно для цієї платформи. Однак, починаючи з версії 4, компанія Maxon почала розробляти програму і для комп'ютерів Windows та Macintosh,

мотивуючи це бажанням охопити ширшу аудиторію та зростаючою нестабільністю ринку Amiga після банкрутства компанії Commodore. На 2022 рік Cinema 4D є найпростішою у використанні програмою для 3D художників, принаймні, так заявлено на сайті розробників. Усі основні переваги також присутні: зручність у використанні, всіма улюблений інтуїтивний інтерфейс, стабільність (усі нові функції тестуються перед виходом), інтегрована довідка, процедурний робочий процес і доступність програми в різних варіантах залежно від цілей використання.



*Рис. А.1. Інтерфейс програми Сінета 4D*

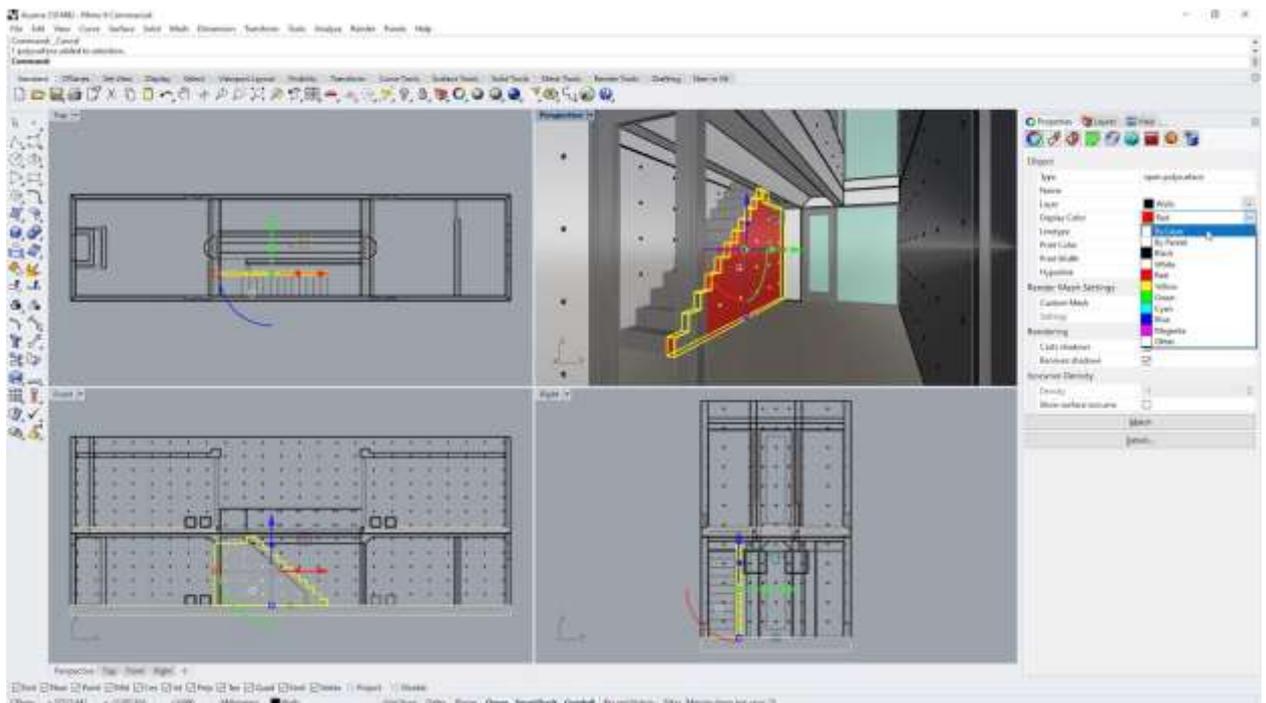
**Компас 3D** - програмне забезпечення для створення об'ємних моделей. Воно засноване на математичних розрахунках і є чудовим варіантом для здійснення інженерних проектів. Можливості програми включають не тільки побудову моделі, а й математичні розрахунки для подальшого її виготовлення.



*Рис. А.5. Интерфейс программы Компас 3Д*

**Rhino** - застосовується в архітектурі, кораблебудуванні, дизайні, а також мультимедійних технологіях. Набула поширення внаслідок об'ємного функціоналу та можливості імпорту й експорту різноманітних типів файлів. Rhino - комерційне програмне забезпечення для тривимірного NURBS-модельювання розробки Robert McNeel & Associates. Переважно використовується в промисловому дизайні, архітектурі, корабельному проектуванні, ювелірному та автомобільному дизайні, у CAD/CAM проектуванні, швидкому прототипуванні, реверсивному проектуванні, а також у мультимедіа та графічному дизайні. Rhino спеціалізується на NURBS-модельюванні. До плагінів, які розробляє McNeel, входять Flamingo (рейтрейс рендеринг), Penguin (нефотореалістичний рендеринг), Bongo (анімація) і Brazil (складний рендеринг). Існує понад 1000 сторонніх плагінів для Rhino. Як і в багатьох інших програмах моделювання, у Rhino є своя мова скриптів, заснована на Visual Basic, а SDK дає змогу читати і записувати файли безпосередньо. Розробка Rhino почалася в 1992 році як плагін для AutoCAD з ініціативи людини на ім'я Боб МакНіл (Bob McNeel) - бухгалтера за освітою і просто дуже захопленої людини, яка волею долі опинилася серед розробників програмного забезпечення спершу для

бухгалтерських компаній, а потім і пакетів 3D-графіки. Однак практично відразу після початку своєї діяльності як партнера компанії Autodesk, Боб МакНіл зібрав свою команду і, об'єднавшись у McNeel & Associates, переключився на створення самостійного застосунку для Windows. Формат файлу Rhino (.3DM) використовується для передачі геометрії NURBS. Розробники Rhino започаткували проєкт openNURBS, щоб надати розробникам графічного програмного забезпечення інструменти для точного передавання геометрії між додатками. Зростаюча популярність Rhino ґрунтується на його різноманітності, розмаїтті функціонального застосування, швидкому навчанні, відносно невеликій вартості та можливості імпорту/експорту майже 30 різних форматів, що дають змогу використовувати Rhino як "конвертер" у робочому процесі.



*Рис. А.6. Інтерфейс програми Rhinoceros*

**Wings 3D** - елементарна програма для тривимірного моделювання, що дає змогу працювати з простими моделями. Простий і нехитрий інтерфейс серйозно спрощує завдання фахівцям-початківцям. Також вихідний код програми відкритий для модифікації.

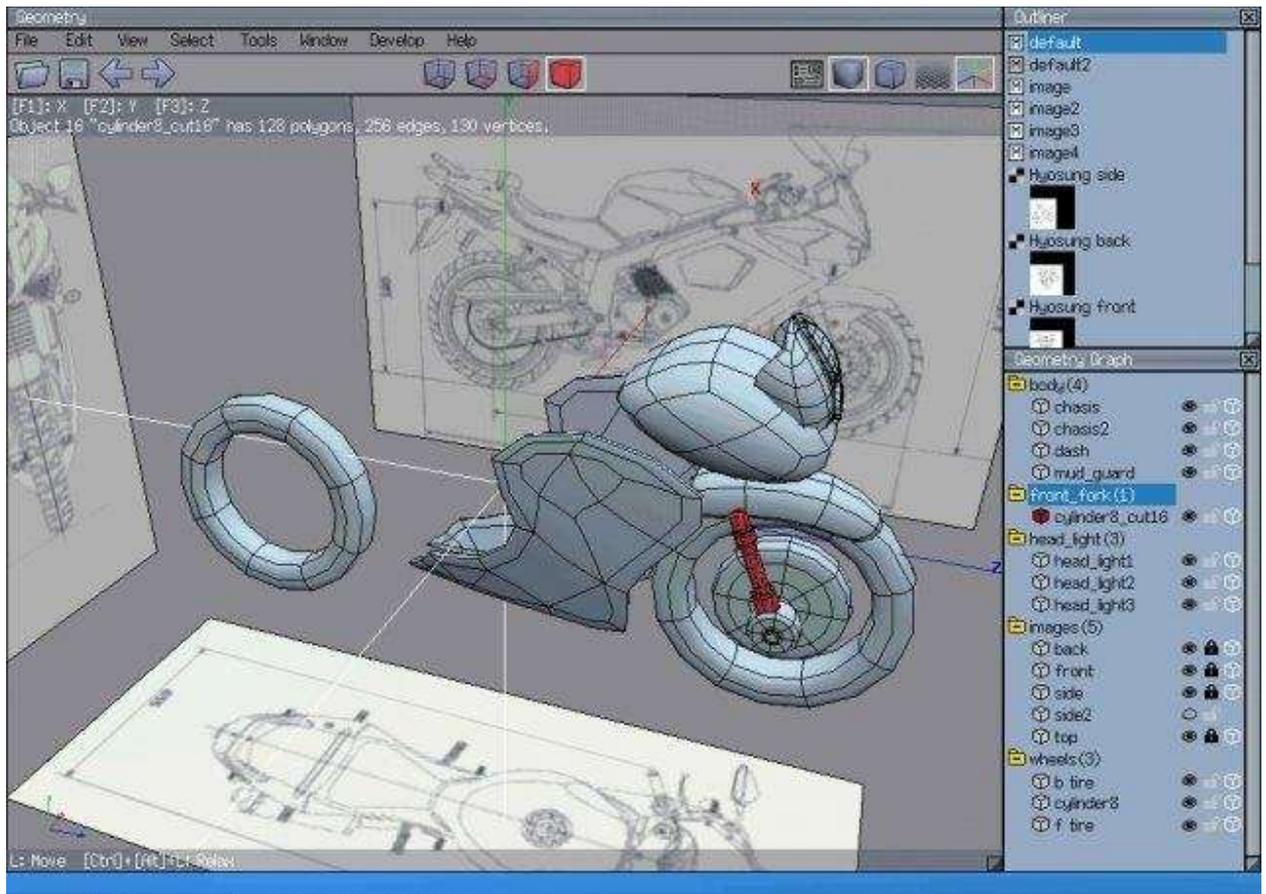


Рис. А.7. Інтерфейс програми Wings 3D

**Google SketchUp** - дає змогу створювати і змінювати безліч видів моделей, доповнюючи їх новими елементами і текстурами. Має великий функціонал для роботи над об'єктами різної складності.

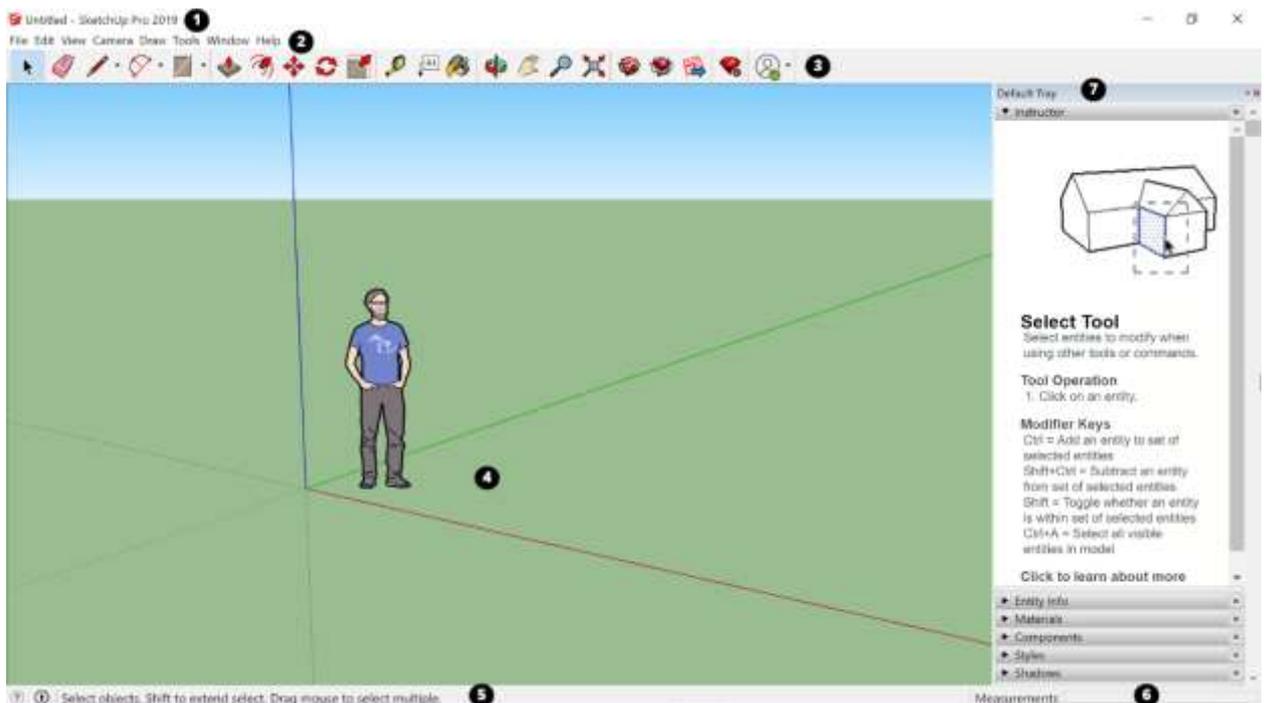


Рис. А.8. Інтерфейс програми Google SketchUP

**Blender** - програма, що дає змогу проводити рендеринг, анімацію, монтаж і подальшу обробку. Представлений функціонал можна розширити з використанням плагінів. Годиться для навчання основам 3D-моделювання. Blender - це вільний і відкритий набір програмних засобів тривимірної комп'ютерної графіки, що використовується для створення анімаційних фільмів, візуальних ефектів, мистецтва, 3D-друкованих моделей, анімаційної графіки, інтерактивних 3D-додатків, віртуальної реальності, а раніше - відеоігор. Можливості Blender включають 3D-моделювання, UV-мапінг, текстурювання, цифрове малювання, редагування растрової графіки, такелаж і скіннінг, моделювання рідини і диму, моделювання частинок, моделювання м'якого тіла, ліплення, анімацію, переміщення сірників, рендеринг, анімаційну графіку, відеомонтаж і композитинг. Blender підтримує різноманітні геометричні примітиви, включаючи полігональні сітки, криві Безьє, поверхні NURBS, метаболи, ікосфери, текст, а також систему моделювання n-кутників під назвою B-mesh. Існує також вдосконалена система полігонального моделювання, доступ до якої можна отримати через режим редагування. Вона підтримує такі функції, як видавлювання, скошування та розбиття.

Модифікатори застосовують неруйнівні ефекти, які можуть бути застосовані при рендерингу або експорті, наприклад, поверхні розбиття. Blender має мультироздільну цифрову скульптуру, яка включає в себе динамічну топологію, "запікання", ремешинг, ресиметрію та децимацію. Останнє використовується для спрощення моделей з метою експорту (прикладом є ігрові активи).

