

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської кваліфікаційної роботи
на ступінь вищої освіти бакалавр

на тему: «РОЗРОБКА ПЛАГІНУ ДЛЯ СПРОЩЕНОГО СТВОРЕННЯ 3D
МОДЕЛЕЙ У ПРОГРАМІ BLENDER МОВОЮ PYTHON»

Виконав: студент 4 курсу, групи ПД– 44
спеціальності

121 Інженерія програмного
забезпечення

(шифр і назва спеціальності)

Косенко Д.М.

(прізвище та ініціали)

Керівник Корецька В.О.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Київ – 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Спеціальність - 121 Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного
забезпечення

_____ О.В. Негоденко

«____» _____ 2023 року

ЗАВДАННЯ
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Косенко Денис Максимович

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка плагіну для спрощеного створення 3D моделей у програмі Blender мовою Python»

Керівник роботи _____ к.п.н., доцент, професор кафедри Корецька В.О.

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

затверджені наказом вищого навчального закладу від «24» лютого 2023 року №26.

2. Строк подання студентом роботи «1» червня 2023 року

3. Вихідні дані до роботи:

3.1.Офіційна документація Python

3.3.Офіційна документація Blender

3.4.Науково-технічна література

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

4.1.Аналіз актуальності та огляд існуючих додатків

4.2.Аналіз інструментів реалізації та розробка структури плагіну

4.3.Реалізація програмної частини додатку.

4.4.Висновки

5. Перелік графічного матеріалу

5.1. Титульний слайд

5.2. Мета, об'єкт та предмет дослідження

5.3 Задачі дипломної роботи

5.4 Аналіз аналогів

5.5 Вимоги до плагіну

5.6 Програмні засоби реалізації

5.7 Опис функціоналу плагіну

5.8 Екранні форми

5.9 Приклад використання плагіну

5.10 Апробація результатів дослідження

5.11 Висновки

6. Дата видачі завдання «25» лютого 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Утвердження теми бакалаврської роботи	24.02.2023	Виконано
2	Підбір науково-технічної літератури	26.02.2023 – 15.03.2023	Виконано
3	Аналіз актуальності та дослідження існуючих додатків	15.03.2023 – 20.03.2023	Виконано
4	Аналіз та вибір інструментів для розробки плагінів	20.03.2023 – 14.04.2023	Виконано
5	Проектування та реалізація	14.04.2023 – 04.05.2023	Виконано
6	Вступ, реферат та висновки	04.05.2023 – 12.05.2023	Виконано
7	Попередній захист	15.05.2023 – 1.06.2023	
8	Захист роботи	1.06.2023	

Студент _____
(підпис)

Керівник роботи _____
(підпис)

Косенко Д.М.
(прізвище та ініціали)

Корецька В.О.
(прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи с. 48, рис. 29, джерел 18

Актуальність теми – З кожним днем популярність сфери 3D-моделювання зростає, особливо з ростом попиту на віддалену роботу. З ростом популярності зростає і кількість новачків. Багато хто вважає цю галузь важкою. Тому було запропоновано створити унікальний плагін, який би спростив поріг входу для новачків у цю сферу, а також прискорив процес створення візуальної складової гри для невеликих студій.

Об'єкт дослідження – Процес створення 3D моделей та плагінів.

Предмет дослідження – Плагіни для спрощеного створення 3D моделей.

Мета роботи – Спрощення та прискорення процесу створення 3D-моделей у Blender за допомогою розробленого плагіну.

Методи дослідження – Опитування користувачів, дослідження літератури, тестування та зворотний зв'язок.

Галузь використання – Розробка ігор, кіно та анімація, архітектура, реклама.

ЗМІСТ

РЕФЕРАТ.....	6
<u>ВСТУП.....</u>	<u>8</u>
<u>1. АНАЛІЗ АКТУАЛЬНОСТІ ТА ОГЛЯД ІСНУЮЧИХ АНАЛОГІВ.....</u>	<u>10</u>
1.1 Історія розвитку 3D графіки.....	10
1.2 Внесок 3D графіки у різні сфери.....	11
1.3 Аналіз актуальності обраної теми.....	16
1.4 Огляд аналогів, їх переваги на недоліки.....	17
<u>2. АЛГОРИТМ РОБОТИ ДОДАТКУ, ВИБІР СЕРЕДОВИЩА І МОВИ ПРОГРАМУВАННЯ.....</u>	<u>25</u>
2.1 Вибір середовища для розробки.....	25
2.2 Огляд середовища розробки Blender.....	36
2.3 Вибір мови програмування.....	41
2.4 Алгоритм роботи додатку.....	43
<u>3. РОЗРОБКА ПЛАГІНУ.....</u>	<u>49</u>
3.1 Загальний огляд засобів для розробки.....	49
3.2 Blender API.....	49
3.3 Python API.....	52
3.4 Процес розробки плагіну.....	53
<u>ВИСНОВКИ.....</u>	<u>56</u>
<u>ПЕРЕЛІК ПОСИЛАНЬ.....</u>	<u>57</u>
<u>ДОДАТКИ.....</u>	<u>59</u>

ВСТУП

Актуальність теми:

У сучасному світі все більше людей зацікавлені 3D моделюванням, оскільки це стає необхідним елементом візуалізації у різних галузях, включаючи кіноіндустрію, відеоігри, архітектуру та дизайн. Існує потреба у швидкому і ефективному процесі створення якісних 3D моделей. Враховуючи популярність Blender та мови програмування Python, розробка плагіну для спрощення цього процесу є актуальною задачею.

Ступінь вивченої проблеми:

Проблема створення 3D моделей у Blender вже добре вивчена, але відсутність зручних інструментів і плагінів, спрямованих на спрощення цього процесу, є актуальним недоліком. Існуючі рішення недостатньо задовольняють потреби користувачів, особливо початківців та невеликі студії.

Специфіка джерельної бази:

Для дослідження та розробки плагіну будуть використовуватися наукові статті, онлайн-документація Blender та Python, книги та електронні ресурси, пов'язані з 3D моделюванням, Blender та розробкою плагінів.

Об'єкт дослідження:

Процес створення 3D моделей та плагінів.

Предмет дослідження:

Плагіни для спрощеного створення 3D моделей.

Мета роботи:

Спрощення та прискорення процесу створення 3D-моделей у Blender за допомогою розробленого плагіну.

Завдання роботи:

1. Аналіз існуючих методів створення 3D моделей у Blender.
2. Реалізація плагіну мовою Python для програми Blender.
3. Тестування та валідація розробленого плагіну.
4. Оцінка ефективності та порівняння з існуючими рішеннями.

Методика дослідження:

Для досягнення поставлених завдань буде використана комбінація методів, включаючи аналіз літературних джерел, дослідження функціональності Blender, розробку програмного коду мовою Python, тестування та порівняння результатів.

Наукова новизна роботи:

Науковою новизною роботи є розробка плагіну, який спрощує та прискорює процес створення 3D моделей у Blender. Даний плагін буде мати унікальну функціональність та інтерфейс, що відрізнятимуть його від існуючих рішень.

Практична значущість результатів дослідження:

Результати дослідження будуть мати практичне значення для користувачів Blender, зокрема початківців та невеликих студій, які прагнуть швидко та ефективно створювати якісні 3D моделі. Розроблений плагін буде сприяти підвищенню продуктивності, зручності та якості процесу 3D моделювання.

1. АНАЛІЗ АКТУАЛЬНОСТІ ТА ОГЛЯД ІСНУЮЧИХ ДОДАТКІВ

1.1. Історія розвитку 3D графіки

3D-графіка — це одна з технологій комп'ютерної графіки, яка використовується для створення зображень, згенерованих комп'ютером (CGI). Вона включає процес створення тривимірних моделей об'єктів за допомогою спеціалізованих програм. 3D-об'єкти широко використовуються в різних сферах, таких як веб-дизайн, мобільні додатки, віртуальна та доповнена реальність (VR і AR) і створення персонажів для ігор і фільмів.

Розвиток технологій і образотворчого мистецтва призвів до зміни способу відображення об'єктів на двовимірній площині, наприклад, на полотні, папері чи екрані мобільного пристрою. Раніше художник міг використовувати перспективу, щоб створити ілюзію глибини малюнка, і фотографія могла представляти те саме завдання. Разом з появою комп'ютерів з'явилася можливість моделювати тривимірний простір і предмети.

Тривимірна комп'ютерна графіка була розроблена в 1960-х роках. Перші векторні зображення склалися з точок і кривих, визначених математичними формулами. Іван Сазерленд та Девід Еванс заснували перший у світі факультет комп'ютерної графіки в університеті штату Юта, США. У цей час була створена перша програма Sketchpad, яка є прабатьком усіх сучасних 3D-редакторів.[1]

Спочатку перші комп'ютери могли працювати тільки з векторними зображеннями. Потім з'явилася растрова графіка, яка дала можливість зображувати об'єкти у вигляді сітки пікселів. У майбутньому можливості візуалізації розширилися, дозволивши використовувати текстури та тіні для створення більш реалістичних зображень.[2]

Основою сучасної тривимірної графіки є математичні поняття. Визначення поверхні в тривимірному просторі неможливо без знання геометрії і володіння функціями. Проте сучасні програми, такі як 3ds Max, Cinema 4D, Blender, ZBrush, Autodesk Maya, Unreal Engine, Houdini, дозволяють створювати 3D-зображення без

виконання складних математичних розрахунків вручну.

1.2. Внесок 3D графіки у різні сфери

З розвитком процесорів, дисплеїв, програмного забезпечення та інших технологій з'являлося все більше можливостей використання графіки. 3D-технології незамінні при розробці промислового дизайну, проектуванні будівель, створенні фільмів і відеоігор, веб-сайтів і додатків. 3D графіка застосовується майже у всіх сферах, зараз ми розглянемо докладніше деякі з них:

1. Реклама і брендинг

Тривимірна графіка широко використовується в сфері реклами та брендингу, де її головним призначенням є створення візуальних ефектів і привернення уваги потенційних клієнтів.[3]

Завдяки 3D моделюванню можна створювати реалістичні зображення товарів і предметів, що дозволяє детальніше і точніше показати їх характеристики і переваги. Це може бути особливо корисно для продуктів, які ще не випущені на ринок, або для продуктів, які важко продемонструвати в реальному житті.

3D-графіка також дозволяє створювати ефектні анімації та відео, які запам'ятовуються глядачам і залишають яскраве враження. Такі ролики можна використовувати як рекламу на телебаченні, в Інтернеті або в кінотеатрах.

Крім того, її можна використовувати для створення унікальних та оригінальних дизайнів упаковки, логотипів, банерів та інших елементів брендування. Тривимірна графіка дозволяє експериментувати з кольорами, текстурами та освітленням, у зв'язку з чим можна створювати унікальні дизайни, що запам'ятовуються.

Загалом, використання 3D-графіки в рекламі та брендингу дозволяє створювати більш ефективні та переконливі кампанії, привертати увагу потенційних клієнтів і запам'ятовуватися в їхній свідомості.

2. Мобільний Інтерфейс

Використання 3D-графіки в мобільних інтерфейсах стає все більш

популярним. Все завдяки тому, що 3D-елементи можуть значно покращити візуальний досвід і функціональність мобільних програм, ось декілька прикладів:

Анімовані 3D-іконки та елементи керування. Використання 3D-елементів може додати додаткову глибину та інтерактивність інтерфейсу, підвищуючи залучення користувачів. Наприклад, ви можете додати анімовані 3D-іконки до відкритих вкладок або спадних меню.[4]

3D об'єкти для візуалізації товару. Якщо ваша програма надає користувачеві доступ до продукту, то використання 3D-моделей може покращити його візуальне представлення. Наприклад, програма для купівлі меблів може використовувати 3D-моделі, щоб дозволити користувачеві побачити, як виглядатимуть меблі в їхньому домі.

3D-анімація для навчання та розваг. 3D-анімації можуть покращити процес навчання та зробити його більш інтерактивним та цікавим. Наприклад, анатомічний додаток може використовувати 3D-анімацію, щоб показати внутрішні органи та їх функції.

3D моделювання для створення доповненої реальності. Доповнена реальність (AR) – це технологія, яка додає віртуальні об'єкти до реального світу. 3D-моделювання дозволяє створювати ці віртуальні об'єкти, які можна використовувати в програмах AR. Наприклад, додаток для покупки одягу з доповненою реальністю може використовувати 3D-моделювання для створення віртуальних примірок для користувача.

3. Медицина

Використання 3D-графіки в медицині стає все більш поширеним. Його можна використовувати для створення віртуальних моделей органів і тканин людини, для моделювання хірургічних процедур, для навчання студентів-медиків і для покращення розуміння захворювань і процесів в організмі.[5]

Одним із найпоширеніших застосувань тривимірної графіки в медицині є створення віртуальних моделей органів і тканин. Ці моделі можна генерувати з медичних зображень, таких як МРТ і КТ, і використовувати для діагностики та планування операції. Наприклад, хірург може використовувати віртуальну модель

пацієнта, щоб зрозуміти, яким буде оптимальний підхід до операції.

Також вона використовується для імітації хірургічних процедур, що дозволяє хірургам тренуватися на віртуальних пацієнтах. Це знижує ризик помилок і підвищує точність процедури.

Інше важливе застосування 3D-графіки в медицині – це навчання студентів-медиків і покращення розуміння хвороб і процесів в організмі. За допомогою віртуальних моделей органів і тканин можна краще зрозуміти анатомію та функціонування людського тіла, що сприяє підвищенню якості медичної освіти та практики.

Загалом використання 3D-графіки в медицині може підвищити якість медичної практики та освіти, знизити ризик помилок і підвищити точність процедур, заощадити час і гроші.

4. Кіно

Тривимірна графіка стала незамінним інструментом у кіноіндустрії, дозволяючи створювати приголомшливі візуальні ефекти та сцени, які були б неможливі традиційними методами.[6]

Одним із перших фільмів із використанням 3D-графіки був «Термінатор 2: Судний день» (1991), у якому створено CGI-ефекти для персонажа з рідкого металу T-1000. Потім технологія набула все більшого поширення в кіноіндустрії, і в наш час практично всі фільми з використанням візуальних ефектів містять елементи 3D-графіки.

Одним із найпомітніших прикладів використання 3D-графіки в кінематографі є фільм «Аватар» (2009), який повністю створений за технологією 3D. У цьому фільмі створена вся планета Пандора з живими істотами, рослинністю та унікальною атмосферою завдяки використанню 3D-моделювання та анімації.

Також 3D-графіка широко використовується при створенні супергеройських фільмів, наприклад, «Месники» (2012), «Людина-павук» (2012), «Залізна людина» (2008), де створення реалістичних костюмів і спец. ефекти були б неможливі без використання 3D технологій.



Рисунок 1.1 — Приклад 3D графіки у фільмі «Гобіт» (кінотрилогія)

5. Ігри

Використання 3D-графіки в іграх має довгу історію і почалося ще в 70-х роках з появою перших ігрових автоматів. З часом технології розвивалися, і можливості комп'ютерної графіки в іграх ставали ширшими та виразнішими.

3D графіка використовується в іграх для створення реалістичних і деталізованих світів, персонажів і об'єктів. Це дозволяє створювати середовища,

які можна досліджувати та взаємодіяти з ними в інтерактивному режимі. 3D-графіка також забезпечує більш складну та реалістичну фізику, світлові ефекти, анімацію та візуальні ефекти, які роблять ігровий процес більш захоплюючим і веселим.

За допомогою 3D-графіки можна створювати ігри різних жанрів, такі як шутери, RPG, платформери, гонки, стратегії та багато інших. Це також дозволяє створювати багатокористувацькі ігри, де гравці можуть взаємодіяти з іншими гравцями в реальному часі.[7]

З розвитком технологій тривимірна графіка в іграх стала більш складною та детальною, що призвело до створення більших та реалістичніших ігрових світів. Крім того, розвиток віртуальної та доповненої реальності відкрив нові можливості для використання 3D-графіки в іграх і дозволив гравцям глибше та реалістичніше зануритися в ігровий світ, створюючи новий рівень взаємодії та участі з ігровим процесом.

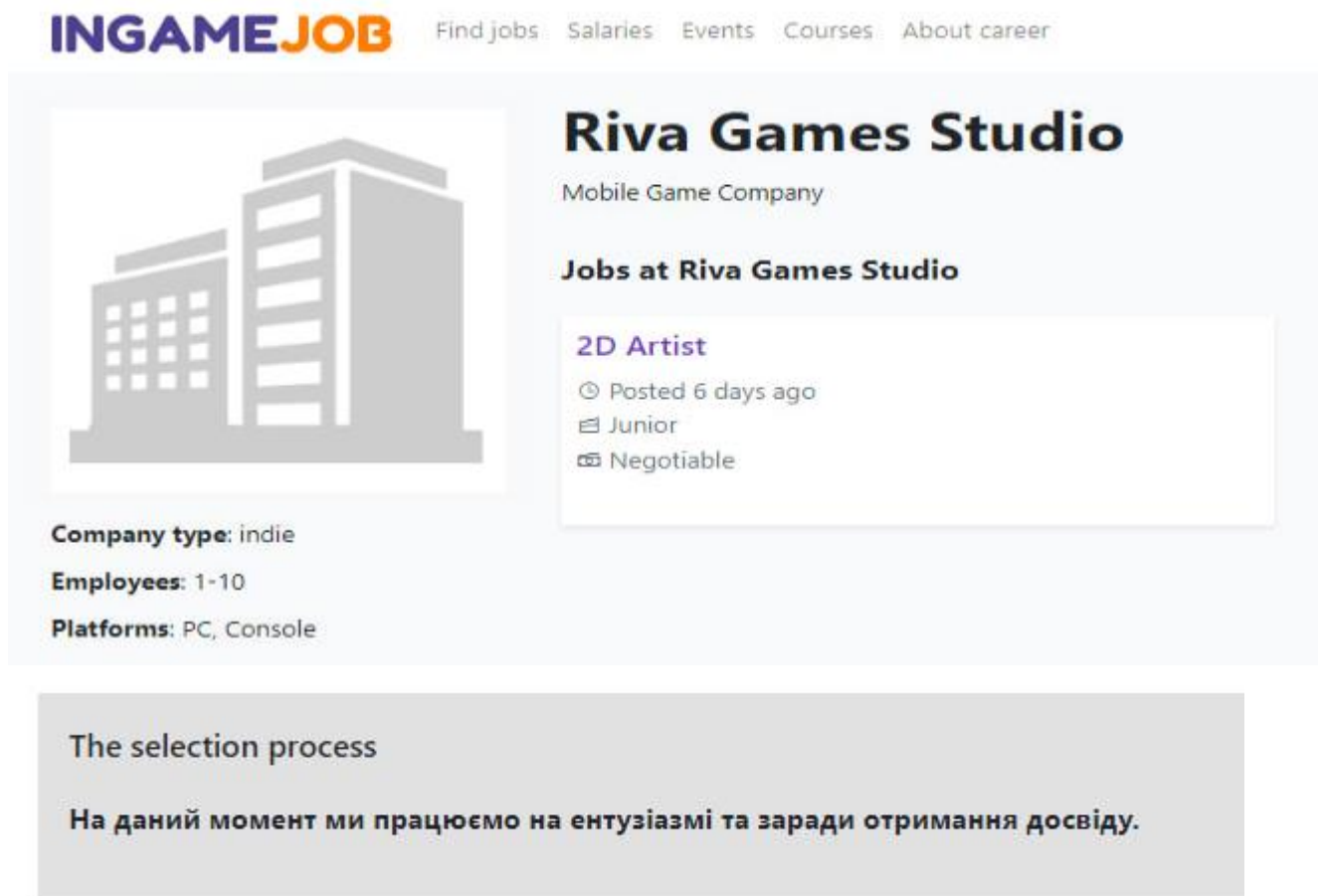


Рисунок 1.2 — Гра Elden Ring (PC / PlayStation 5 / Xbox One)

1.3. Аналіз актуальності обраної теми

Проаналізувавши вищезгадане, у кожного може з'явитися бажання спробувати себе у цій сфері, і він піде завантажувати потрібні програми. Але для користувачів-початківців, Blender може бути непростим і складним у використанні. Створення 3D моделей вимагає знань та досвіду, і може зайняти багато часу та зусиль.

Робота над 3D-моделями це складне і тривале завдання, яке може зайняти багато часу та зусиль, особливо для невеликих студій або фрілансерів, які не мають доступу до дорогих інструментів і програмного забезпечення. Це може обмежити їхні можливості та уповільнити процес створення якісних 3D-моделей.



INGAMEJOB Find jobs Salaries Events Courses About career

Riva Games Studio
Mobile Game Company

Jobs at Riva Games Studio

2D Artist
 ☉ Posted 6 days ago
 📄 Junior
 💰 Negotiable

Company type: indie
Employees: 1-10
Platforms: PC, Console

The selection process
 На даний момент ми працюємо на ентузіазмі та заради отримання досвіду.

Рисунок 1.3 — Приклад indie студій

Однак розробка плагіна для Blender - чудовий спосіб допомогти таким користувачам, надаючи доступні та ефективні інструменти 3D-моделювання.[8]

Плагін може надати користувачеві додаткові функції та можливості для прискорення процесу створення 3D-моделей, такі як швидке створення поверхонь і форм, інструменти для освітлення, а також автоматизація деяких процесів. Це може значно скоротити витрати часу та ресурсів, дозволяючи користувачам зосередитися на творчій стороні своїх проєктів.

Крім того, плагін може бути корисним для новачків у 3D-моделюванні, яким важко вивчити нові інструменти та функції Blender. Плагін може зробити процес навчання простішим, доступнішим та інтуїтивно зрозумілішим для початківців.

Отже, плагін, який надає доступні та потужні інструменти 3D-моделювання, може зробити процес створення 3D-моделей набагато простішим і швидшим, а також дозволить користувачам зосередитися на творчих аспектах своєї роботи. Таким чином, плагін може бути цінним інструментом для невеликих студій і фрілансерів, які шукають способи підвищити ефективність своєї роботи.