Blender має систему геометричних вузлів для процедурного та неруйнівного створення та маніпулювання геометрією. Вперше її було додано у Blender 2.92, який зосереджується на розсіюванні та інстансуванні об'єктів. Вона має форму модифікатора, тому її можна накладати на інші різні модифікатори. Система використовує атрибути об'єктів, які можна змінювати та перевизначати за допомогою рядкових входів. Атрибути можуть включати

позиції, нормалі та UV-карти. Всі атрибути можна переглянути в редакторі таблиць атрибутів. Утиліта Geometry Nodes також має можливість створення примітивних сіток. У Blender 3.0 до Geometry Nodes була додана підтримка створення та модифікації об'єктів кривих. У Blender 3.0 робочий процес Geometry Nodes був повністю перероблений за допомогою полів, щоб зробити систему більш інтуїтивно зрозумілою та працювати подібно до вузлів шейдерів. Блендер можна використовувати для імітації диму, дощу, пилу, тканини, рідин, волосся та твердих тіл. Можливості анімації на основі ключових кадрів у Blender'і включають зворотну кінематику, арматуру, гаки, деформації на основі кривих і решітки, ключі форм, нелінійну анімацію, обмеження і зважування вершин. Крім того, інструменти Grease Pencil дозволяють виконувати 2D анімацію в рамках повного 3D конвеєра.

Внутрішній рушій рендерингу з рендерингом ліній розгортки, непрямим освітленням та оклюзією навколишнього середовища, який може експортуватися в широкий спектр форматів; рушій рендерингу трасування шляху під назвою Cycles, який може використовувати переваги графічного процесора для рендерингу. Cycles підтримує Open Shading Language, починаючи з версії Blender 2.65. Гібридний рендеринг Cycles можливий у версії 2.92 з Optix. Тайли розраховуються за допомогою GPU у поєднанні з центральним процесором. EEVEE - це новий фізично заснований візуалізатор реального часу. Він працює як візуалізатор фінальних кадрів, а також як рушій, що керує вікном перегляду Blender'a у реальному часі для створення ресурсів. Blender дозволяє створювати процедурні та вузлові текстури, а також текстурний, проєктивний, вершинний, ваговий та динамічний малюнок.

Blender має композитор на основі вузлів у конвеєрі рендерингу, який прискорюється за допомогою OpenCL. Він також включає нелінійний редактор відео, який називається Video Sequence Editor (VSE), з підтримкою таких ефектів, як розмиття за Гаусом, градація кольорів, переходи вицвітання

та стирання та інші перетворення відео. Однак, у VSE немає вбудованої багатоядерної підтримки для рендерингу відео.

Blender підтримує написання сценаріїв на Python для створення користувацьких інструментів, прототипування, ігрової логіки, імпорту/експорту з інших форматів та автоматизації завдань. Це дозволяє інтегруватися з декількома зовнішніми рушіями візуалізації за допомогою плагінів/аддонів.

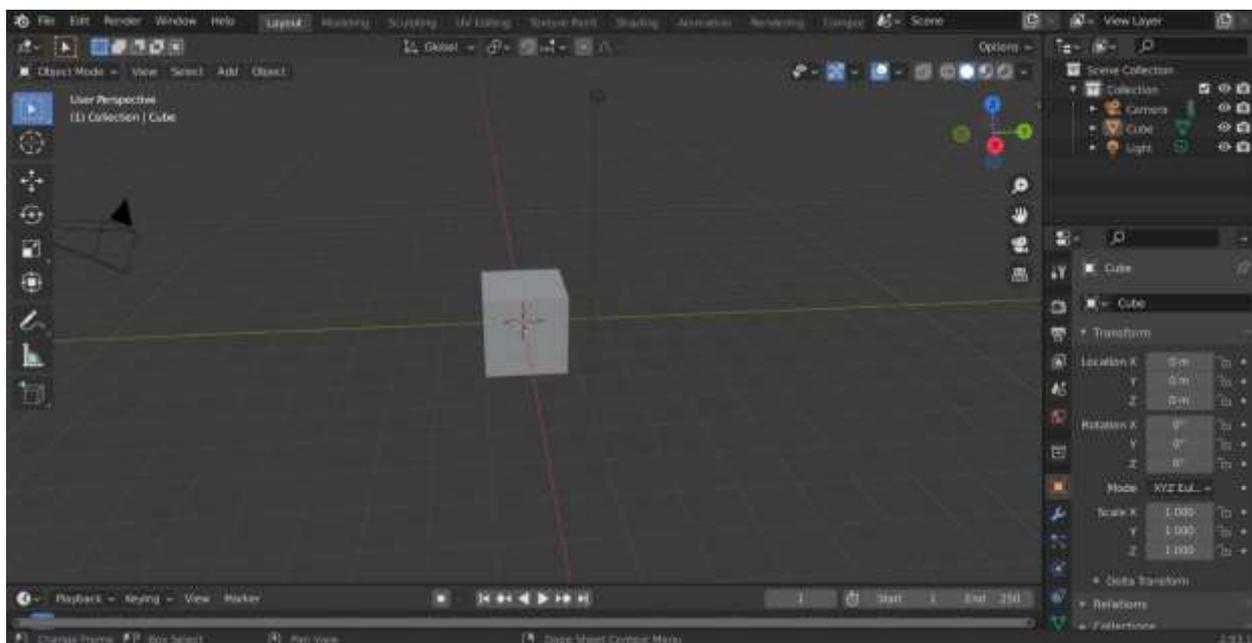
Програма підтримує різноманітні формати 3D файлів для імпорту та експорту, серед яких Alembic, 3D Studio (3DS), FBX, DXF, SVG, STL (для 3D друку), UDIM, USD, VRML, WebM, X3D та OBJ.

Blender Game Engine був вбудованим графічним і логічним рушієм реального часу з такими функціями, як виявлення зіткнень, рушій динаміки і програмована логіка. Він також дозволяв створювати автономні застосунки в реальному часі, починаючи від архітектурної візуалізації і закінчуючи відеоіграми. У квітні 2018 року рушій було вилучено з майбутньої серії випусків Blender 2.8, через оновлення та доопрацювання рушія, що відставали від інших ігрових рушіїв, таких як Godot з відкритим вихідним кодом та Unity. В анонсах 2.8 команда Blender'a спеціально згадала рушій Godot як підходящу заміну для мігруючих користувачів Blender Game Engine. Blender має багато режимів для роботи з об'єктами, основними з яких є Режим Об'єкта (Object Mode) та Режим Редагування (Edit Mode), які перемикаються за допомогою клавіші Tab. Режим Object (Об'єкт) використовується для маніпулювання окремими об'єктами як єдиним цілим, в той час як режим Edit (Редагування) використовується для маніпулювання фактичними даними об'єкта. Наприклад, режим об'єкта можна використовувати для переміщення, масштабування і обертання цілих полігональних сіток, а режим редагування - для маніпулювання окремими вершинами однієї сітки. Існує також кілька інших режимів, таких як Vertex Paint, Weight Paint і Sculpt Mode.

Графічний інтерфейс Blender'a будує свою плиткову віконну систему поверх одного або декількох вікон, наданих базовою платформою. Одне вікно платформи (часто розміром на весь екран) поділяється на секції та підрозділи, які можуть мати будь-який тип вигляду Блендера або тип вікна. Користувач може визначити декілька макетів таких вікон Blender'a, які називаються екранами, і швидко перемикається між ними, вибираючи з меню або за допомогою комбінацій клавіш. Власними елементами графічного інтерфейсу кожного типу вікна можна керувати за допомогою тих же інструментів, що і 3D-виглядом. Наприклад, аналогічними елементами керування можна збільшувати та зменшувати масштаб GUI-кнопок, збільшувати та зменшувати масштаб у 3D-вікні. Видовий екран графічного інтерфейсу та компонування екрану повністю налаштовуються користувачем. Можна налаштувати інтерфейс для конкретних завдань, таких як редагування відео, UV-мапінг або текстуровання, приховуючи функції, які не використовуються для виконання завдання.

Cycles підтримує рендеринг на GPU, який використовується для прискорення часу рендерингу. Існує три режими рендеринга на GPU: CUDA, який є кращим методом для старих відеокарт Nvidia; OptiX, який використовує апаратні можливості трасування променів архітектури Turing та архітектури Ampere від Nvidia; та OpenCL, який підтримує рендеринг на відеокартах AMD Radeon (з доданою підтримкою Intel Iris та Xe у версії 2.92). Інструментальне програмне забезпечення, пов'язане з цими режимами візуалізації, не входить до складу Blender'a і має бути окремо встановлене та налаштоване згідно з відповідними інструкціями з їхніх джерел. Також підтримується декілька графічних процесорів, які можуть бути використані для створення ферми рендерингу, що має декілька графічних процесорів, однак не збільшує доступну пам'ять, оскільки кожен графічний процесор може отримати доступ лише до власної пам'яті. Починаючи з версії 2.90, це обмеження SLI-карт порушено за допомогою NVlink від Nvidia. Apple's Metal API отримав початкову реалізацію в blender 3.1 для комп'ютерів Apple з

чіпами M1 та відеокартами AMD. Бенчмарк Blender 3.2 показує великі переваги давно підтримуваної CUDA проти новішого API OptiX та абсолютно нового HIP для апаратного забезпечення AMD. Деякі покращення продуктивності будуть доступні для HIP та OptiX у Blender 3.3 та 3.4. У Blender 3.3 LTS з'явилася підтримка графічних процесорів Intel Arc.



*Рис. А.9. Інтерфейс програми Blender*

## Додаток Б.

### Визначення застосунку для створення тривимірних моделей та рушія для реалізації редактора

Важливо зробити два конкретні вибори, а саме обрати редактор для створення та редагування тривимірних моделей та, що більш важливіше – обрати рушій для реалізації застосунку. Власне, вибір редактора стоїть серед наступних редакторів:

- Autodesk 3dsMax
- Maya
- AutoCad
- Cinema 4D
- Компас 3D
- Rhinoceros
- Blender
- Wings 3D
- Google SketchUp

**Autodesk 3dsMax** – чудовий, провірний часом редактор, але є декілька важливих зауважень. Можна перелічувати мінуси інтерфейсу, структури і архітектури якою він побудований, це достатньо стара програма, яка бере початок з 1996-го року і яка надбудовувалась два десятиліття, стаючи важчою і зберігаючи недоліки від версії до версії. Останні роки на програму виходили великі, глобальні оновлення що намагаються покращити ситуацію і осучаснити редактор як за інтерфейсом так і за інструментарієм, але принципи роботи все ті самі. Це в будь-якому випадку не погана програма, але коли є чудова можливість почати вивчення тривимірної графіки, можливо це не найкращий вибір. Що ще більш важливо, 3ds Max – дорогий. Це досить дорогий редактор, звісно, є можливість оформити ліцензію для

студентів, але це потребує часу і певної “волокіти”, без якої цілком можливо обійтись.

**Maya** – стандарт індустрії, з яким працюють майже всі студії у світі. Архітектура програми зручна для побудови робочого процесу, завдяки просунутому API (від англ. application programming interface - набір бібліотек, які відповідають різним функціям Maya). Свого часу це дало змогу ще більше зміцнити свої позиції в індустрії. Сьогодні видавець, Autodesk, а до цього - Alias, постачає редактор у навчальні заклади, тож більшість спеціалістів вчать робити анімацію саме в цій програмі. Монополізація поряд із гнучкістю функціоналу та централізованою підтримкою - те, що робить Maya популярною серед гігантів медіа-індустрії. Програму використовували у фільмах "Аватар", "У пошуках Немо", франшизах "Трансформери" і "Льодовиковий період", "Холодне серце" та багатьох інших. Але я не розглядаю програму для створення персонажної анімації, для створення графіки у кіно, а при цьому програма достатньо складна в опануванні і вивченні, в неї своєрідний інтерфейс для якого надто складно звикати як для того базового набору інструментів що мені потрібен для проекту. Крім цього – як і з 3ds Max, знову полягає питання ліцензування.

**AutoCad** - програма була створена для автоматизованого проектування і креслення. Перша версія програми була випущена в далекому 1982 р. Використовується в промислових галузях: архітектура, машинобудування тощо. Один із головних мінусів програми autocad. Складність у прив'язці інформаційних даних до об'єктів, що викликає незручності в роботі. Виходячи з того, що програма достатньо не молода, багато елементів програми вже не актуальні на даний період часу. В цьому проекті в мене немає потреби створення моделей за кресленнями, тому такий робочий процес тільки надто ускладнить роботу, взагалі не відповідаючи потребам. Вивчати такий складний професійний редактор і займатись питанням його ліцензування це складнощі які хотілось би оминати.

**Cinema 4D** – чудовий редактор, більш близький за своєю зручністю і простотою до того що я шукаю як ідеальний редактор для роботи. Але одразу постає ряд проблем: Тільки платне поширення. Досить високі системні вимоги. Неможливість завантажувати модулі та плагіни окремо в міру потреби - надаються тільки пакетно. Значне навантаження на апаратні ресурси ПК. Ця програма зараз – основний інструмент для створення рухомою графіки у рекламі, але вже декілька років вона точно втрачає свої позиції на ланці редактора для створення тривимірних моделей.

**Компас 3D** – вже стала стандартом для інженерів підприємств промислового виробництва, завдяки простому інтерфейсу та широким можливостям моделювання для інженерів. Її часто використовують архітектори та будівельники для розроблення креслень будівель і металоконструкцій. Здебільшого, орієнтована на промислове виробництво різноманітних приладів, пристроїв та апаратів. По тим самим причинам що і AutoCad, цей інструмент не підходить для виконання моїх задач.

**Rhinoceros** – ще одна програма де процес тривимірного моделювання зведено в створення креслень. Геометрію для креслень доводиться малювати руками. Ні, формально ви, звичайно, можете покласти потрібну проекцію деталі на креслення. Але ви тут же натрапите на те, що як краї деталі він сприймає не самі краї деталі, а краї всіх поверхонь, що входять до неї. Які можуть бути дуже кривими, якщо ви перед тим робили яке-небудь обрізання країв, що проходить по вже існуючій поверхні. Програма станом на сьогодні вже дуже застара і неактуальна, вже навіть не піднімаючи питання ліцензування та ціни.

**Wings 3D** - програма працює на основі відкритого вихідного коду і застосовується для 3D моделювання. У ній користувачі можуть працювати з різними нескладними моделями. Завдяки мінімальному і доступному інтерфейсу робота фахівців-початківців значно полегшується, а за допомогою відкритого коду програма за необхідності модифікується. І хоч значні недоліки за які я відмітав варіанти до цього – відсутні, є інший не

менш критичний для мене – програма застара, занадто, не відповідає сучасним запитам від тривимірного редактора, а також, через стару архітектуру – може працювати нестабільно.

**Google SketchUp** – короткий безкоштовний період та в цілому, знову-ж, це робота з CAD моделями в яких немає жодної необхідності і це лише загальмую робочий процес.

**Blender 3D** - безкоштовний програмний продукт, призначений для створення і редагування тривимірної графіки. Програма поширена на всіх популярних платформах, має відкритий вихідний код і доступна абсолютно безплатно всім охочим. Blender – є однією з найпопулярніших програм для 3D моделювання вважається Blender. Це багатофункціональний редактор, знайомство з яким буде корисним для тих, хто тільки вступає у світ тривимірної графіки і прагне зрозуміти основні принципи роботи цієї галузі. Blender зможе познайомити з основними особливостями 3D моделювання, а також запропонує використовувати зрозумілі інструменти для створення або редагування моделей.