1.4. Огляд аналогів, їх переваги та недоліки

Існує багато плагінів та інструментів, які допомагають спростити та прискорити процес створення 3D-моделей у Blender. У цьому розділі ми розглянемо кілька аналогів плагіна, і розберемо їх переваги і недоліки.

1. HardOps

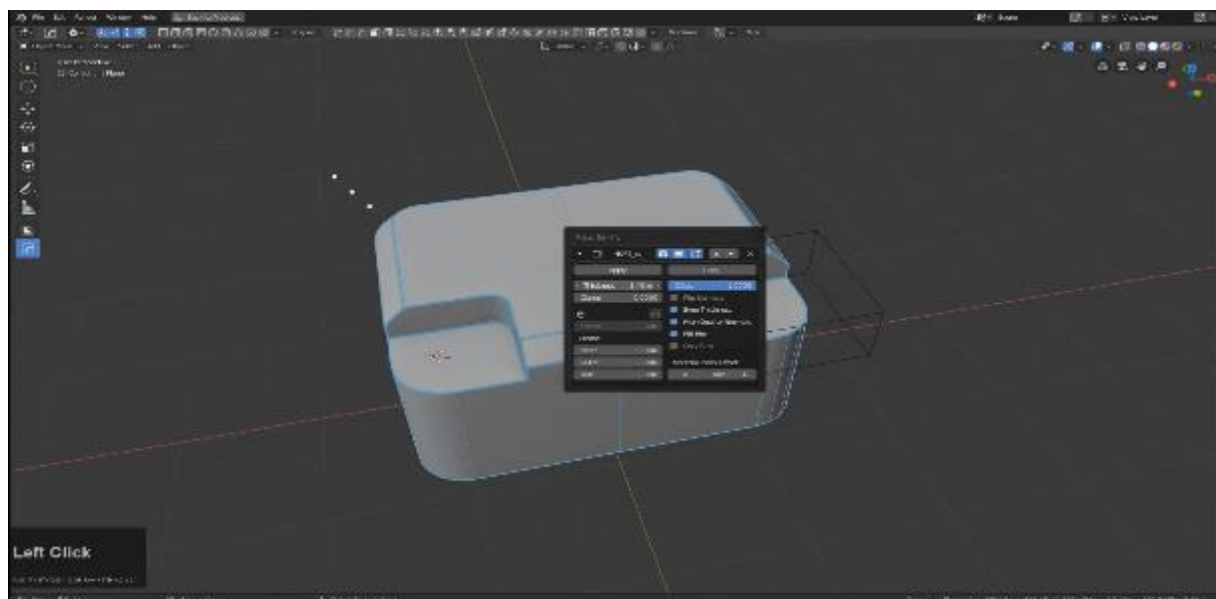


Рисунок 1.4 — Інтерфейс плагіну HardOps

HardOps — популярний плагін для Blender, призначений для прискорення процесу моделювання та створення складних геометричних фігур. Його створили Джоел Бертон і Андре Льюїс.

Переваги HardOps:

1. Прискорення процесу моделювання: HardOps пропонує безліч інструментів, які спрощують та прискорюють процес моделювання, такі як автоматична генерація ребер, кривих та багато інших.

2. Створення складних геометричних форм: HardOps надає потужний інструментарій для створення складних форм, які важко чи неможливо створити вручну

3. Легкість використання: Плагін має інтуїтивний інтерфейс, який дозволяє швидко освоїти його та почати використовувати у роботі.

4. Підтримка оновлень: HardOps постійно оновлюється, додаючи нові функції та покращення.

Недоліки:

1. Обмеження використання: Плагін може бути корисним лише тим, хто працює зі створенням 3D-моделей у Blender.

2. Необхідність вивчення додаткових інструментів: Використання плагіна вимагає вивчення додаткових інструментів та методів роботи у Blender.

3. Ціна: HardOps не є безкоштовним плагіном. Для отримання доступу до всіх функцій та оновлень необхідно придбати ліцензію.

2. MeshMachine

Плагін MeshMachine для Blender — це потужний інструмент моделювання, який дозволяє автоматично створювати складні геометричні фігури, а також покращувати існуючу модель. Цей плагін був розроблений відомим 3D-художником Максимом Романенко і є одним із найпопулярніших плагінів для Blender.

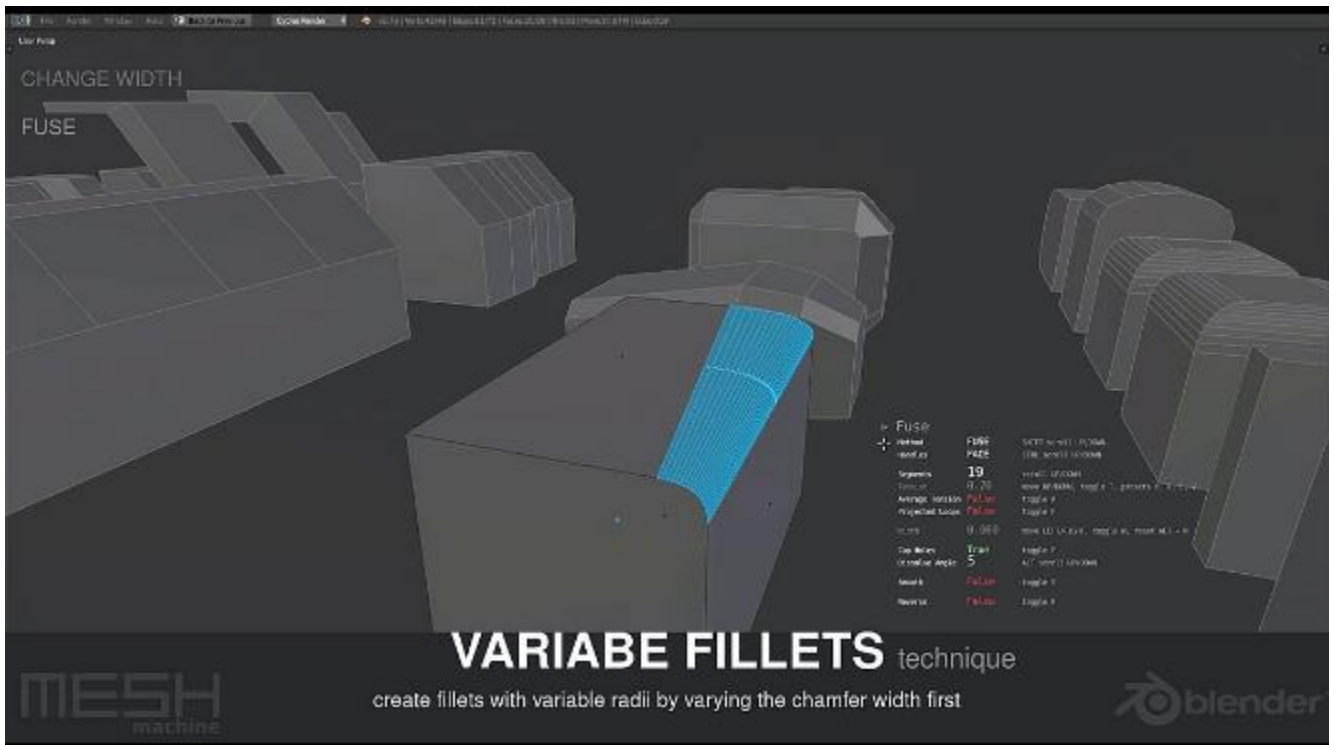


Рисунок 1.5 - Приклад роботи MeshMachine

Переваги MeshMachine:

1. Автоматичне створення геометричних форм: MeshMachine дозволяє створювати складні геометричні форми, такі як тори, спіралі, витки та багато інших, лише кілька кліків.

2. Поліпшення існуючих моделей: MeshMachine також дозволяє оптимізувати геометрію існуючих моделей. Це може допомогти прискорити процес роботи з моделлю та покращити її продуктивність.

3. Наочність і зручність використання: інтерфейс MeshMachine інтуїтивно зрозумілий і простий у використанні. Користувач може легко налаштувати параметри створених форм і отримати бажаний результат.

Недоліки MeshMachine:

1. Ціна: MeshMachine є комерційним плагіном і коштує близько 80 доларів США. Це може бути досить високою ціною для деяких користувачів.

2. Не завжди підходить для всіх завдань: MeshMachine є потужним інструментом для створення складних форм, але не завжди може бути придатним для всіх завдань, таких як створення базових форм.

3. BoxCutter

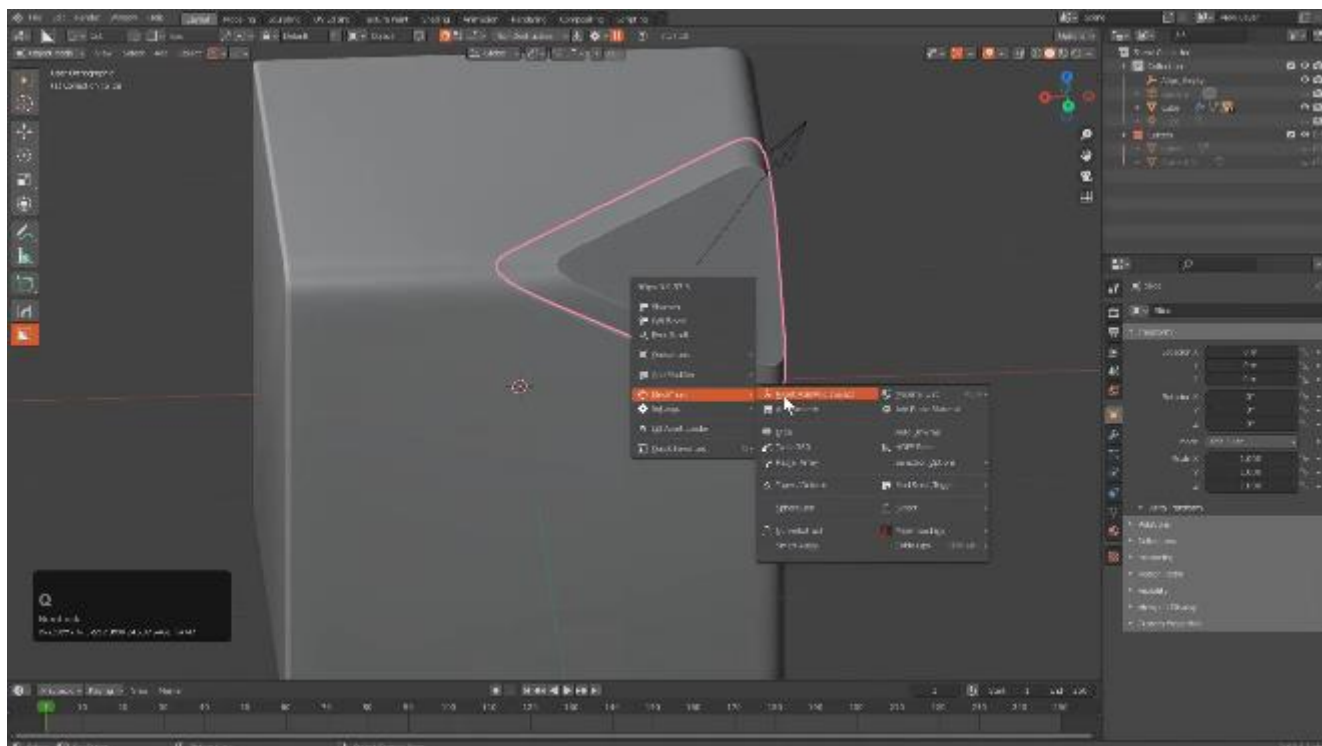


Рисунок 1.6 - Інтерфейс плагіну BoxCutter

Плагін BoxCutter — це інструмент для Blender, який дозволяє створювати високоякісні моделі за допомогою логічних операцій. Він призначений для швидкого створення складних форм і геометричних деталей шляхом створення кубічних і прямокутних вирізів у 3D-моделях.