Представлені практично всі опції, доступні користувачам професійних збірок. Люди з певним досвідом роботи можуть простежити вплив одразу кількох різних програм, від яких Blender взяв для себе потроху. Однак при цьому додаток не втратив своєї самостійності та індивідуальності. Додаток Blender прийнято позиціонувати як багатофункціональний інструмент для роботи з тривимірною графікою, анімацією або навіть створення комп'ютерних ігор. Розглядаючи подібний функціонал, користувач явно очікує серйозних вимог до комп'ютерних ресурсів. Однак дистрибутив важить всього 70 Мбайт, що здається абсолютно неможливим в умовах повноцінного функціонування всіх інструментів. Але всі елементи пакета дійсно працюють повною мірою з відмінною продуктивністю. При такому маленькому розмірі програма може дати все необхідне у сфері тривимірної графіки:

- 3D моделювання. Користувачеві доступна величезна кількість інструментів для створення і редагування 3D моделей найрізноманітніших рівнів складності. Причому моделювати об'єкти можна за допомогою доступних примітивів, полігонів, NURBS-кривих і кривих Безьє. Додатково передбачено функціонал для формування метасфер і керування формою за допомогою булевих операцій. Не варто забувати і про технології Subdivision Surface і найбільш зрозумілі інструменти створення скульптур. За аналогією з професійними збірками, тут передбачені модифікатори для зміни форми моделей.
- Створення анімації. Незважаючи на те, що редактор сам по собі розрахований на моделювання, анімація представлена теж непогано. Можна використовувати традиційну скелетну анімацію або риггінг, інверсну кінематику, різні обмежувачі та багато іншого. Усі коефіцієнти і параметри налаштовуються за допомогою вбудованих інструментів. Додатково представлено динаміку тіл різної твердості та формування анімації дрібних частинок.
- Текстури. Можна накладати відразу кілька текстур на один і той самий об'єкт. Є маса інструментів для текстурування, зокрема UV-маппінг і функція часткового налаштування. Значно полегшує роботу налаштування шейдерів.
- Малювання. Є багато вбудованих засобів для створення начерків пензлями прямо у вікні програми. Зараз цю функцію використовують для більш зручного формування двомірної анімації.
- Інструменти візуалізації. Є відразу кілька попередньо встановлених засобів для показу результату роботи, а також передбачена сумісність зі сторонніми рендерами від різних розробників.
- Відеоредактор. Про наявність цієї функції можуть не здогадатися навіть дуже досвідчені користувачі. Однак у програмі дійсно передбачено редактор відеороликів з непоганим інструментарієм.

Сама по собі програма Blender розрахована на тих, хто ще тільки починає свій шлях у світі об'ємного моделювання. Школярам вона дає змогу розвивати просторове мислення, а також подарує можливість створювати красиві листівки, відеоролики або навіть повноцінні 3D моделі. Можливо, представлений інструментарій матиме вплив і допоможе визначитися з майбутньою професією. Трохи досвідченіші користувачі можуть використовувати програму як хобі або навіть як джерело заробітку. Можливостей програми цілком достатньо для того, щоб успішно займатися розробкою дизайнів сайтів, логотипів, рекламних роликів або будь-яких банерів. Програма Blender вважається однією з найпростіших і найдоступніших в освоєнні. Саме з цієї причини її найчастіше рекомендують новачкам і тим, хто збирається займатися 3D моделюванням не в комерційних цілях але зараз це і достатньо популярний інструмент в ніші тривимірного моделювання і люди професійно працюють у Блендер цілими студіями. З моменту своєї появи і до сьогоднішнього дня програма Blender неодноразово оновлювалася і доповнювалася новими функціями. У неї з'являлося все більше шанувальників, які вкрай високо оцінювали представлений інструментарій.

Серед переваг програми найчастіше виділяють:

- Швидкість. Програма запускається набагато швидше за своїх конкурентів і до того ж практично миттєво реагує на всі команди навіть на не дуже потужних системах. Саме невимогливість до апаратної частини комп'ютера є визначальним фактором для вибору на користь Blender. Використовувати будь-який більш просунутий редактор на слабких комп'ютерах не вийде.
- Функціональність. На відміну від великої кількості аналогів, Blender заздалегідь містить усі необхідні інструменти для вирішення найрізноманітніших завдань. Тут можна робити практично все, що завгодно, включно зі створенням тривимірних

об'єктів, накладенням текстур, налаштуванням шейдингу, рендерингом і композитингом. У низці інших програм подібної функціональності можна домогтися тільки встановленням великої кількості додаткових плагінів, тобто сторонніх інструментів.

- Універсальність. За допомогою інструментів Blender користувач може створювати об'єкти для ігор, мультфільмів або кінострічок. Навіть передбачено роботу з анімаціями, яка, втім, перебуває на не дуже високому рівні.
- Швидкий внутрішній рендерер Cycles. Ця підсистема функціонує як на GPU, так і на CPU. Розробники Blender впровадили цю можливість задовго до того, як вона стала визнаним стандартом.
- Рушій EEVEE, що дає змогу в режимі реального часу переглядати всі текстури та накладені ефекти.
- Просунутий UI. У додатку Blender цей компонент дійсно приємніший і зручніший, тоді як інтерфейс більшості конкурентів має як мінімум застарілий вигляд. Ця проблема пояснюється тим, що розробники старих систем просто не хочуть нічого змінювати, щоб не змушувати користувачів звикати до нових елементів керування.

Не обійшлося і без недоліків, серед яких найчастіше називають:

- Інструменти можуть бути не такими потужними, як того вимагає робочий процес. Цей недолік вимагає деякого уточнення: деякі інструменти не працюють належним чином тільки з тієї причини, що користувач не знайомий з їх правильним застосуванням. На ділі виявляється, що велику кількість вбудованих функцій можна без проблем налаштувати під свої потреби.
- Часті оновлення. Регулярне доопрацювання і зміни, безсумнівно, підвищують актуальність програми. Однак вони ж постійно змінюють інтерфейс або вводять нові, не до кінця зрозумілі функції. У результаті

користувачам доводиться щоразу заново звикати до системи. До того ж регулярне поповнення функціоналу відбивається на кількості багів і недоробок, які виявляються вже в процесі використання.

З врахуванням всього, мій вибір як тривимірного редактору випав саме на Blender, крім того, саме з в цьому редакторі я маю вже досить розвинуті, широкі знання та навички роботи.

Платформою для реалізації редактору я обрав рушій Unreal Engine, головними перевагами для мене стала повна безкоштовність і вільне поширення, величезна кількість навчальних матеріалів, інтегрована мова візуального програмування BluePrint та використання мови програмування C++, також перевагою стала наявність певного минулого досвіду роботи з рушієм.

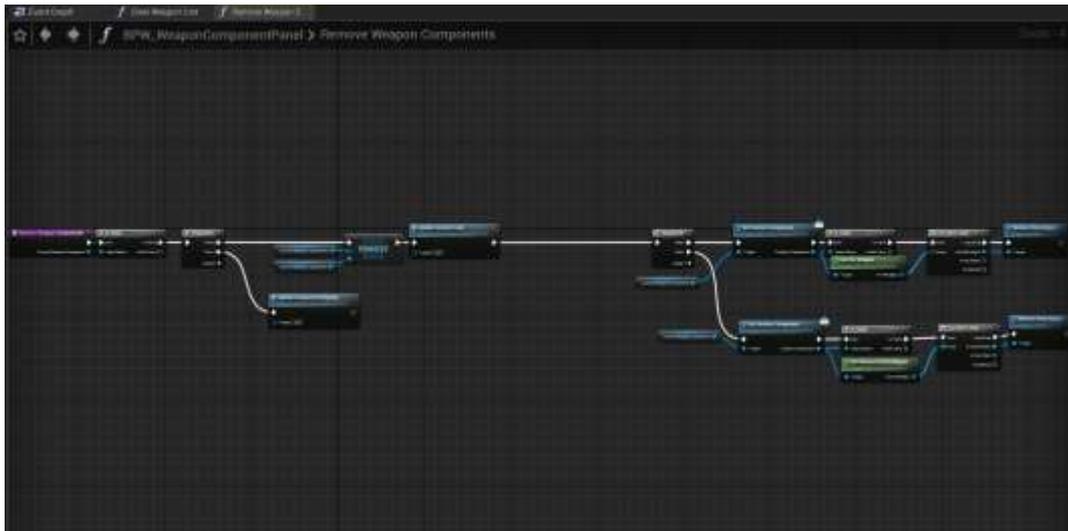
**Додаток В.**  
**Приклади створених елементів інтерфейсу та фрагменти програмного коду до них**



*Рис. В.1. Дизайн головного меню програми в редакторі "Unreal Engine"*

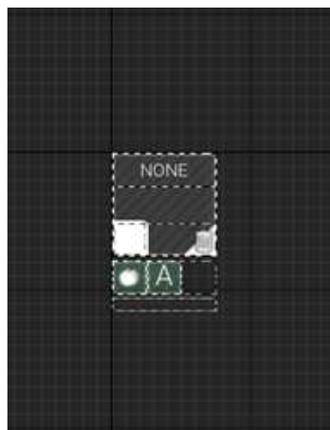


*Рис. В.2. Дизайн основного вікна програми в редакторі "Unreal Engine"*

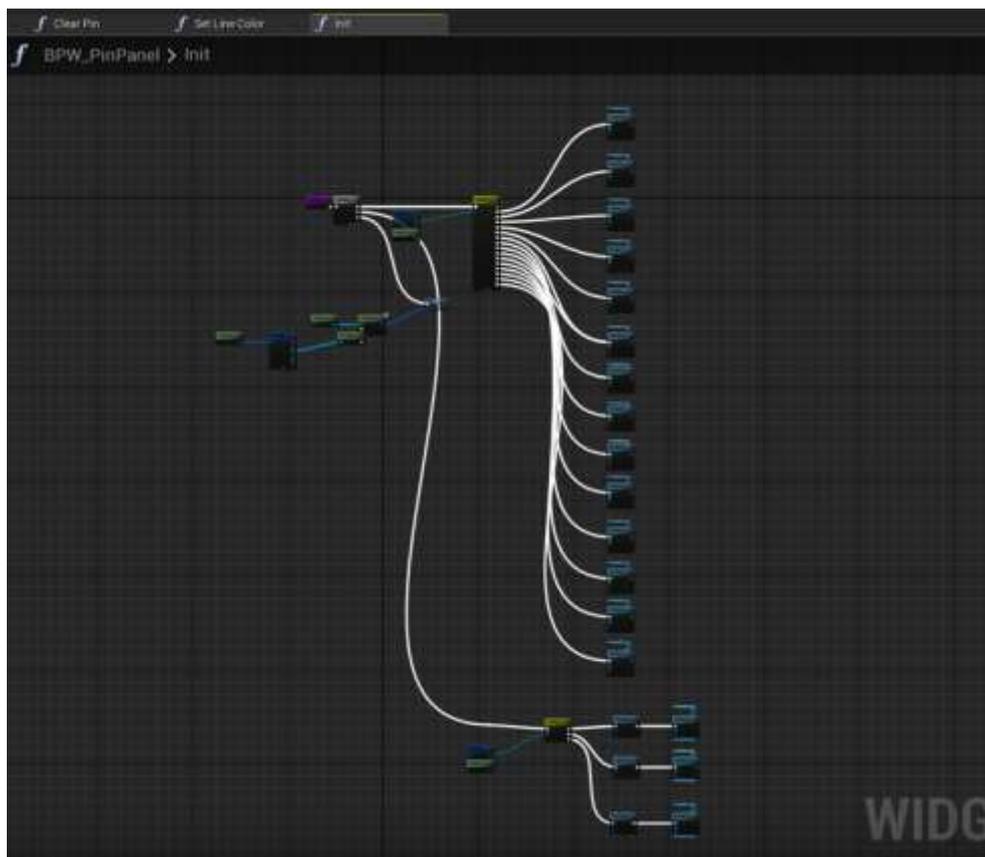


*Рис. В.3. Огляд програмного коду інтерфейсу основного вікна мовою "Blue Print" в редакторі "Unreal Engine"*

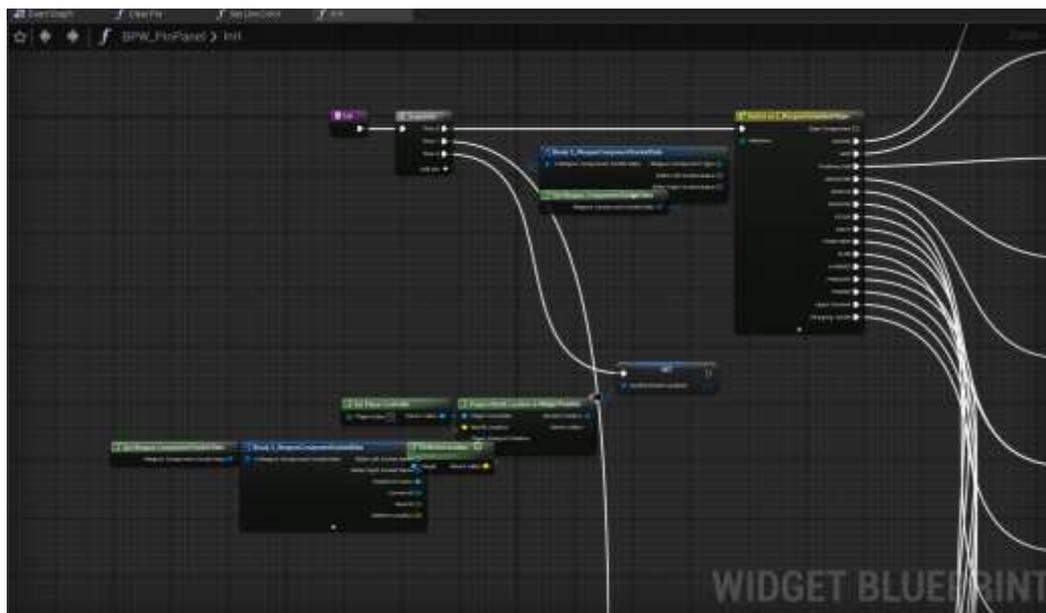
Комірка вибору компоненту для генерування зброї.