Переваги BoxCutter включають:

1. Зручність використання: BoxCutter має простий та інтуїтивно зрозумілий інтерфейс, що дозволяє швидко створювати складні моделі.
2. Гнучкість: BoxCutter можна використовувати для створення різних об'єктів, від простих форм до складних деталей.
3. Висока продуктивність: BoxCutter забезпечує швидке виконання операцій над моделями, що дозволяє скоротити час, необхідний створення моделей.

З недоліків плагіну:

1. Деякі проблеми з продуктивністю: BoxCutter може працювати повільно з великими моделями.
2. Обмеження використання: BoxCutter не завжди підходить для створення

складних моделей, таких як органічні форми або моделі з безліччю кривих.

4. RetopoFlow

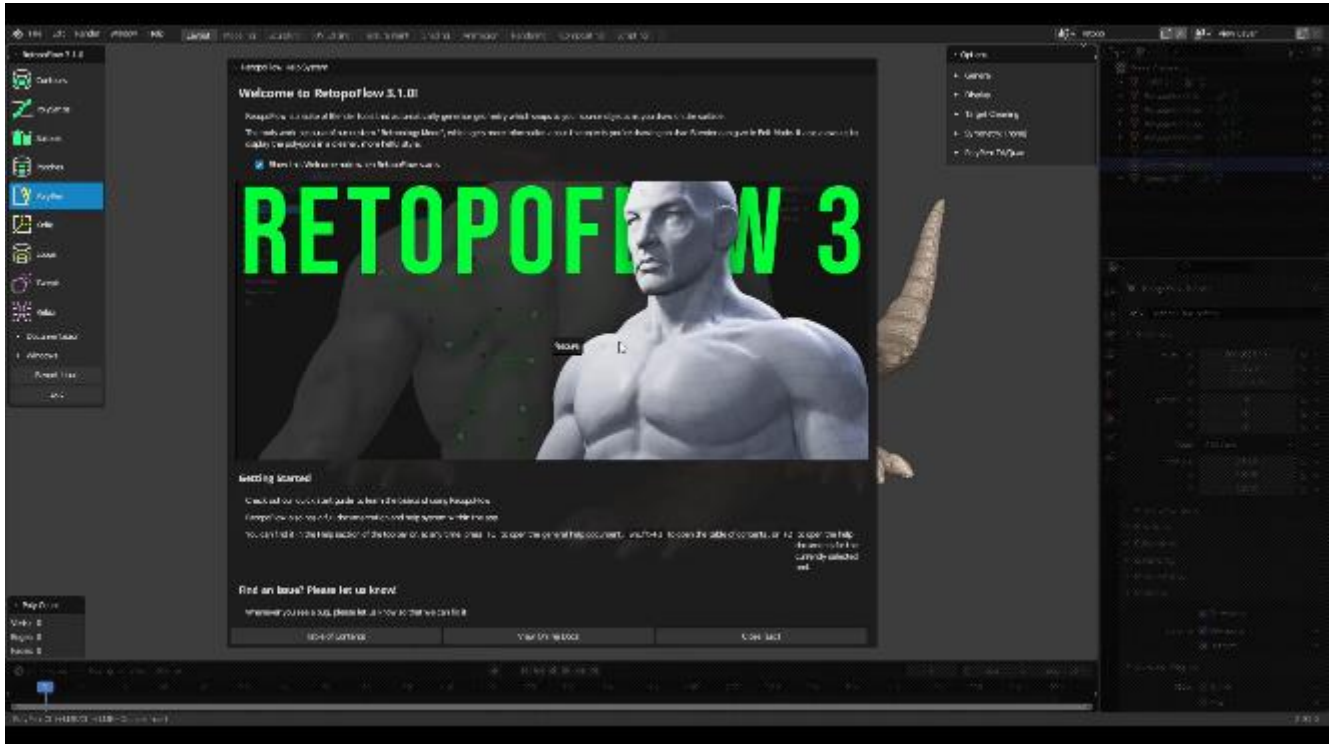


Рисунок 1.7 - Приклад старту роботи з плагіном RetopoFlow

RetopoFlow — це плагін Blender для швидкої та ефективної ретопології 3D-моделей. RetopoFlow був розроблений для покращення процесу роботи з моделями, створеними на основі сканувань або під час реконструкції існуючих моделей.

Перевагами RetopoFlow є:

1. Інструменти ретопології: RetopoFlow має інструменти, які роблять процес ретопології більш зручним та швидким. Ці інструменти дозволяють створювати нові топологічні поверхні з урахуванням існуючих моделей.

2. Інтерфейс: Інтерфейс RetopoFlow зручний та інтуїтивно зрозумілий. Це дозволяє користувачам легко освоїти плагін, а також суттєво прискорює процес роботи над проектами.

3. Інтеграція: RetopoFlow інтегрується без проблем у робочий простір Blender і може бути легко використаний з іншими плагінами та інструментами.

З недоліків можна перерахувати:

1. Вартість: RetopoFlow є комерційним плагіном, який вимагає оплати для

отримання доступу. Це може бути перешкодою для багатьох користувачів, особливо для тих, хто тільки починає знайомитися з Blender.

2. Обмеження: RetopoFlow орієнтований на ретопологію, і не має таких потужних функцій, як HardOps і VoxCutter, які можуть бути використані для більш загальних завдань, пов'язаних із створенням 3D моделей.

5. KIT OPS



Рисунок 1.8 - Приклад роботи з KIT OPS

KIT OPS — плагін для Blender, який дозволяє створювати складні об'єкти за допомогою бібліотек блоків (компонентів), створених іншими користувачами або самим користувачем. Його можна використовувати для створення різних об'єктів, таких як будинки, меблі, транспортні засоби, механічні пристрої тощо.

Основною перевагою KIT OPS є його здатність створювати складні моделі шляхом комбінування готових комплектів (KIT) - наборів готових елементів, таких як стіни, підлога, двері, вікна та інші деталі. Комплекти можуть бути створені користувачем або завантажені з бібліотеки KIT OPS. Завдяки такому модульному підходу процес створення детальних моделей спрощується та прискорюється, що дозволяє зосередитися на композиції та макеті сцени.

KIT OPS також гнучкий і настраюється. Він пропонує багато параметрів і опцій для редагування згенерованих елементів, що дозволяє користувачеві змінювати їхні розміри, матеріали та інші атрибути. Крім того, плагін підтримує набори пензлів, які дозволяють користувачам створювати власні пензлі та маски для більш точного контролю над текстуруванням.

Однак у KIT OPS є і деякі недоліки. По-перше, плагін із самого початку має обмежену бібліотеку наборів, і користувачі можуть зіткнутися з обмеженнями у виборі доступних елементів. У таких випадках їм може знадобитися створити власних китів або шукати додаткові ресурси.

По-друге, використання KIT OPS може бути складним для новачків у Blender. Розуміння та освоєння всіх функцій і налаштувань плагіна вимагає часу та вивчення. Деякі функції може бути важко зрозуміти без попереднього досвіду роботи з Blender.

Загалом, KIT OPS — це потужний інструмент Blender, який значно спрощує та прискорює процес створення складних 3D-моделей і сцен. Він пропонує гнучкість і налаштування, хоча може знадобитися додаткова робота під час створення власних елементів. Досвідчені користувачі Blender знайдуть KIT OPS як цінний інструмент для вдосконалення своїх

Проаналізувавши розглянуті аналоги, можна зробити висновок, що кожен з них має свої переваги і недоліки, і не може вирішити всі проблеми, пов'язані зі створенням 3D моделей. HardOps має потужні функції, але новачкам може бути важко користуватися. MeshMachine простий у вивченні, але не такий потужний, як HardOps. VoxCutter гнучкий у використанні, але має обмежений набір інструментів.

Нижче наведена таблиця порівняння аналогів і плагіна, що розробляється:

Таблиця 1.1 – Порівняння аналогів

	HardOps	MeshMachine	BoxCutter	RetopoFlow	KITOPS	TownMaker
Вартість	-	+	+	+	-	+
Кількість функцій	+	-	+	-	-	-
Необхідність вивчення додаткових функцій	+	-	+	+	-	-
Оптимізація геометрії	+	+	-	+	-	-
Простота використання	-	-	-	+	+	+
Продуктивність	-	-	-	+	-	+
Інші особливості	Постійна підтримка оновленнями	Не підходить для створення базових форм	Не підходить для створення органічних форм та моделей з безліччю кривих	Використовується тільки для ретопологій	Необхідність використовувати сторонні бібліотеки, що призводить до додаткових витрат	Швидке створення повноцінних моделей, модульна система

У зв'язку з цим розробка кращого і актуальнішого плагіна для Blender, який буде враховувати переваги і недоліки цих аналогів, є актуальним завданням. Такий плагін повинен забезпечувати максимальну зручність і швидкість роботи з 3D моделями, мати інтуїтивно зрозумілий інтерфейс для всіх користувачів, незалежно від рівня досвіду. Також важливим фактором є гнучкість плагіна, що дозволяє вирішувати широкий спектр завдань, пов'язаних зі створенням 3D-моделей.

2. АЛГОРИТМ РОБОТИ ДОДАТКУ, ВИБІР СЕРЕДОВИЩА І МОВИ ПРОГРАМУВАННЯ

2.1. Вибір середовища для розробки

Створення плагінів для 3D-програм може бути досить складним процесом, що вимагає знань програмування, математики та графіки. Однак існує багато інструментів і бібліотек, які допоможуть спростити цей процес.

Blender — це потужна та безкоштовна програма для 3D-моделювання та анімації, яка має вбудовану систему плагінів під назвою Python API. Blender дозволяє створювати плагіни на Python, який є досить простою та популярною мовою програмування. API Blender Python містить багато функцій і методів для маніпулювання об'єктами, матеріалами, світлом і анімацією.

Однією з переваг Blender є те, що він має відкритий вихідний код, що робить його гнучким і налаштованим відповідно до потреб користувача. Крім того, Blender має величезну спільноту користувачів і розробників, які створюють додаткові інструменти та плагіни для програми. Це значно спрощує процес створення плагінів і підвищує доступність для початківців розробників.

Загалом Blender надає широкий спектр інструментів і опцій для створення плагінів, що робить його зручним вибором для розробників і любителів.

Незважаючи на те, що для цієї роботи був обраний Blender, він має багато аналогів, які будуть далі розглянуті, попутно вказавши на їх переваги і недоліки.

1. Autodesk Maya

Maya є одним із найпопулярніших програмних продуктів для 3D-моделювання, анімації та візуалізації. Історія його розвитку почалася в кінці 1990х років, коли Alias Research розробляла нову програму для створення комп'ютерної графіки.[9]

Перша версія Maya була випущена в лютому 1998 року. Це було інтегроване середовище для створення 3D-графіки, яке включало інструменти для моделювання, анімації, візуалізації, компонування тощо. На той час Maya була

однією з небагатьох програм, яка могла запропонувати такі широкий набір функцій і можливостей.



Рисунок 2.1 — Приклад інтерфейсу Maya

Autodesk Maya названо на честь санскритського слова $\square\square\square\square$ māyā, майя, що означає «ілюзія». Майя була в трьох версіях:

1. Maya Unlimited - найповніший і найдорожчий пакет. Містить розширення Hair, Fur, Maya Muscled, Fluid Effects, Cloth та інші.

Вона була випущена компанією Alias|Wavefront (пізніше придбаною Autodesk) у 2002 році як відповідь на потреби професійних аніматорів і модельєрів. Maya Unlimited містить багато додаткових інструментів і технологій, які були розроблені для створення складніших і реалістичніших анімацій і зображень.

2. Maya Complete — базова версія пакета, в якій присутній повноцінні блоки моделювання та анімації, але відсутні модулі фізичної симуляції. Ця версія створена для невеликих студій і індивідуальних користувачів, яким не потрібен

повний набір функцій і можливостей Maya Unlimited.

Maya Complete надає користувачам основні інструменти для створення 3D-графіки, такі як моделювання, текстурювання, анімація та візуалізація. Він також містить деякі додаткові інструменти, такі як Paint Effects (інструмент рослинності та води), Toon Shading (інструмент комічних ефектів) та інструменти для створення динамічних ефектів, таких як вибухи, дим тощо.

Однак, порівняно з Maya Unlimited, Maya Complete має деякі обмеження, такі як обмежена кількість інструментів для створення тварин і персонажів, обмежена кількість інструментів для створення рідини та газу та обмежена кількість інструментів для динамічних ефектів.

3. Maya Personal Learning Edition – безкоштовний пакет для некомерційного використання. Є функціональні обмеження, обмеження розмір візуалізованого зображення, позначка водяними знаками фінальних зображень. Ідеальний варіант для тих хто хоче себе спробувати у цій сфері, або перейти з однієї програми до іншої та оцінити функціонал програми.

З основного функціоналу можна виділити:

1. Моделювання: у Maya є різні інструменти для створення 3D-моделей, наприклад багатокутне, NURBS і моделювання на підрозділи. За допомогою цих інструментів користувачі можуть створювати складні 3D-моделі, використовуючи різні техніки.

2. Анімація: Maya дозволяє створювати анімацію за допомогою ключових кадрів, кривих і доріжок. Він також має потужні інструменти для зняття кісток і шкіри, які дозволяють створювати складні анімації персонажів.

3. Текстурювання: у Maya є набір інструментів для створення текстур, включаючи інструмент для малювання текстур та інструменти для малювання моделей. Він також підтримує широкий спектр форматів текстур, включаючи PSD, JPEG і TIFF.

4. Текстурювання: у Maya є набір інструментів для створення текстур, включаючи інструмент для малювання текстур та інструменти для малювання моделей. Він також підтримує широкий спектр форматів текстур, включаючи PSD,

JPEG і TIFF.

5. Найцікавішою особливістю на мою думку буде можливість співпраці: Maya має функцію співпраці, яка дозволяє кільком користувачам працювати над одним проектом одночасно.

Це лише деякі з багатьох функцій, доступних у Maya. Завдяки своєму потужному функціоналу Maya є одним з лідерів у сфері створення 3D-графіки.

Говорячи про переваги та недоліки, почнемо з переваг:

1. Потужний набір інструментів: Maya має широкий набір інструментів для 3D-моделювання, анімації, текстурування та візуалізації.

2. Гнучкість: Maya — це гнучкий інструмент, який можна налаштувати для роботи в різних галузях, таких як кіно, телебачення, відеоігри, архітектура тощо.

3. Масштабованість: Maya може використовуватися для створення як простих моделей, так і складних сцен із великою кількістю об'єктів.

4. Підтримка сторонніх плагінів: Maya підтримує широкий спектр сторонніх плагінів, які можуть розширити її функціональність і можливості.

5. Кросплатформенність: Maya працює на різних операційних системах, таких як Windows, macOS і Linux.

Недоліки:

1. Висока вартість: Maya є досить дорогим програмним забезпеченням, яке може бути недоступним для багатьох користувачів.

2. Крута крива навчання: Maya має круту криву навчання, тому користувачеві може знадобитися деякий час, щоб оволодіти основними навичками.

3. Вимогливість до апаратного забезпечення: Maya потребує потужного апаратного забезпечення, яке може бути дорогим і недоступним для всіх користувачів.

4. Потрібні знання в інших програмах: Maya може вимагати знання інших програм, таких як Adobe Photoshop і ZBrush, що може збільшити час, необхідний для створення 3D-моделей.

5. Не підходить для всіх типів проектів: хоча Maya може використовуватися в різних галузях, це не завжди найкращий інструмент для всіх типів проектів.

Отже, Maya — це дуже потужний інструмент для створення 3D-графіки з широким набором функцій для моделювання, анімації, текстурювання та візуалізації. Вона має багато інструментів і функцій, які дозволяють створювати складні 3D-моделі, анімацію та спецефекти. Завдяки підтримці API Python користувачі можуть автоматизувати багато завдань і створювати власні інструменти та плагіни. Однак через високу вартість і високий бар'єр доступу Maya може бути складним інструментом для тих, хто тільки починає працювати з 3D-графікою. Однак для професіоналів, які прагнуть створювати високоякісну 3D-графіку та анімацію, Maya є одним із найкращих інструментів на ринку.

2. 3ds Max

3ds Max — ще один популярний комерційний 3D-пакет від Autodesk. Він використовується для створення високоякісних 3D-моделей, анімацій, візуалізацій і спецефектів для фільмів, ігор, архітектури та дизайну. [10]



Рисунок 2.2 – Приклад процесу моделювання у 3ds Max

Історія 3ds Max починається в 1990 році, коли Autodesk придбала пакет анімації 3D Studio у Yost Group. Протягом наступних кількох років Autodesk продовжувала розвивати 3D Studio, додаючи нові функції та покращуючи існуючі.

У 1996 році 3D Studio Max R1 була випущена для Windows NT, яка включала новий механізм візуалізації та інші вдосконалення. Назву змінено на "3ds Max", щоб відрізнити її від попередніх версій 3D Studio.

У наступні роки 3ds Max продовжував розвиватися, додаючи нові функції та можливості для створення складних 3D-моделей і анімацій. Версія 4.0 додала інструменти для створення рідких і газоподібних об'єктів, а версія 5.0 додала підтримку шейдерів і фрагментних програм. У пізніших версіях додано інструменти для створення складніших матеріалів, покращено інструменти моделювання та анімації, а також підтримку нових форматів файлів.

3ds Max має широкий набір функцій та інструментів для 3D-моделювання, анімації, візуалізації та спеціальних ефектів. Деякі з основних функцій 3ds Max включають:

1. Моделювання: 3ds Max надає різноманітні інструменти моделювання для створення складних 3D-моделей, включаючи багатокутне моделювання, моделювання на основі NURBS, моделювання на основі підрозділів, ліплення та інші.

2. Анімація: 3ds Max дозволяє створювати анімацію за допомогою різних інструментів, таких як ключові кадри, шляхові точки, сплайни тощо. Він також має інструменти для створення фізичних симуляцій, таких як моделювання тканин, рідин і твердих речовин.

3. Спецефекти: у 3ds Max є інструменти для створення різних спецефектів, зокрема вибухів, диму, вогню, води тощо. Він також має інструменти для створення різноманітних елементів навколишнього середовища, таких як рослини, трава та дерева.

4. API Python: Як і інші програми, 3ds Max також підтримує API Python, який дозволяє користувачам автоматизувати різні завдання та створювати власні інструменти та плагіни.

5. Сумісність: 3ds Max сумісний з багатьма іншими програмами та форматами файлів, що полегшує роботу з ним у різноманітних проектах. Він також має багато плагінів, які розширюють його функціональність.

Переваги 3ds Max:

1. Широкий набір функцій та інструментів: 3ds Max має багато інструментів для моделювання, анімації, візуалізації та спеціальних ефектів, що робить його зручним інструментом для роботи в різних галузях.

2. Потужний механізм візуалізації: 3ds Max має потужний вбудований механізм візуалізації, який дозволяє створювати високоякісні зображення та анімацію.

3. Висока швидкість роботи: 3ds Max може швидко обробляти і працювати з великими проектами.

4. API Python: 3ds Max підтримує API Python, дозволяючи користувачам створювати власні інструменти та плагіни, а також автоматизувати різні завдання.

Недоліки:

1. Висока вартість: 3ds Max є одним із найдорожчих інструментів 3D-моделювання та анімації, що може відштовхнути новачків, фрілансерів або невеликі студії.

2. Високі вимоги до комп'ютера: для виконання великих проектів і складних завдань 3ds Max потрібен потужний комп'ютер.

3. Складність у використанні: 3ds Max має досить складний інтерфейс, і може знадобитися багато навчання та практики, щоб повністю опанувати його.

4. Обмежена підтримка інших мов програмування: хоча 3ds Max підтримує Python, підтримка інших мов програмування, таких як C++, може бути обмежена.

Загалом 3ds Max надає потужний і гнучкий набір інструментів для створення високоякісної 3D-графіки, але він може бути несумісним з деякими проектами через високу вартість і складність використання.

3. Cinema 4D



Рисунок 2.3 - Інтерфейс Сінема 4D

Сінема 4D — це програма для 3D-моделювання, анімації та візуалізації, розроблена німецькою компанією Maxon Computer GmbH. Сінема 4D вперше було випущено в 1990 році для комп'ютерів Amiga. Після того, як Maxon переключився на розробку для ПК і Mac, Сінема 4D став одним із найпопулярніших програм для 3D-графіки в галузі.[11]

Сьогодні Сінема 4D широко використовується для створення різних видів тривимірної графіки, включаючи анімацію, спецефекти, рекламу, фільми, телешоу тощо. Програма надає широкий набір інструментів для моделювання, анімації, текстурування та освітлення, що робить її зручною для роботи в різних галузях.

Основні функції Сінема 4D:

1. Моделювання: Сінема 4D має інтуїтивно зрозумілий і потужний набір інструментів для створення 3D-моделей, включаючи інструменти для роботи з багатокутниками, точками та кривими.
2. Анімація: Сінема 4D дозволяє створювати складну 3D-анімацію за допомогою таких інструментів, як ключові кадри, сплайни, криві та динаміка.
3. Освітлення: Сінема 4D дозволяє створити складну систему освітлення для

3D-сцен, включаючи використання різних типів джерел світла та налаштувань освітлення.

Дивлячись на переваги, найважливішим буде:

1. Інтуїтивно зрозумілий інтерфейс: Cinema 4D має простий та інтуїтивно зрозумілий інтерфейс, який полегшує початківцям розпочати роботу.