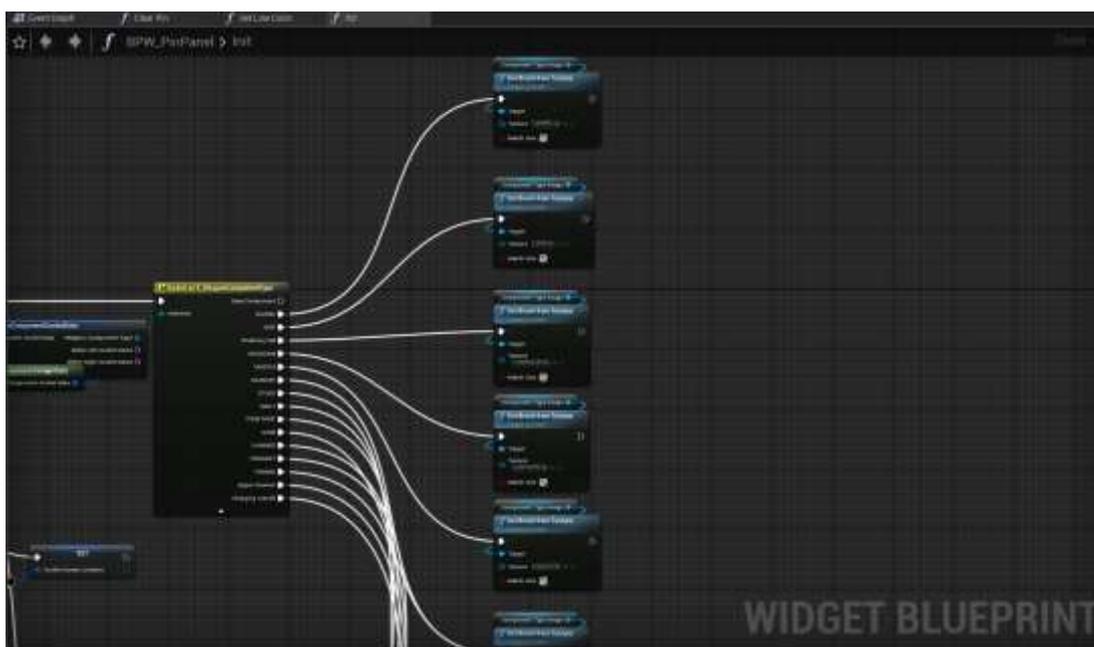
*Рис. В.4. Дизайн комірки вибору компоненту сцени в редакторі "Unreal Engine"*



*Рис. В.5. Програмний код комірки вибору компоненту сцени мовою "Blue Print" в редакторі "Unreal Engine"*

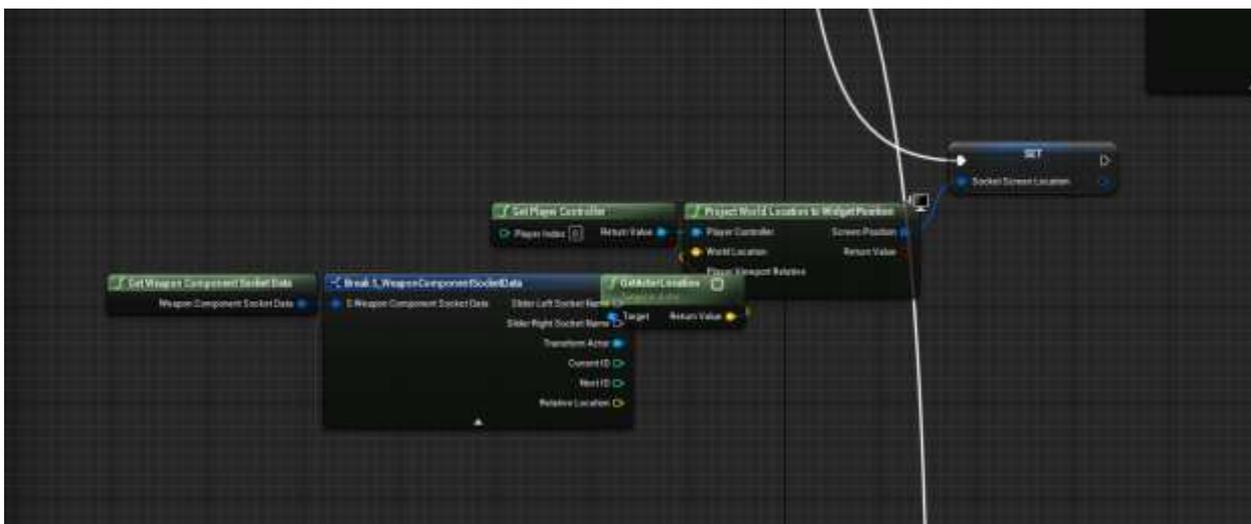


*Рис. В.6. Програмний код комірки вибору компоненту сцени мовою "Blue Print" в редакторі "Unreal Engine"*

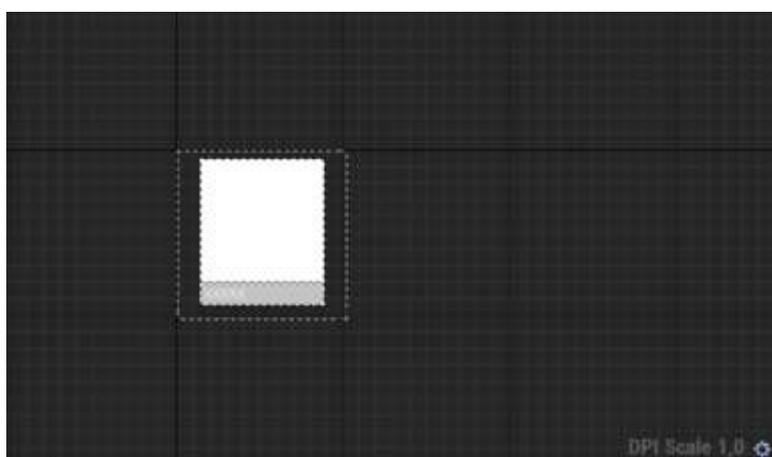


*Рис. В.7. Програмний код комірки вибору компоненту сцени мовою "Blue Print" в редакторі "Unreal Engine"*

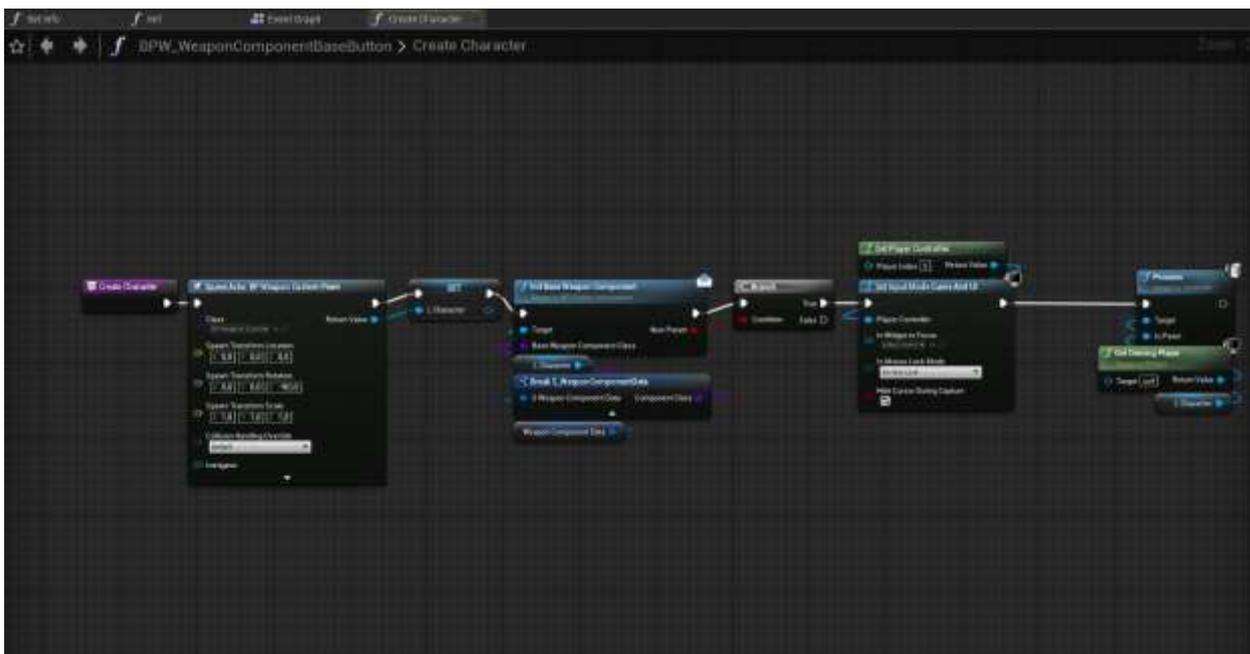
Встановлюється початкова позиція інтерфейсу, відповідного до комірки вибору компонентів зброї.



*Рис. В.8. Програмний код комірки вибору компоненту сцени мовою "Blue Print" в редакторі "Unreal Engine"*



*Рис. В.9. Дизайн порожньої комірки в редакторі "Unreal Engine"*

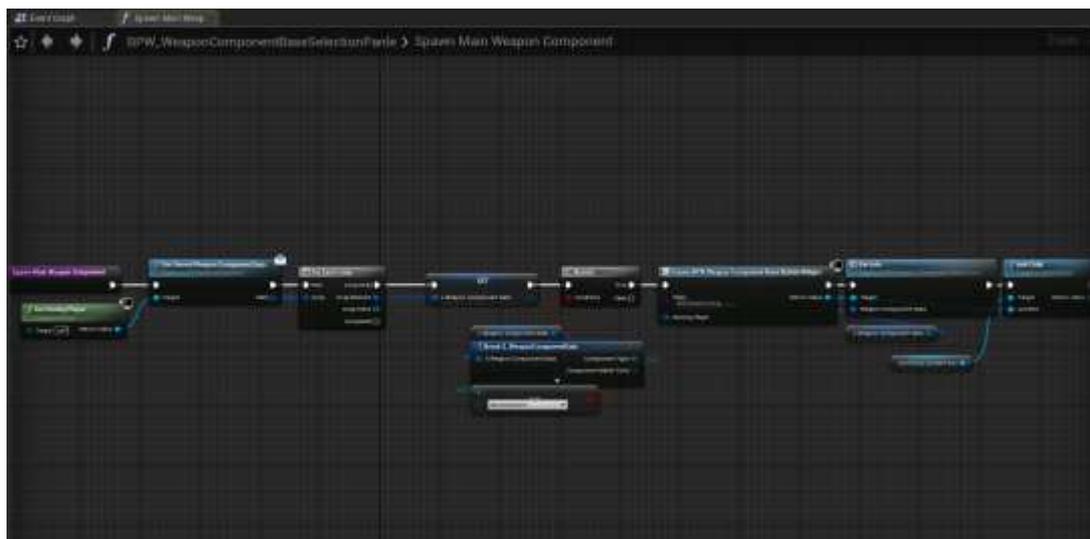


*Рис. В.10. Програмний код положення комірки вибору компоненту сцени мовою "Blue Print" в редакторі "Unreal Engine"*



*Рис. В.11. Панель вибору компонентів*

Вибір всіх наявних компонентів з таблиці а потім фільтрація невідповідних.

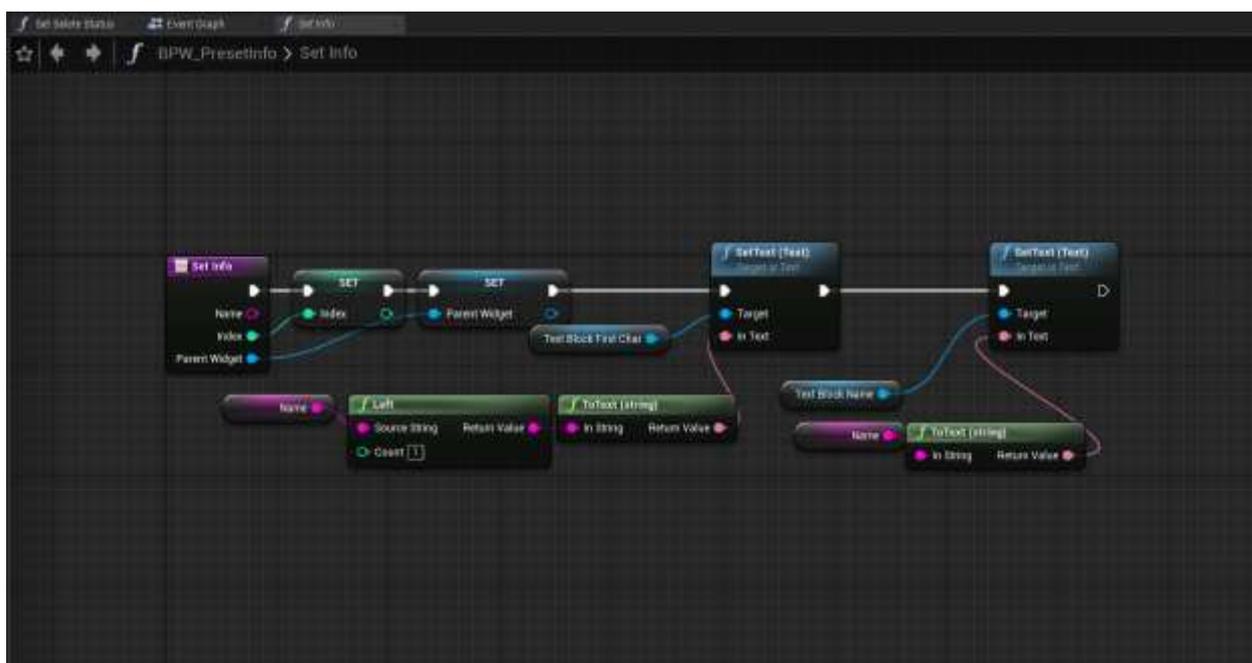


*Рис. В.12. Програмний код панелі вибору компонентів мовою "Blue Print" в редакторі "Unreal Engine"*

Контекстна інформаційна панель з інформацією про компонент зброї.

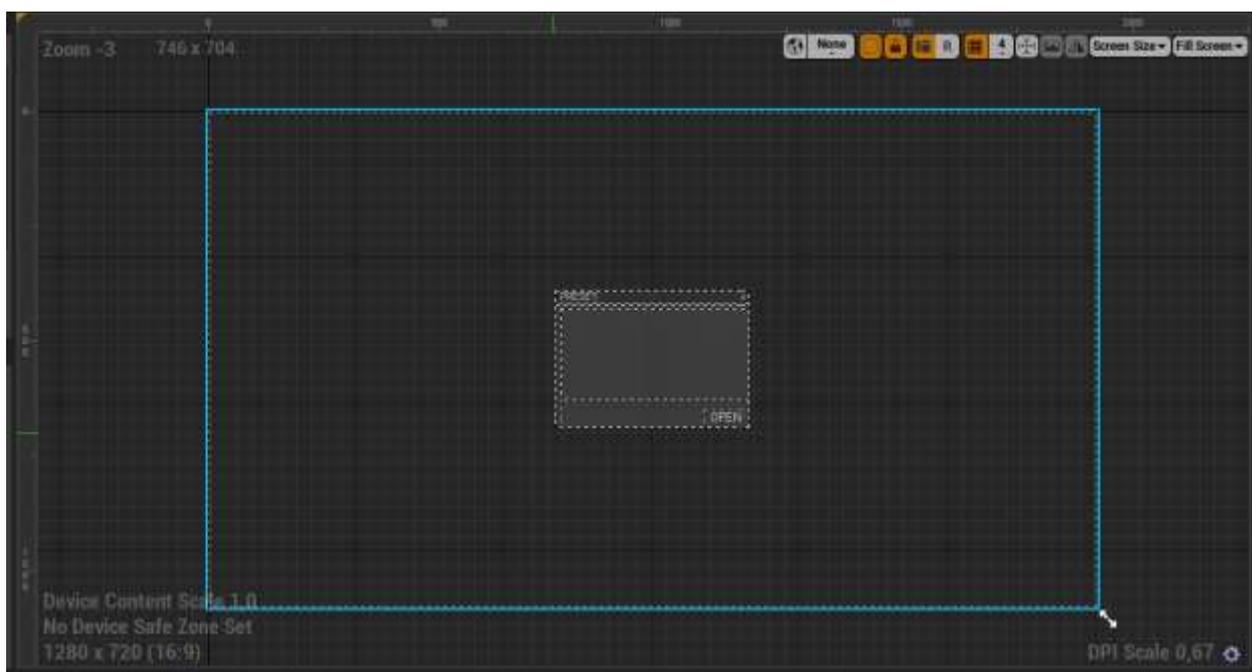


*Рис. В.13. Інтерфейс збереження готової сцени*

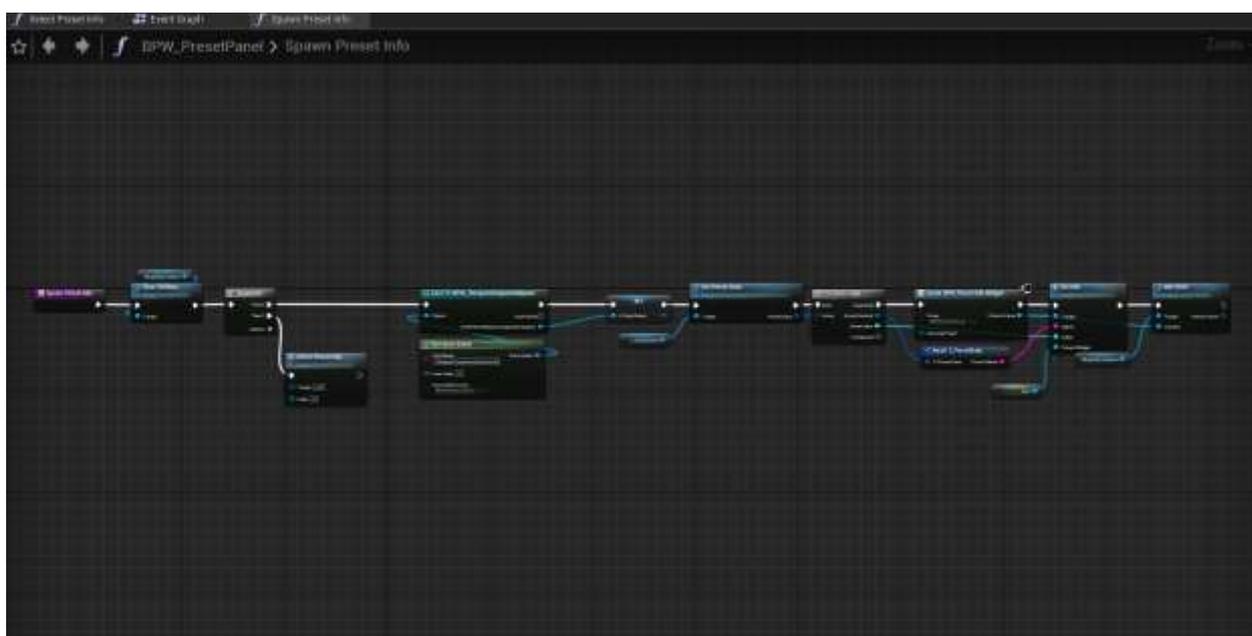


*Рис. В.14. Програмний код панелі збереження сцени мовою "Blue Print" в редакторі "Unreal Engine"*

Панель завантаження готового набору згенерованої сцени.

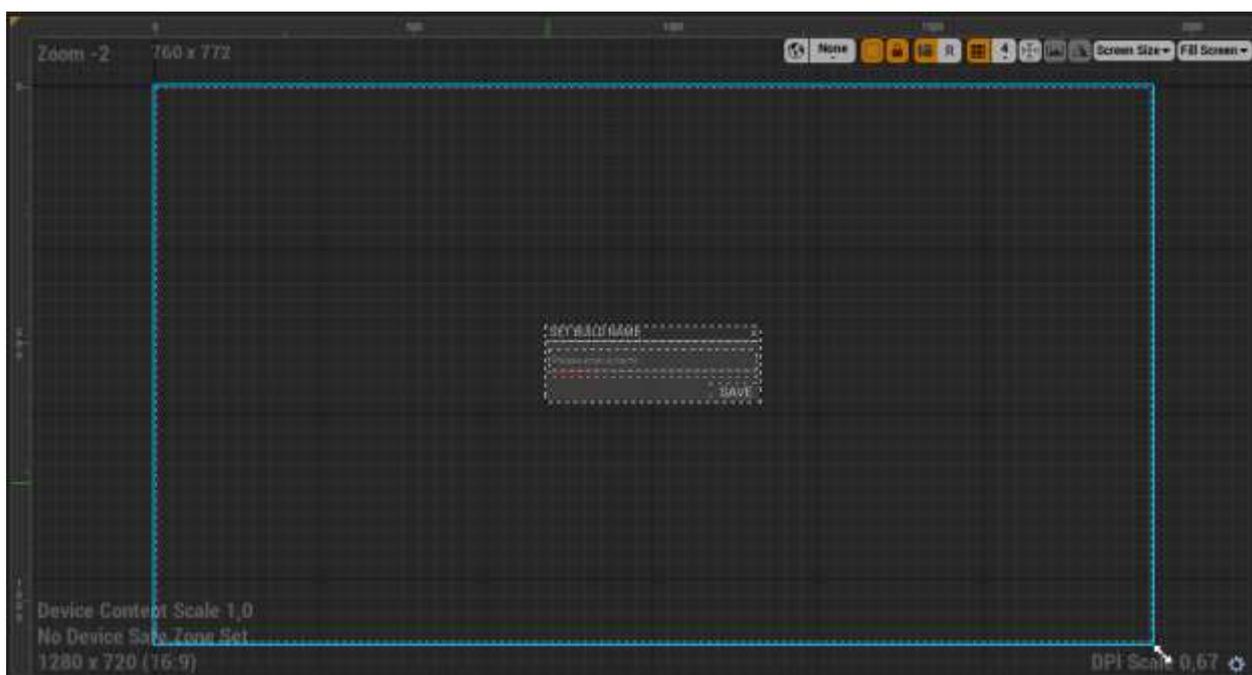


*Рис. В.15. Панель вибору збереженої сцени*

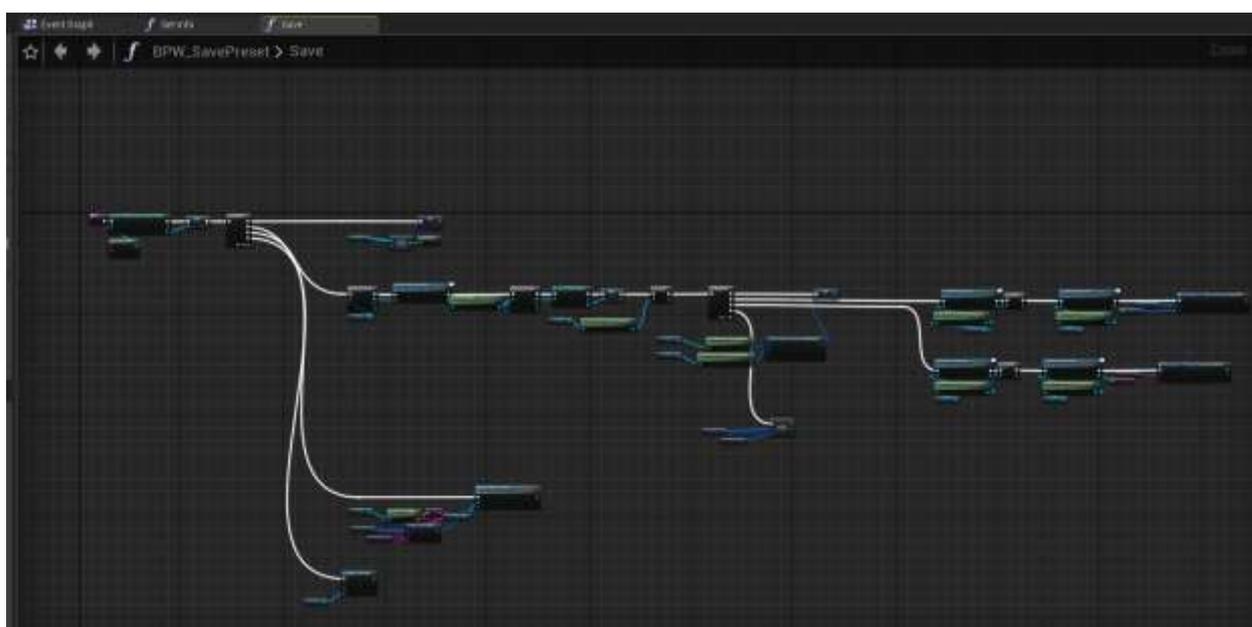


*Рис. В.16. Програмний код панелі вибору збереженої сцени мовою "Blue Print" в редакторі "Unreal Engine"*

Панель створення готового набору згенерованої сцени.



*Рис. В.17. Панель збереження готової сцени*



*Рис. В.18. Програмний код панелі збереження готової сцени мовою "Blue Print" в редакторі "Unreal Engine"*

Задається базовий клас, індекс базового класу повинен бути рівний 0, тому що він додається першим.

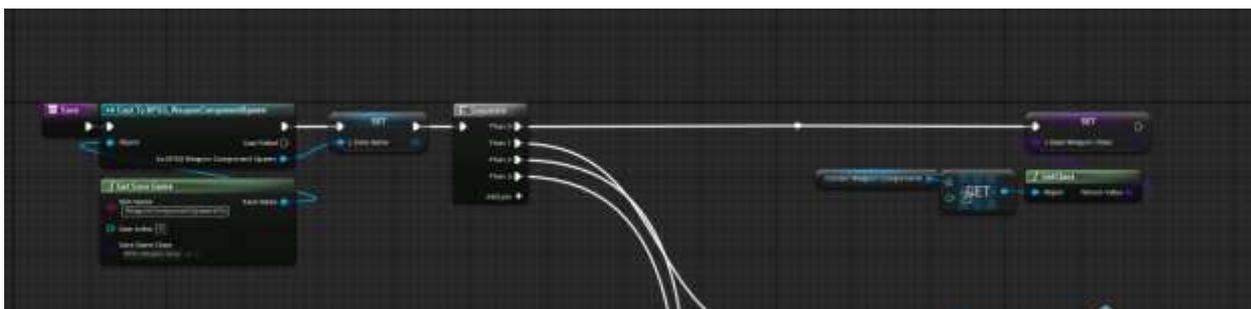


Рис. В.19. Програмний код панелі збереження готової сцени мовою "Blue Print" в редакторі "Unreal Engine"

Зберігання всієї інформації у SaveGame.

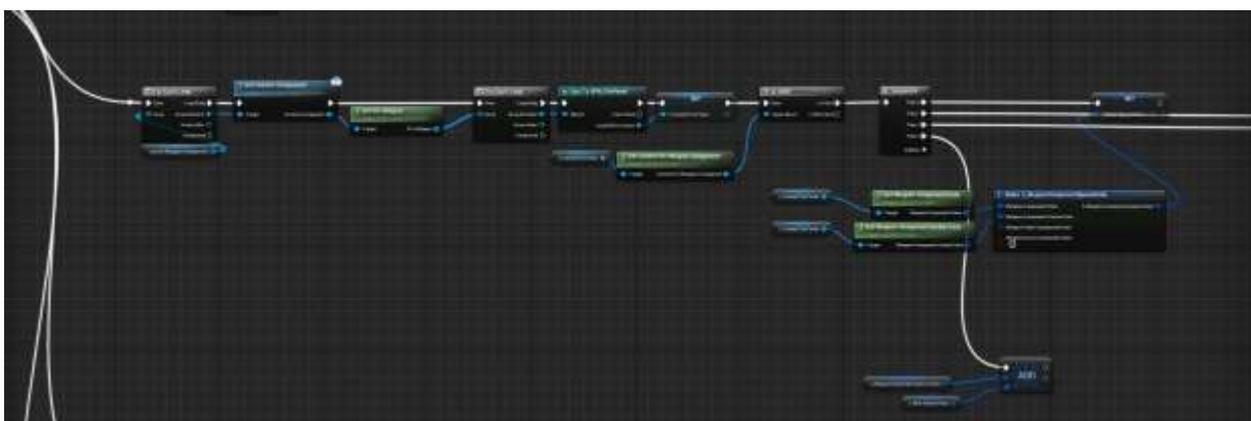


Рис. В.20. Програмний код панелі збереження готової сцени мовою "Blue Print" в редакторі "Unreal Engine"

Створення нових даних.

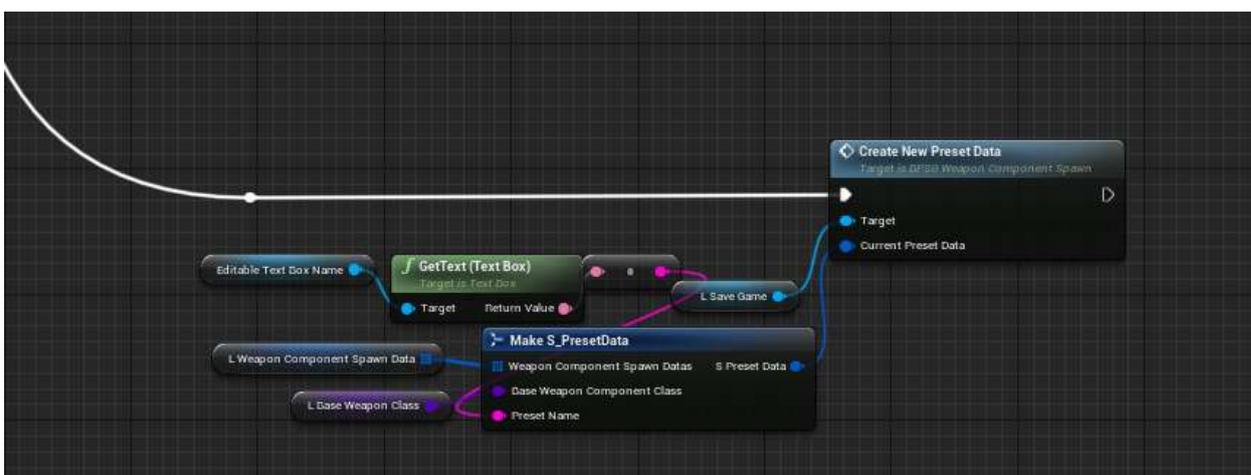
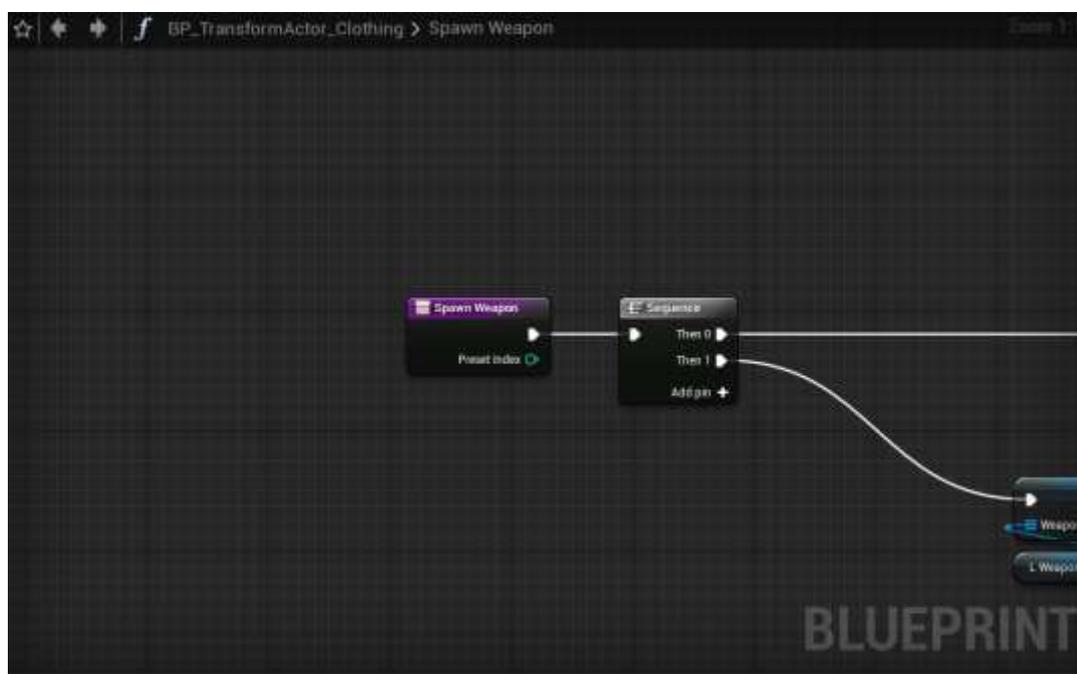