2. Широкий набір функцій: Cinema 4D має великий набір інструментів для моделювання, текстурювання, анімації, візуалізації та багатьох інших завдань, що робить його дуже гнучким і простим у роботі.

3. Сумісність: Cinema 4D може імпортувати та експортувати файли в різних форматах, таких як FBX, OBJ, Alembic та інших, що дозволяє вам взаємодіяти з іншими програмами та використовувати його в різних проектах.

4. Рендеринг: Cinema 4D має власний вбудований рендерер під назвою Physical Renderer, який дозволяє отримувати високоякісні зображення та анімацію, а також підтримує інтеграцію з іншими рендерерами, такими як Arnold і Octane Render.

Недоліки:

1. Ціна: Cinema 4D не є безкоштовним програмним забезпеченням і має досить високу ціну, що може зробити його недоступним для початківців користувачів і малого бізнесу.

2. Не підтримує деякі функції іншого програмного забезпечення: деякі користувачі можуть виявити, що Cinema 4D не підтримує деякі функції, доступні в іншому програмному забезпеченні 3D.

3. Обмежені можливості в деяких областях: деякі користувачі можуть виявити, що Cinema 4D має обмежені можливості в деяких областях, наприклад симуляції рідини та твердого тіла, для виконання яких може знадобитися використання додаткового програмного забезпечення.

4. Потрібне хороше апаратне забезпечення: Cinema 4D є досить ресурсомістким програмним забезпеченням і потребує потужного комп'ютера для виконання великих проектів.

4. Zbrush

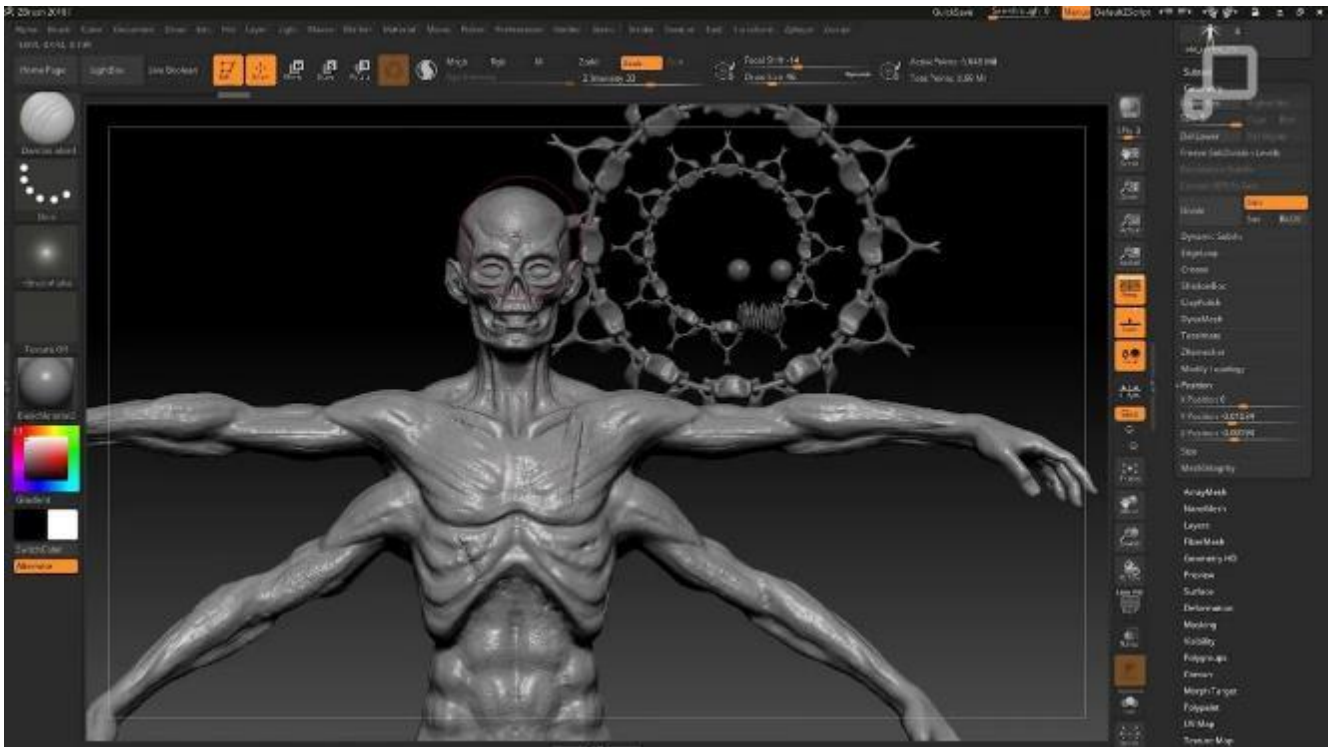


Рисунок 2.4 - Процес скульптингу у Zbrush

ZBrush — це цифрове програмне забезпечення для ліплення та малювання, яке дозволяє художникам створювати високодеталізовані 3D-моделі за допомогою інтуїтивно зрозумілого інтерфейсу та потужних інструментів.

Історія ZBrush почалася в 1999 році, коли американський художник Орест Дрейпер почав розробляти власний цифровий інструмент для ліплення. У 2000 році Орест приєднався до Pixologic для подальшого розвитку свого інструменту.

Перша версія ZBrush була випущена в 2001 році, і з тих пір програма продовжує розвиватися і вдосконалюватися. ZBrush став одним із найпопулярніших цифрових інструментів для ліплення та малювання завдяки своїй потужності та гнучкості, а також здатності працювати з високим рівнем деталізації.

Сьогодні ZBrush використовується в різних галузях, включаючи ігри, кіно та анімацію, а також для створення прототипів і дизайну продуктів у цій галузі.[12]

Основні функції ZBrush включають:

1. Інтуїтивно зрозумілий інтерфейс і зручне управління інструментами.
2. Потужні цифрові інструменти для ліплення, включаючи інструменти для текстурювання, малювання та полірування.

3. Вміння працювати з високодеталізованими моделями та створювати реалістичні ілюстрації та анімацію.

4. Можливість інтеграції з іншими програмами цифрового моделювання та візуалізації, такими як Maya та Blender.

Дивлячись на переваги Zbrush:

1. Інтуїтивно зрозумілий інтерфейс: ZBrush має інтуїтивно зрозумілий інтерфейс, який робить його доступним навіть для початківців. Інструменти розташовані в логічній послідовності, що дозволяє швидко знайти потрібні функції та прискорити робочий процес.

2. Потужні цифрові інструменти для ліплення: ZBrush надає величезний набір цифрових інструментів для ліплення, які дозволяють створювати високодеталізовані 3D-моделі. Інструменти дозволяють працювати з формами, текстурами, пензлями і багатьма іншими елементами, що робить процес створення моделей більш продуктивним і ефективним.

3. Високоякісний результат: ZBrush дозволяє створювати високоякісні 3D-моделі з високим рівнем деталізації. Це дозволяє художникам створювати більш реалістичні та детальні моделі, що, у свою чергу, покращує якість візуальних зображень та анімації.

4. Гнучкість роботи: ZBrush дозволяє вам працювати з високим рівнем гнучкості. У програмі є функція збереження історії роботи, яка дозволяє змінювати попередні дії та вносити зміни в модель без порушення її цілісності.

Недоліками Zbrush будуть:

1. Висока вартість: ZBrush є комерційним програмним забезпеченням, що робить його досить дорогим для деяких користувачів. Незважаючи на це, компанія надає можливість користуватися пробною версією програми протягом певного часу, що буде великим плюсом для початківців та фрілансерів.

2. Вимогливий до комп'ютерних ресурсів: для роботи ZBrush потрібен потужний комп'ютер, особливо під час роботи з високодеталізованими моделями. Деякі користувачі можуть зіткнутися з проблемою відсутності необхідного обладнання для роботи з програмою.

3. Необхідність навчання: ZBrush має багато функцій та інструментів, які можуть заплутати нових користувачів. Це може зайняти деякий час

Ще одним недоліком ZBrush є бажане використання для роботи в програмі графічного планшета. Незважаючи на те, що ви можете працювати за допомогою миші, використання графічного планшета дозволяє точніше і природніше моделювати 3D-об'єкти, оскільки дозволяє передавати багаторівневий натиск пензля і кут нахилу, що полегшує роботу художника і підвищує точність. Без графічного планшета художнику може бути складніше отримати бажаний результат.

2.2. Огляд середовища розробки Blender

Blender — це безкоштовне програмне забезпечення для 3D-моделювання, анімації та візуалізації з відкритим кодом. Він був розроблений у Нідерландах у 1995 році в рамках проекту NeoGeo, який потім був перейменований у Not a Number Technologies (NaN). Blender був створений як комерційний продукт, але в 2002 році він став вільно розповсюджуватися під GNU General Public License.[13]

З моменту створення Blender пройшов довгий шлях від простого інструменту 3D-моделювання до повноцінного інструменту CG та анімації. Він набув широкого поширення в колах аніматорів, художників, архітекторів і розробників ігор.

В даний час Blender активно використовується в різних сферах, таких як архітектура, дизайн, мультимедіа та наукова візуалізація. Це також популярний інструмент в індустрії анімації та розробки відеоігор.

Особливістю Blender є його здатність працювати з багатьма форматами файлів, що дозволяє легко обмінюватися даними між різними програмами. Він також підтримує багатопотоковість, що дозволяє використовувати всі ядра процесора для прискорення виконання обчислювальних завдань.



Рисунок 2.5 - Головний інтерфейс програми

Blender має багато інструментів для створення різноманітних об'єктів, анімації, текстурювання, освітлення, рендерингу та багато іншого. Його інтерфейс можна налаштувати відповідно до індивідуальних потреб користувача, що робить його гнучким і простим у використанні.

Крім того, Blender має активну спільноту користувачів і розробників, які постійно працюють над удосконаленням програми. Вони створюють додаткові модулі, скрипти та плагіни, які розширюють функціональні можливості програми та полегшують її використання.

Однією з найважливіших особливостей Blender є його можливості 3D-моделювання. Blender дозволяє створювати 3D-моделі будь-якої складності, від простих геометричних фігур до складних органічних об'єктів. Існує багато інструментів, доступних для моделювання, таких як пензлі, курси, набори інструментів тощо.

Ви можете створювати анімації будь-якої складності, від простих переходів між кадрами до складних рухів персонажів. Blender надає широкий спектр можливостей для роботи з ключовими кадрами, кривими анімації, параметрами швидкості тощо.[14]

Blender також має потужні можливості текстурювання та освітлення. За допомогою Blender ви можете створювати реалістичні текстури, регулювати освітлення сцени, створювати тіні та відображення, а також додавати спеціальні ефекти, такі як дим, вогонь і вода.



Рисунок 2.6 - Spring, мультфільм зроблений у Blender

Не можна забувати і про візуалізацію даних. За допомогою Blender ви можете створювати візуалізації даних, такі як графіки, діаграми та графіки, які можна використовувати в різних наукових та інженерних сферах.