Рис. В.21. Програмний код панелі збереження готової сцени мовою "Blue Print" в редакторі "Unreal Engine"

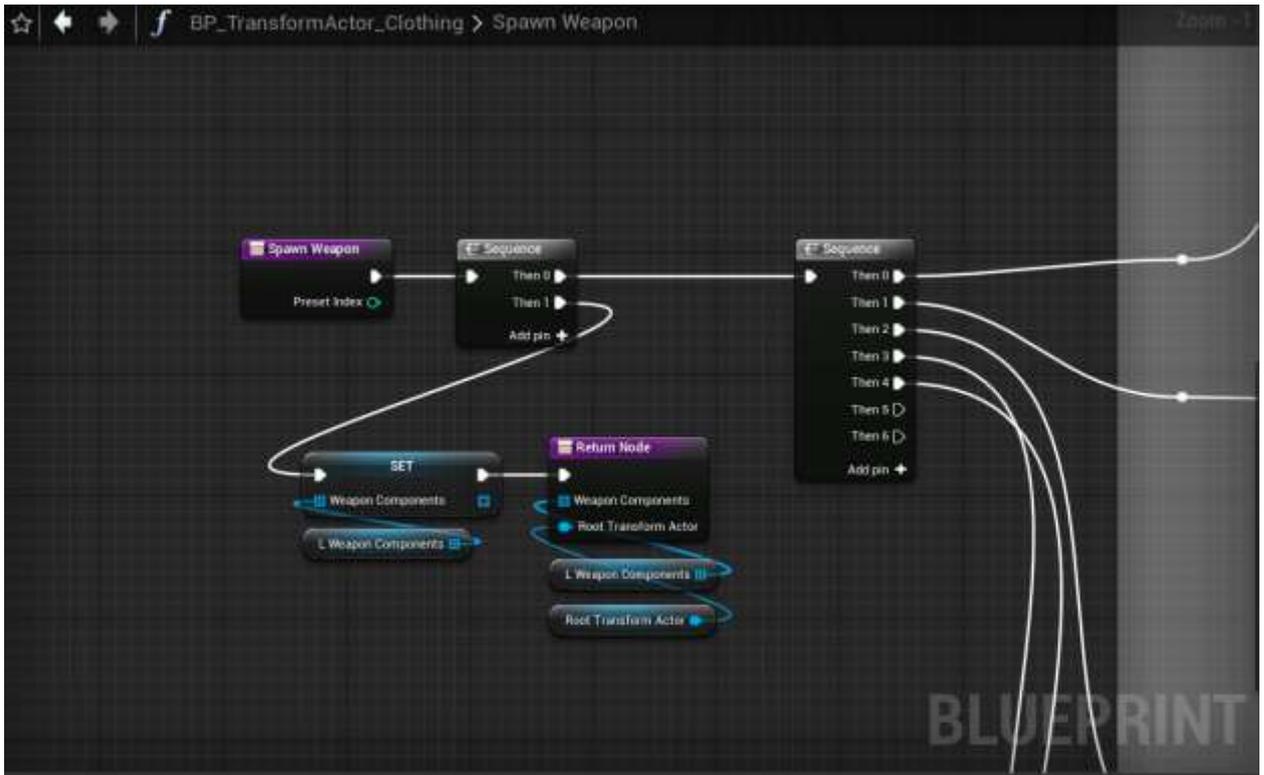
## Додаток Г.

### Опис реалізації функціоналу програми мовою програмування Blueprints



*Рис. Г.1. Огляд програмного коду генерування сцени мовою візуального програмування BluePrint*

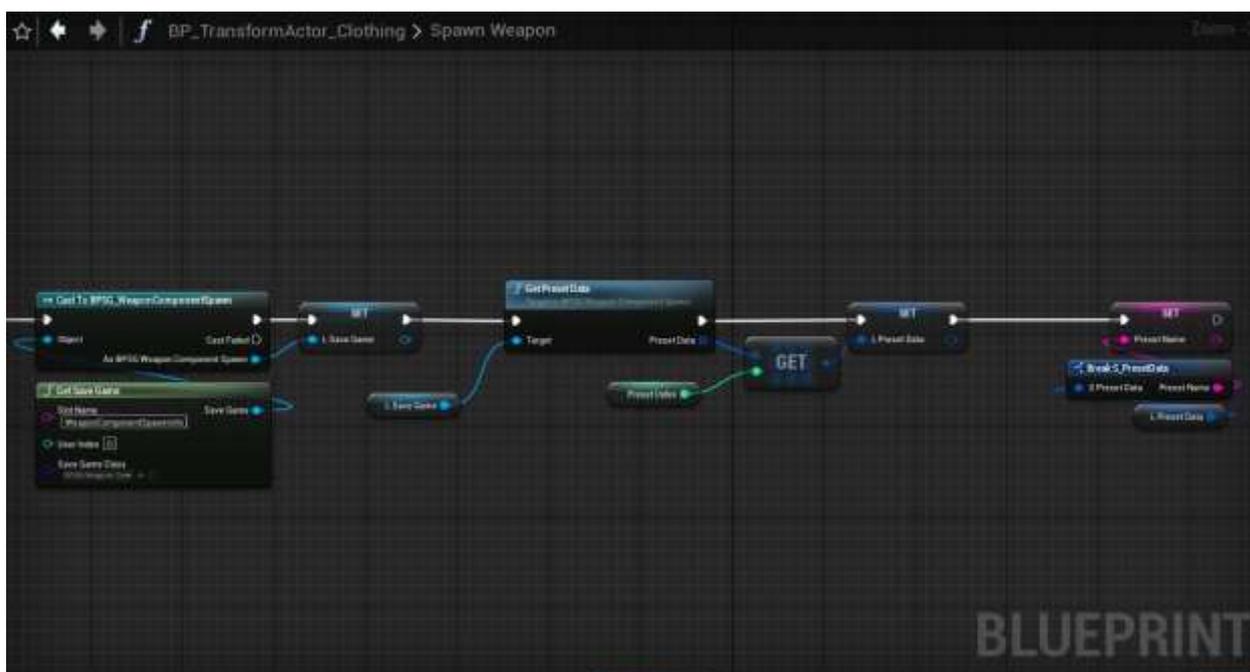
“Spawn Weapon” – викликає базовий компонент зброї, сам клас, оголошує що таке – зброя. Sequence (послідовність) - під час виклику функції вона послідовно викликає сигнали на виходах цієї функції. Ці виклики завжди виконуються в порядку їхнього позначення 1, потім 2, потім 3 і так далі. Цей блок має такі вузли: Input Pins (вхід) Запускає виконання послідовності дій; Then # Виходи, що запускають інші дії в потрібній послідовності; Add pin Додає точку виходу. Для додавання до блоку вузла потрібно натиснути кнопку Add pin, для видалення вузла потрібно викликати з контекстного меню.



*Рис. Г.2.. Огляд програмного коду генерування сцени мовою візуального програмування BluePrint*

В першу чергу (Then 0), виконання слідує до наступної черги (Sequence), робота якої буде детально описана далі. У другу чергу (Then 1) слідує до збору всіх існуючих компонентів, їх виклик та прихід у клас. Вузол “SET” встановлює значення змінної “WeaponComponents”, котра, в свою чергу, є масивом посилань на об'єкти-актори. Вузол “WeaponComponents” що підключено, зчитує значення змінної. Вузол “Return Node” завершує виконання функцій. Повертає вихідні параметри.

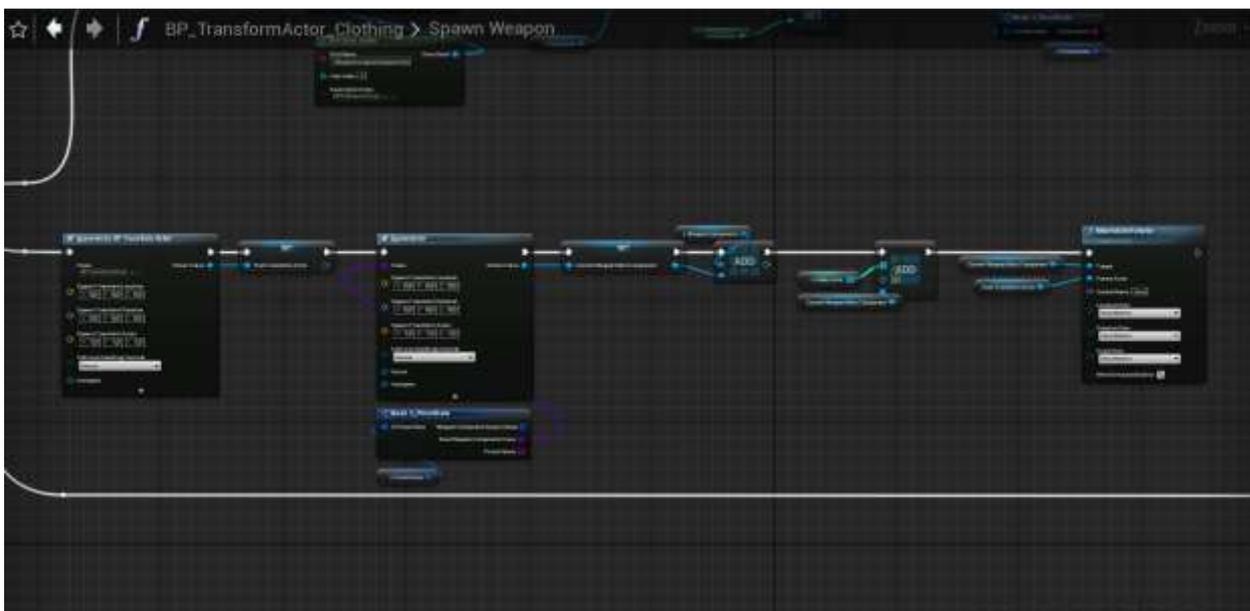
Перший крок (Then 0) наступної черги (Sequence) викликає генератор зброї (меча), тобто перший, базовий компонент, та звертається до класу “Save Game”.



*Рис. Г.3. Огляд програмного коду генерування сцени мовою візуального програмування Blueprint*

“Cast to BPSG\_WeaponComponentSpawn” - посилається на об'єкт породження, “GET” в свою чергу бере об'єкт породження та за заданим масивом та індексом повертає елемент масиву за цим індексом (оскільки це пряме посилання, то можна оперувати безпосередньо з елементом, а зміни, внесені в ін, будуть відображені в масиві).

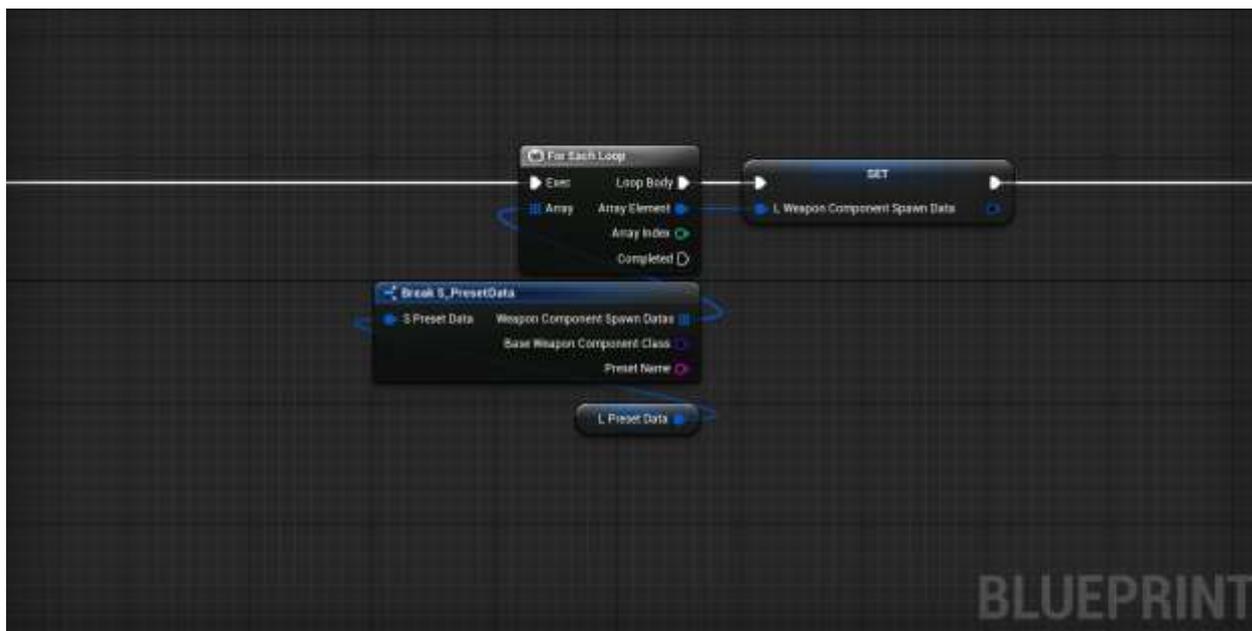
Другий крок (Then 1) встановлює сокети для вставки компонентів, тобто місця на яких і буде поява моделей, фрагментів меча.



*Рис. Г.4. Огляд програмного коду генерування сцени мовою візуального програмування Blueprint*

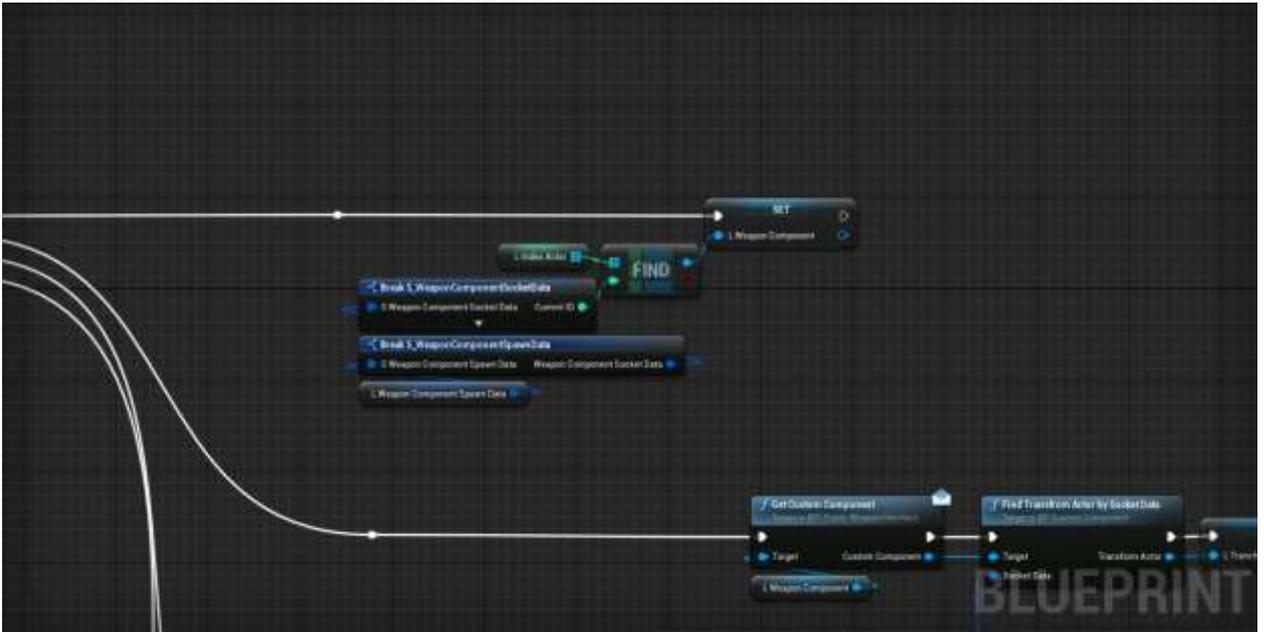
SpawnActor BP Transform Actor – породжують новий “Actor” за вказаною позицією. Перший вузол “ADD” просто додає елемент у масив, але наступний додає ключ та значення до карти. якщо щось вже використовує наданий ключ, він буде перезаписаний новим значенням. Після виклику Key гарантовано асоціюється зі значенням Value до наступної мутації карти. “AttachActorToActor” - приєднує RootComponent цього Actor до Actor, що постачається, за бажанням в іменованому гнізді, по суті, відповідає за появу нових кріплень після додання елементи в існуюче кріплення.

Третій крок (Then 2) відповідає безпосередньо за збір самої зброї. Він веде в нову чергу, але перед цим йде виклик циклу що збирає всі доступні компоненти для цієї зброї, це необхідно на випадок якщо якісь компоненти недопустимі для кріплення до інших.



*Рис. Г.5. Огляд програмного коду генерування сцени мовою візуального програмування BluePrint*

Далі йде новий “Sequence” що по черзі виконую наступні дії. Перший крок - WeaponComponent знаходиться за CurrentID, який є WeaponComponent, що буде приєднаний наступним. Це викликає сам компонент зброї, зброї як клас, і інформацію про всі його сокети, яку взято з попереднього кроку. Також тут викликається сокет для наступного компоненту.



*Рис. Г.6. Огляд програмного коду генерування сцени мовою візуального програмування BluePrint*

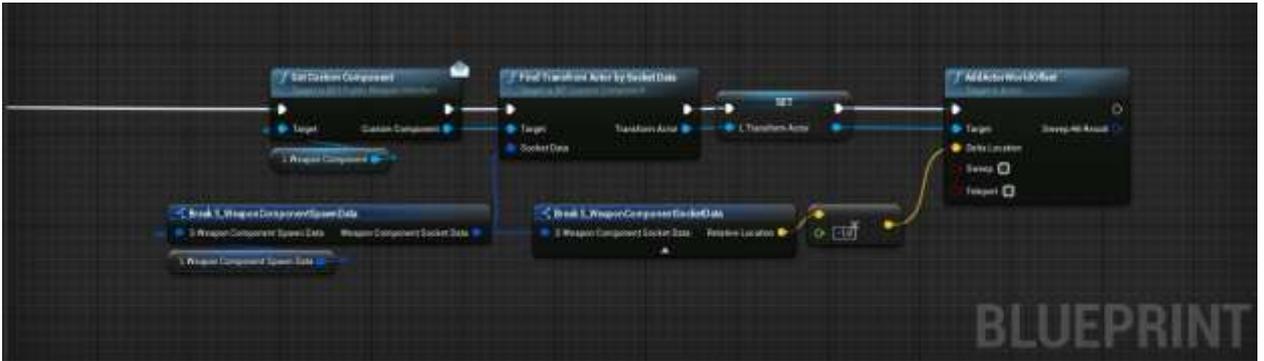
Вузол “FIND” знаходить значення, асоційоване з наданим Ключем. “Break S\_WeaponComponentSocketData” додає вузол, який розбиває аS WeaponComponentSocketData' на свої поля-члени.

Другим кроком - отримаємо Actor точки прикріплення та змінити відносне положення. Це самі компоненти, компоненти зброї які вставляються по одному, покроково, отримуємо компонент та вставляється в сокет, і встановлюється в світі, після чого повторна перевірка, чи є у цього сокета ще сокети.



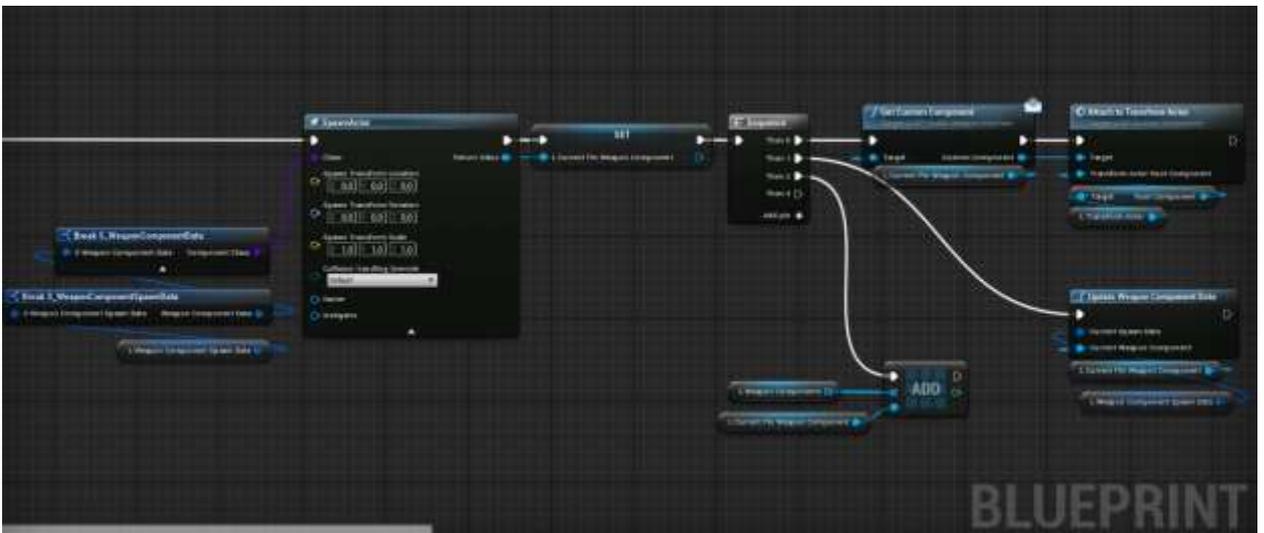
*Рис. Г.7. Огляд програмного коду генерування сцени мовою візуального програмування BluePrint*

Ця частина визначає подальше положення сокета, робить її відносною, відносно самого блока зброї, тобто фізичного об'єкту самої зброї.



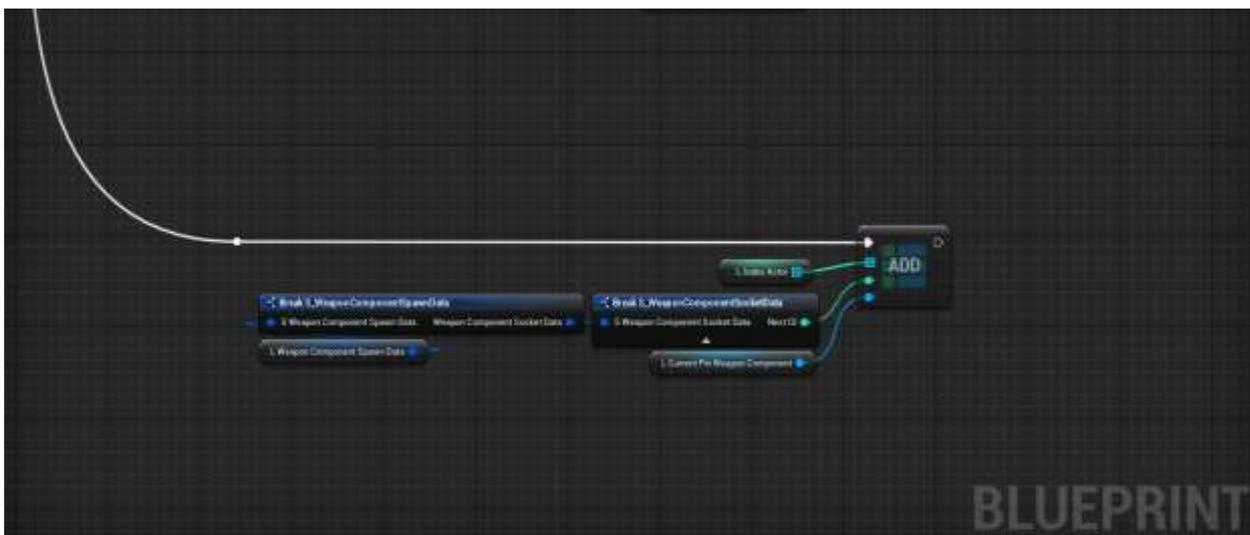
*Рис. Г.8. Огляд програмного коду генерування сцени мовою візуального програмування BluePrint*

Третім кроком - згенеруємо компонент та приєднаємо його до отриманого вище Actor, після чого повертаємось до збереження даних.



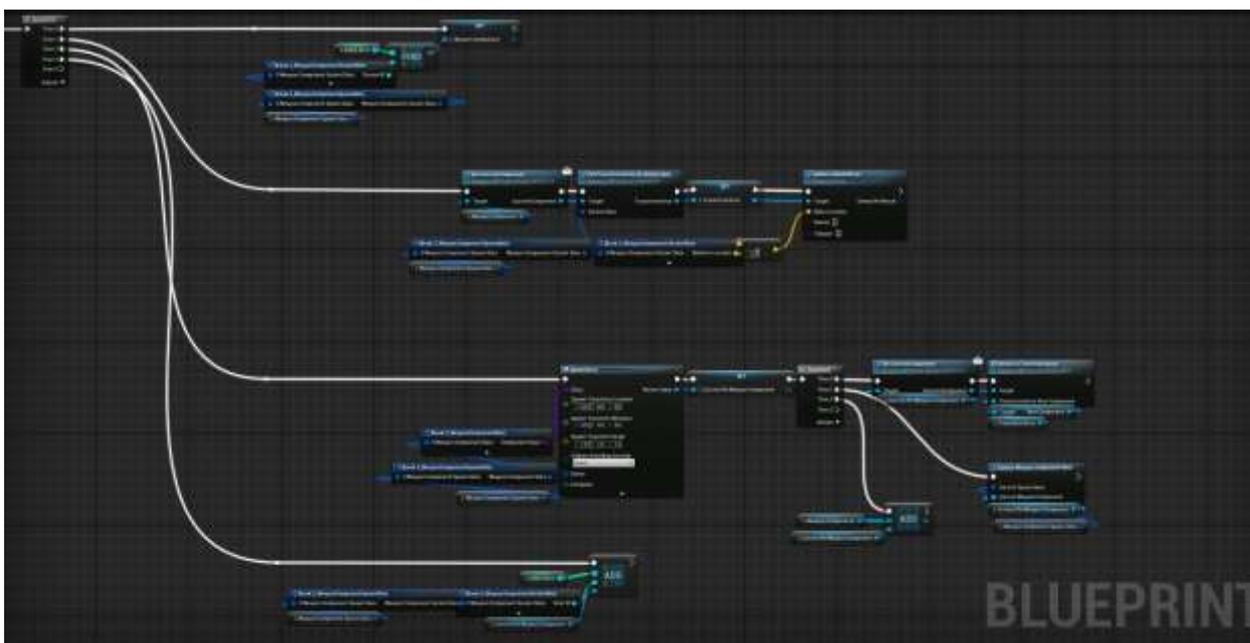
*Рис. Г.10. Огляд програмного коду генерування сцени мовою візуального програмування BluePrint*

Останнім кроком цієї послідовності – додаємо дані.



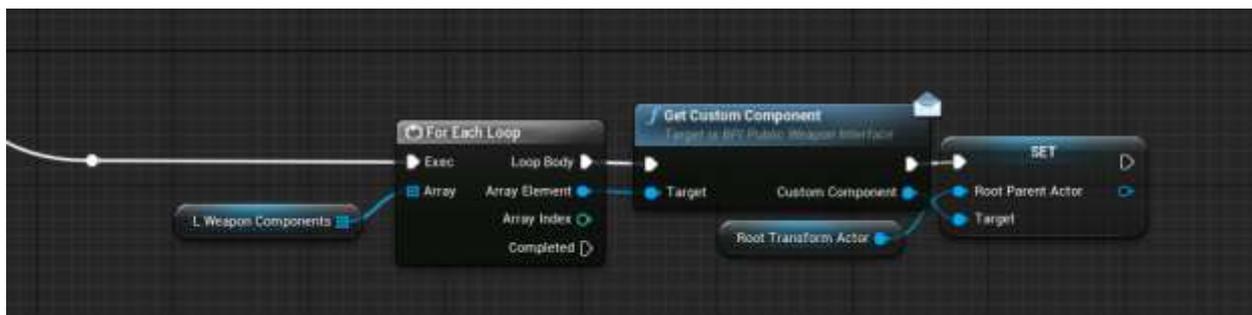
*Рис. Г.11. Огляд програмного коду генерування сцени мовою візуального програмування BluePrint*

Вся гілка виглядає наступним чином.



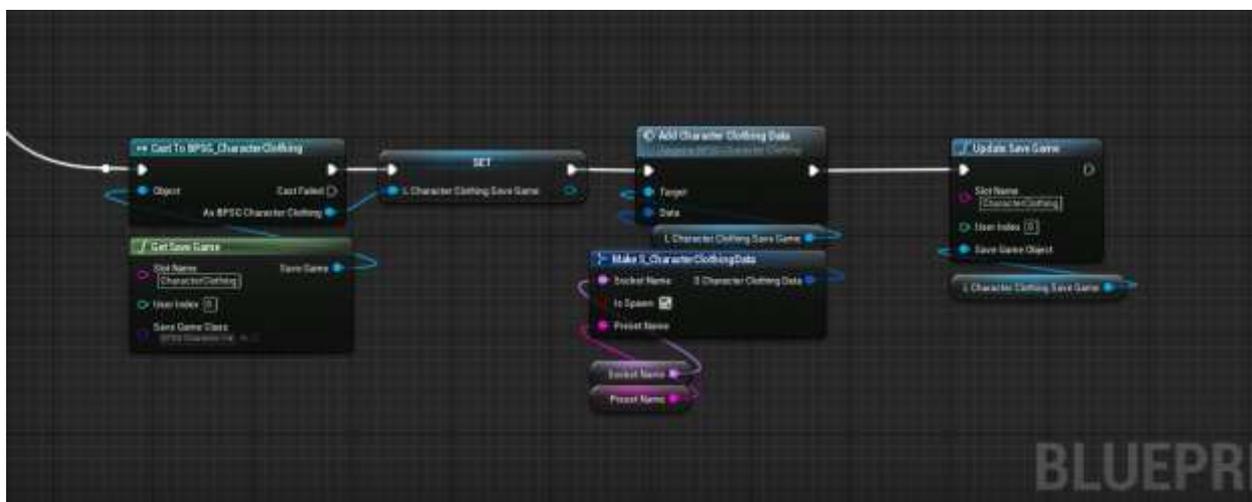
*Рис. Г.12. Огляд програмного коду генерування сцени мовою візуального програмування BluePrint*

Також передбачено і реалізовано функцію що основному використовується для отримання відносного положення всієї зброї для її кріплення, наприклад, до гравця. Ця гілка стоїть кроком чотири (Then 3) у другому та головному “Sequence” всього цього блоку.