Крім того, Blender є кросплатформним, що означає, що його можна використовувати в будь-якій операційній системі, включаючи Windows, Mac і Linux. Це робить Blender ідеальним інструментом для командної роботи та співпраці.

Як вже було згадано, перша і, мабуть, найважливіша перевага Blender — це його відкритий код і безкоштовне розповсюдження. Це означає, що кожен може завантажити та використовувати Blender без обмежень або додаткових витрат. Крім того, будь-хто може змінювати та доповнювати Blender, що робить його

неймовірно гнучким та адаптованим до потреб кожного користувача.

Але, з іншої сторони, це може бути і головним недоліком:

По-перше, багато роботодавців вважають за краще використовувати платні програми, такі як Autodesk Maya, 3ds Max або Cinema 4D, які мають більший функціонал і підтримку, а також інтеграцію з іншими інструментами, що використовуються в галузі. Роботи, які вимагають використання таких програм, можуть не згадувати Blender, навіть якщо Blender може бути ефективним інструментом для вирішення поставлених завдань.[15]

По-друге, деякі роботодавці можуть вважати, що Blender недостатньо популярний, щоб його вказували в оголошеннях про роботу. Однак із зростанням популярності Blender і збільшенням кількості користувачів його згадка в оголошеннях про роботу може стати більш поширеною.

Проте Blender все ще залишається потужним і гнучким інструментом для створення комп'ютерної графіки та анімації, який широко використовується в індустрії. Багато невеликих студій і фрілансерів вибирають Blender як безкоштовну та ефективну альтернативу платним програмам.

Згадуючи про обрану тему, створення плагінів у Blender — це процес, який може надати дивовижні можливості для творчого зростання та вдосконалення функціональності програми. Blender — це відкрита платформа, яка забезпечує зручний і гнучкий інтерфейс для створення плагінів, які можуть значно розширити можливості програми.

Однією з головних переваг створення плагінів у Blender є доступність і зручність API (Application Programming Interface). API Blender надає широкий спектр інструментів і функцій для створення плагінів, включаючи доступ до моделей, матеріалів, світла тощо. Крім того, Blender має вбудований інтерпретатор Python, який дозволяє створювати плагіни за допомогою Python, що робить процес створення плагінів швидким і легким.

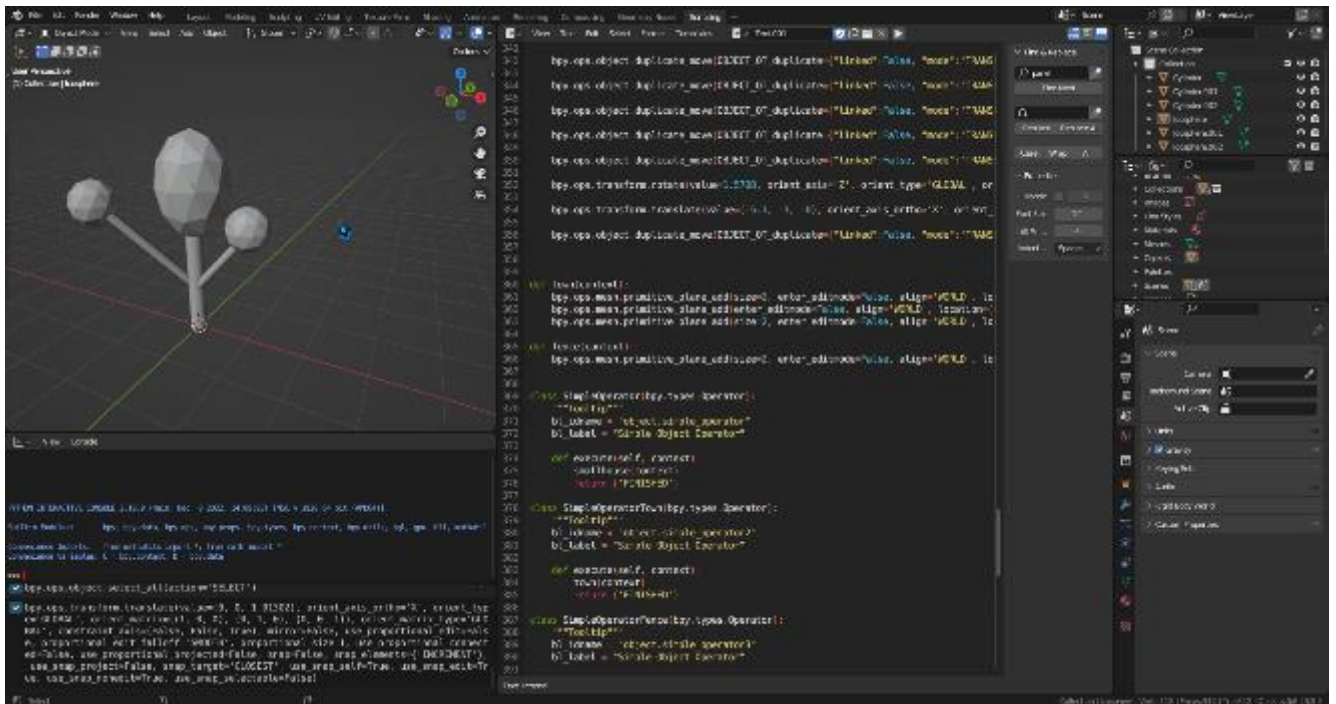


Рисунок 2.7 - Розробка плагіну у Blender

Ще однією перевагою створення плагінів у Blender є активна спільнота, яка підтримує та розробляє Blender та його плагіни. Наявність великої кількості розробників, дизайнерів і художників у спільноті Blender дозволяє швидко отримати допомогу та поради щодо створення плагінів, а також поширювати та просувати свої плагіни серед спільноти.

Крім того, Blender має відкриту архітектуру та вихідний код, що дозволяє розробникам створювати плагіни, які можна інтегрувати з іншими програмами та інструментами. Це робить Blender та його плагіни універсальними та гнучкими інструментами для створення комп'ютерної графіки та анімації.

Загалом, створення плагінів у Blender — веселий та продуктивний процес, який надає можливості для творчого зростання та вдосконалення функціональності програми. Blender — ідеальне середовище для створення плагінів, які можуть значно розширити можливості програми та допомогти вам досягти бажаних результатів.

2.3. Вибір мови програмування

Обираючи мову програмування для створення плагіна Blender, слід враховувати кілька факторів, наприклад продуктивність, доступність API, стабільність і підтримку спільноти. Зараз ми розглянемо декілька з них:

По-перше, продуктивність мови програмування є важливим фактором для плагінів, які виконують складні обчислення або операції з великими обсягами даних. У цьому випадку використання таких мов, як C++ або Rust, може забезпечити кращу продуктивність, ніж інтерпретовані мови, такі як Python або Lua.[16]

По-друге, доступність API Blender також є важливим фактором при виборі мови програмування. Blender надає API для Python, C++ і Lua, що робить їх найбільш підходящими мовами для створення плагінів. Мови, які не мають повної підтримки API Blender, можуть мати обмежений доступ до деяких функцій.

По-третє, стабільність мови програмування також є важливим фактором при виборі мови для створення плагіна. Мови, які мають велику спільноту та довгу історію розробки, такі як Python і C++, можуть забезпечити більш стабільну роботу плагінів, ніж мови, які тільки починають набирати популярність.

По-четверте, підтримка спільноти також може бути важливим фактором при виборі мови програмування. Мови, які мають велику спільноту розробників, забезпечують більший доступ до таких ресурсів, як документація та бібліотеки. Це також забезпечить швидше вирішення проблем, пов'язаних із плагінами.

Розглянемо кожен з аналогів:

1. Python є найпоширенішою мовою програмування для створення плагінів для Blender. Його легко освоїти, він має велику документацію та підтримку спільноти.

2. C++ — ще одна мова програмування, яку можна використовувати для створення плагінів для Blender. Він забезпечує кращу продуктивність, ніж Python, але має вищий поріг входу.

3. Lua — ще одна мова програмування, яку підтримує Blender. Він невеликий

і простий у використанні, але може мати обмежений доступ до деяких функцій API Blender.

4. Java - хоча Java не є офіційно підтримуваною мовою для створення плагінів для Blender, ви можете використовувати JNI (Java Native Interface) для спілкування між Java і C++ для створення плагінів.

Крім того, інші мови програмування можна використовувати в Blender через зовнішні бібліотеки, такі як C#, JavaScript і Rust.

Проаналізувавши усі варіанти, було обрано мову Python.

Python — це інтерпретована мова програмування високого рівня, створена в 1991 році Гвідо ван Россумом. Python швидко став однією з найпопулярніших мов програмування завдяки своїй простоті, читабельності та легкості вивчення.

```
def insertion_sort(array):
    # Перебираємо елементи масиву починаючи з другого
    for i in range(1, len(array)):
        # Запам'ятовуємо поточне значення, щоб не втратити його при зсуві
        current_value = array[i]
        # Перебираємо всі попередні елементи масиву
        j = i - 1
        while j >= 0 and array[j] > current_value:
            # Зсуваємо елементи вправо, щоб створити місце для поточного значення
            array[j+1] = array[j]
            j -= 1
        # Вставляємо поточне значення на правильну позицію
        array[j+1] = current_value
    return array

# Приклад використання
my_array = [5, 2, 8, 1, 3]
sorted_array = insertion_sort(my_array)
print(sorted_array)
```

Рисунок 2.8 - Приклад синтаксису Python

Однією з головних переваг Python є його широке впровадження в галузі та велика спільнота розробників, яка надає обширну документацію та багату

екосистему бібліотек і фреймворків. Blender використовує Python як основну мову для написання плагінів і сценаріїв.

Особливості Python:

1. Простий і легкий у вивченні: Python має простий і зрозумілий синтаксис, який полегшує його вивчення та використання навіть для розробників-початківців.

2. Інтерпретований: Python не потрібно компілювати, він інтерпретується, що дозволяє швидко розробляти код і тестувати.

3. Кросплатформенність: Python може працювати в різних операційних системах, таких як Windows, macOS і Linux.

4. Розширена стандартна бібліотека: Python має багату стандартну бібліотеку, яка включає багато корисних модулів та інструментів розробки.

5. Підтримка об'єктно-орієнтованого програмування: Python повністю підтримує об'єктно-орієнтоване програмування, що полегшує розробку великих і складних програм.

6. Інтеграція з іншими мовами: Python можна легко інтегрувати з іншими мовами програмування, такими як C++ або Java, дозволяючи вам скористатися найкращими функціями та можливостями кожної мови.

Python також має багато бібліотек, які можна використовувати в Blender, наприклад NumPy, SciPy, Matplotlib тощо. Ці бібліотеки дозволяють працювати з великими обсягами даних, науковими обчисленнями, візуалізацією тощо.

Ідеальна сумісність Python із Blender робить його чудовим вибором для створення плагінів і сценаріїв у Blender. Завдяки цьому розробники можуть використовувати багатий вибір бібліотек Python для вирішення різних проблем і покращення продуктивності та функціональності Blender.

2.4. Алгоритм роботи додатку

Робота з плагіном починається із запуску програми Blender. При першому запуску інтерфейс буде виглядати як на рисунку 2.9.