*Рис. Г.13. Огляд програмного коду генерування сцени мовою візуального програмування Blueprint*

Також реалізовано збереження створеної користувачем, згенерованої сцени, ця гілка стоїть у цьому самому “Sequence” наступним кроком, а саме “Then 4”.



*Рис. Г.14. Огляд програмного коду генерування сцени мовою візуального програмування Blueprint*

Далі, передбачено пошук всіх елементів що вже встановлені в зброю, щоб загрузити зброю з збереження.

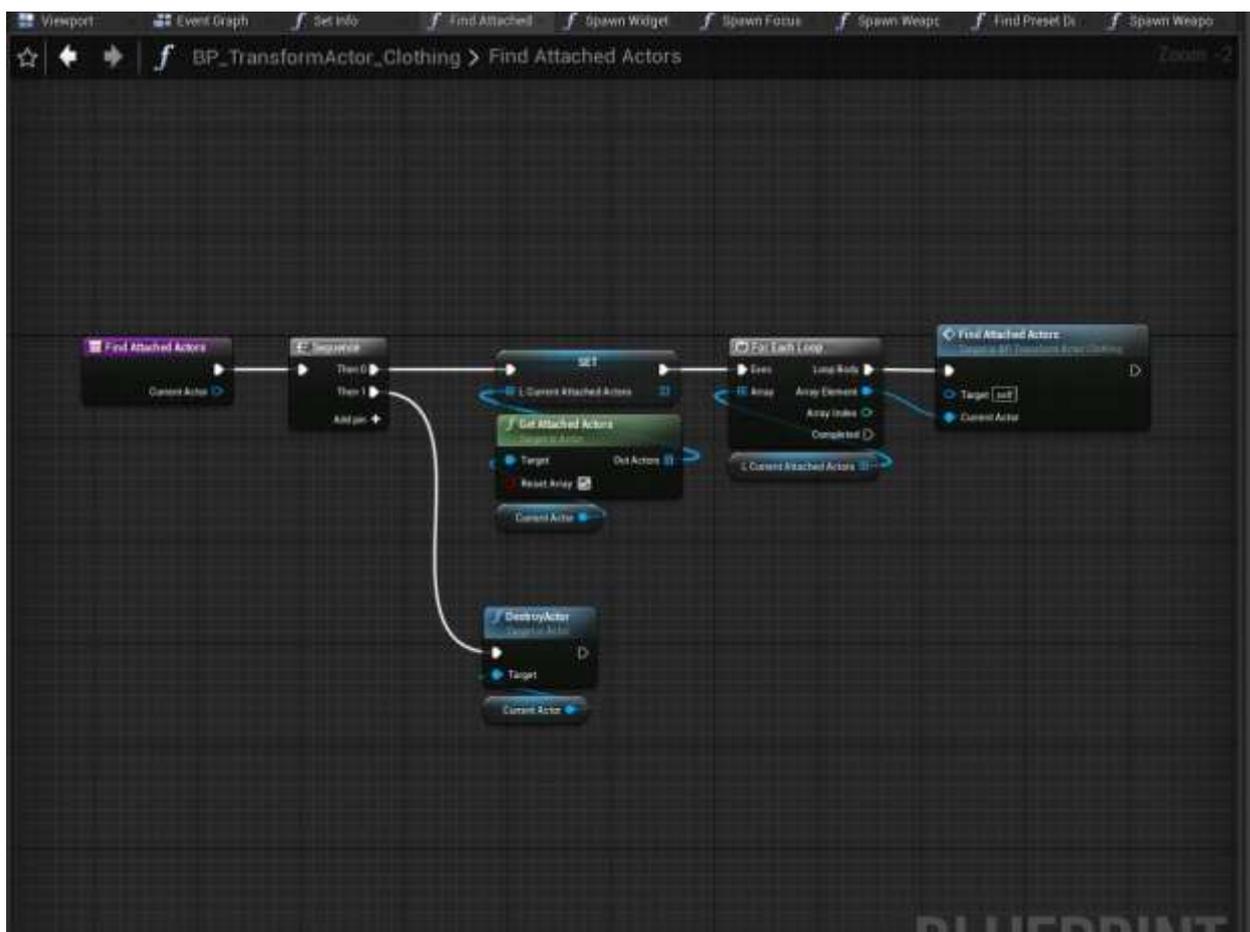


Рис. Г.15. Огляд програмного коду генерування сцени мовою візуального програмування *BluePrint*

Цей блок завантажую інтерфейс користувача.

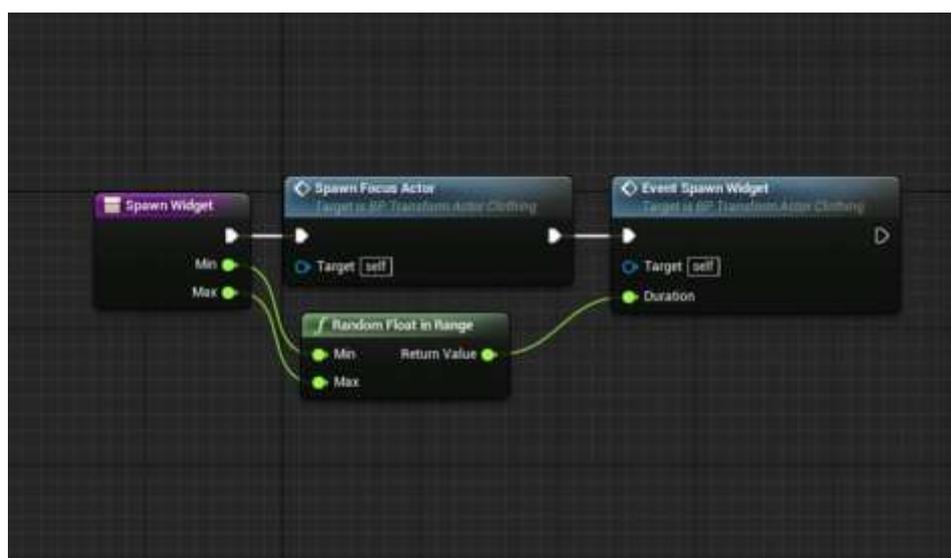
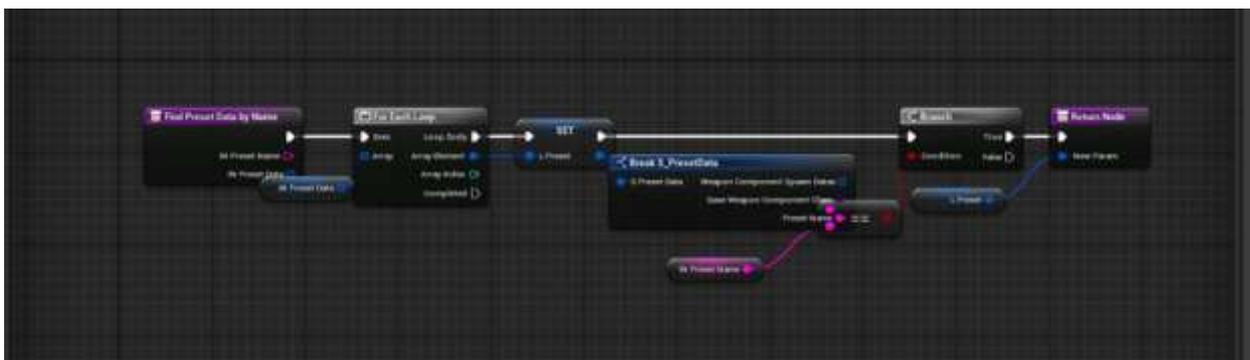


Рис. Г.16. Огляд програмного коду генерування сцени мовою візуального програмування *BluePrint*

Блок пошуку збереженого набору згенерованої сцени.



*Рис. Г.17. Огляд програмного коду генерування сцени мовою візуального програмування BluePrint*

Блок черги виклику в момент подій. Перший – начало запуску застосунку чи симуляції. Друга – тригер колізії, останній – рахує час. З них активний четвертий – це тригер момент виклику інтерфейсу користувача.



*Рис. Г.18. Огляд програмного коду генерування сцени мовою візуального програмування BluePrint*

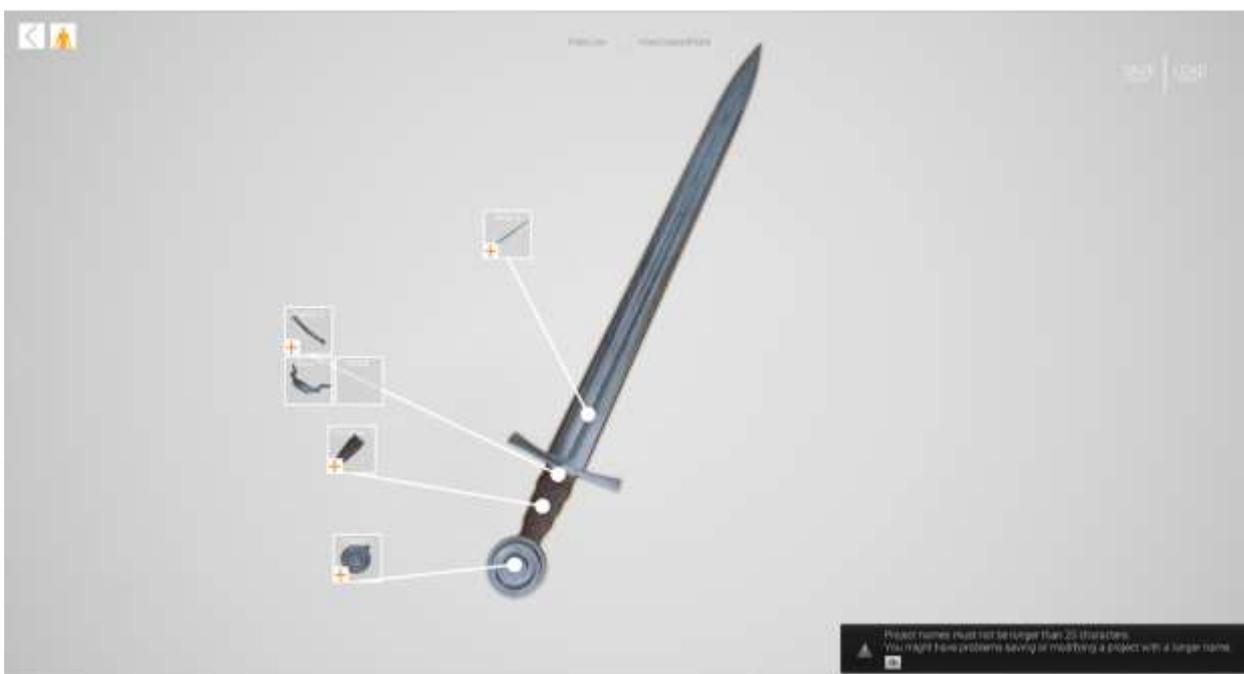
**Додаток Д.**  
**Робоче вікно застосунку та приклади генерування сцени**



*Рис. Д.1. Демонстрація роботи редактора з згенерованою сценою та елементами інтерфейсу*



*Рис. Д.2. Демонстрація роботи редактора з згенерованою сценою та елементами інтерфейсу*



*Рис. Д.3. Демонстрація роботи редактора з згенерованою сценою та елементами інтерфейсу*



*Рис. Д.4. Демонстрація роботи редактора з згенерованою сценою та елементами інтерфейсу*



*Рис. Д.5. Демонстрація роботи редактора з згенерованою сценою та елементами інтерфейсу*



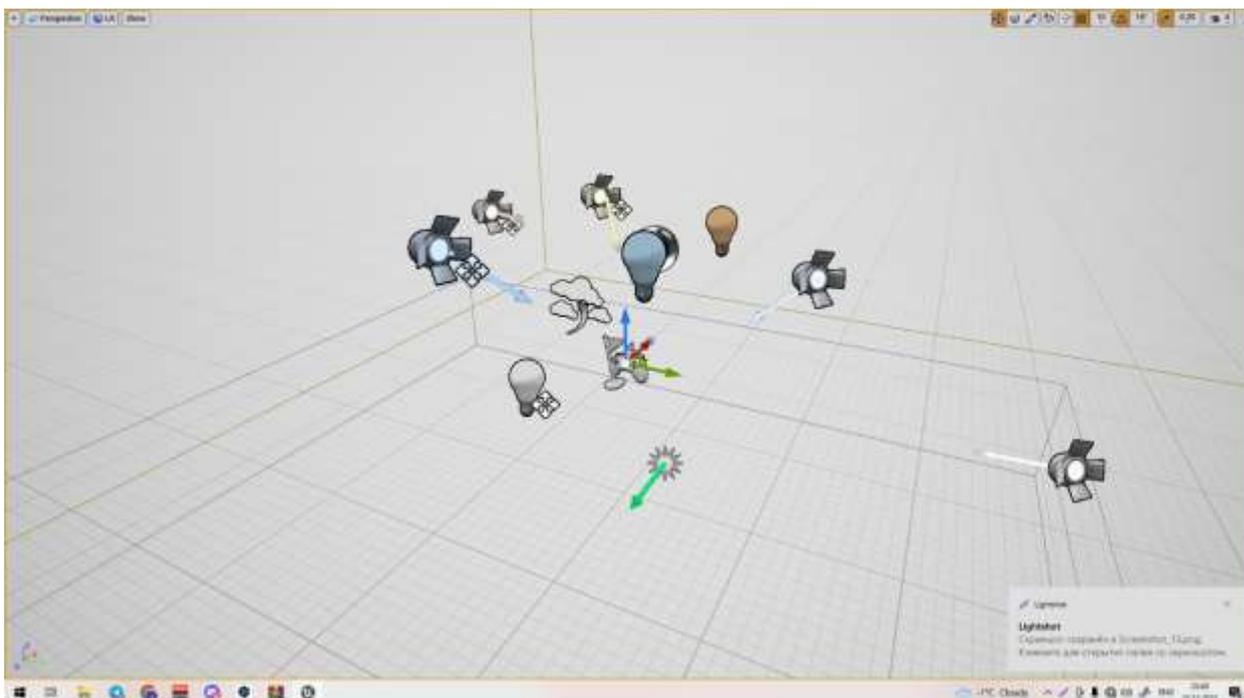
*Рис. Д.6. Демонстрація роботи редактора з згенерованою сценою та елементами інтерфейсу*



*Рис. Д.7. Демонстрація роботи редактора з згенерованою сценою та елементами інтерфейсу*



*Рис. Д.8. Демонстрація роботи редактора з згенерованою сценою та елементами інтерфейсу*



*Рис. Д.9. Демонстрація робочої сцени в інтерфейсу редактора "Unreal Engine"*