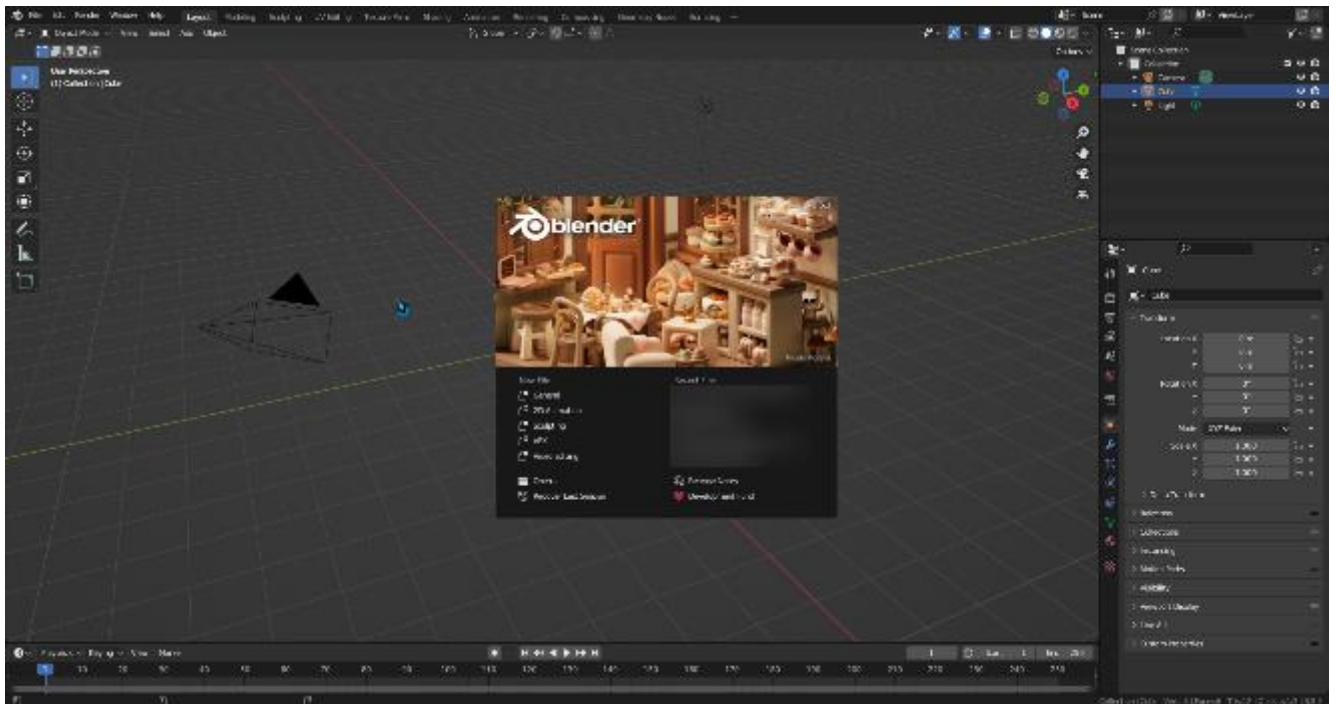


Рисунок 2.9 - Вигляд меню при першому запуску Blender

Наступним кроком буде відкриття вкладки Preferences, знаходження розділу Add-ons та встановлення плагіну.

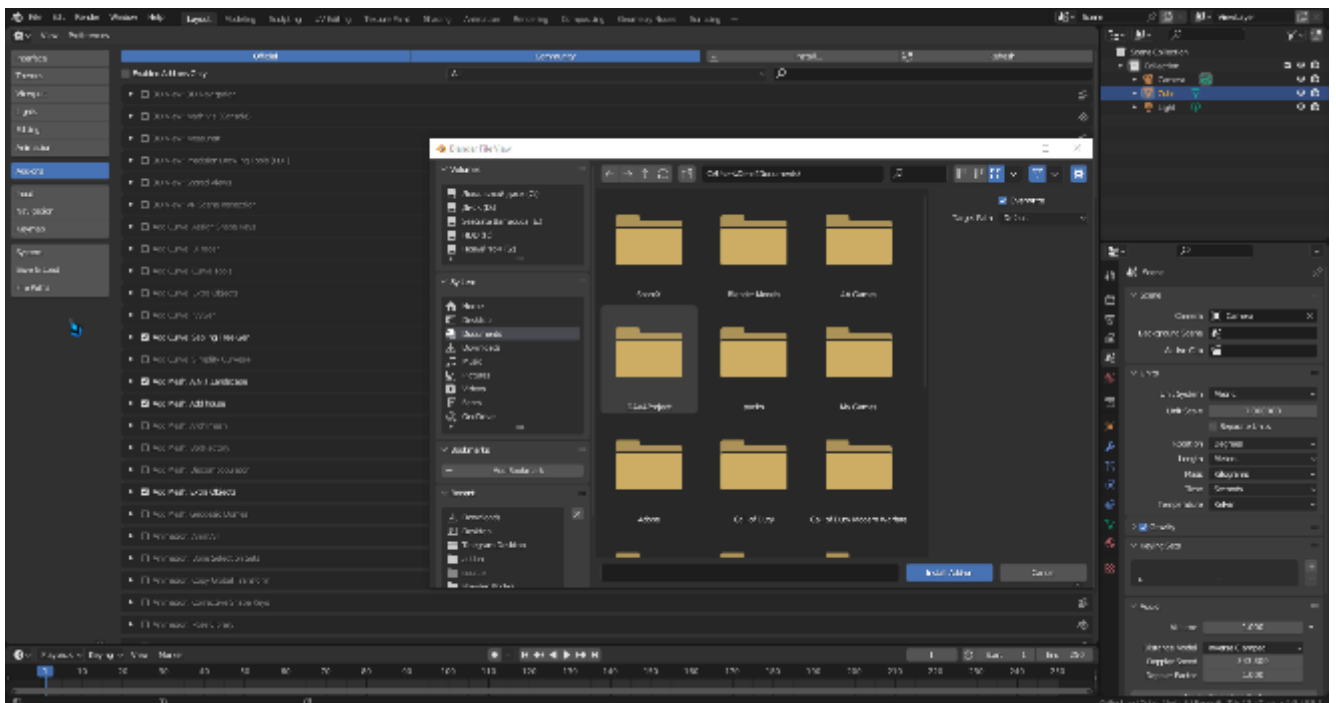


Рисунок 2.10 - Розділ встановлення плагінів

Після встановлення, плагін буде додано до панелі Scene, після чого його можна буде використовувати.

Щоб краще передати суть даного проекту можна подати його через наступні діаграми:

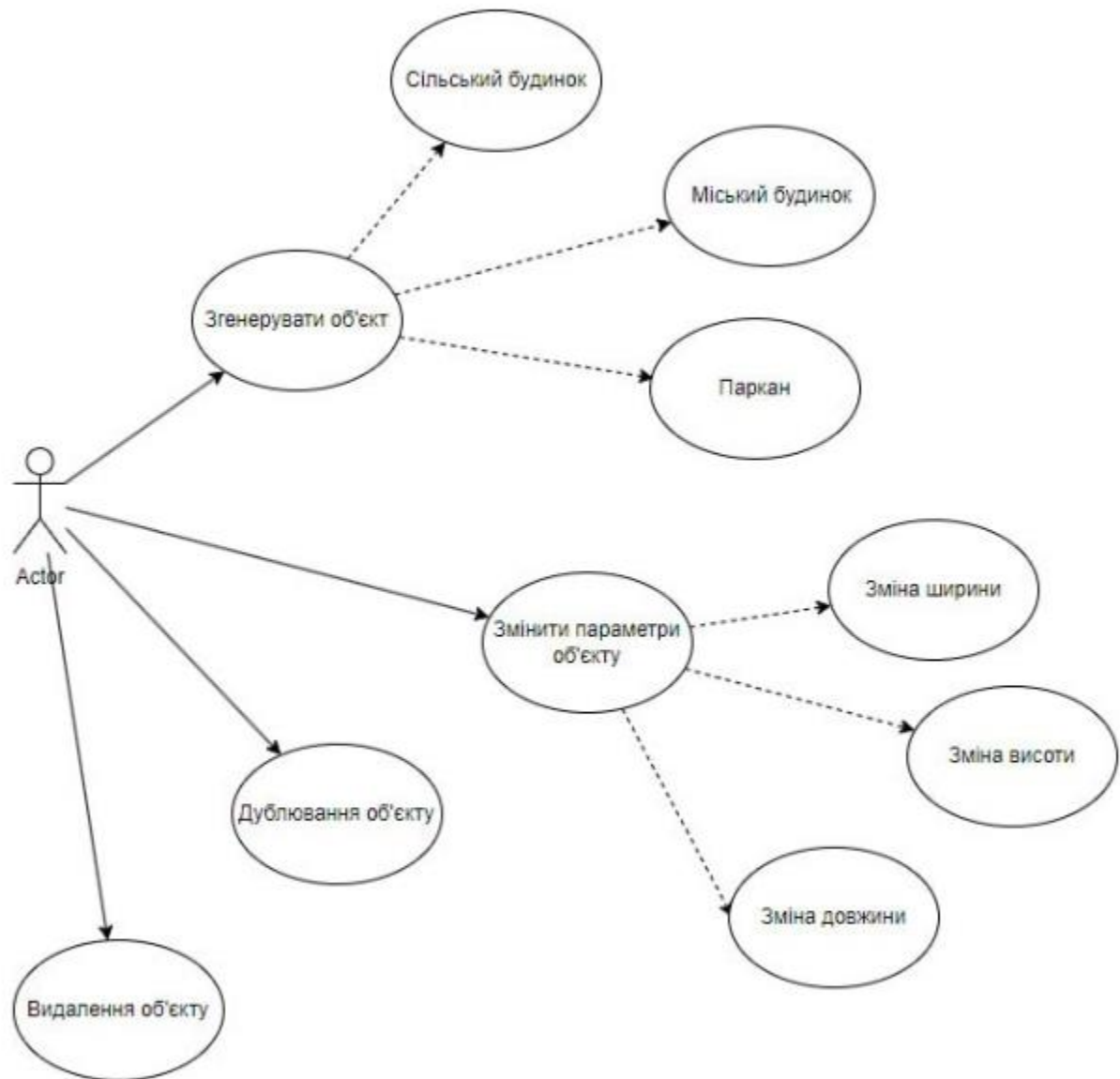


Рисунок 2.11 - Діаграма діяльності

Діаграма діяльності – це наочне уявлення послідовності процесів у графічному вигляді. Він містить вершини, які мають правильні і певні дії, а

переходи з-поміж них відбуваються після завершення дій. Діаграма діяльності – це тип діаграми станів, де кожен стан може включати виконання операції частини чи класу. При розробці складних проектів дуже важливим стає вибір оптимальних рішень. Кожна діаграма дій має початок і кінець, і всі дії, що виконуються, повинні йти зверху вниз. У діаграмі діяльності можна використовувати лише нетригерні переходи. Розгалуження на діаграмі діяльності представлено невеликим ромбом без тексту всередині. Важливим застосуванням діаграм діяльності є їхнє моделювання різних бізнес-процесів.

Наступною йде діаграма стану:

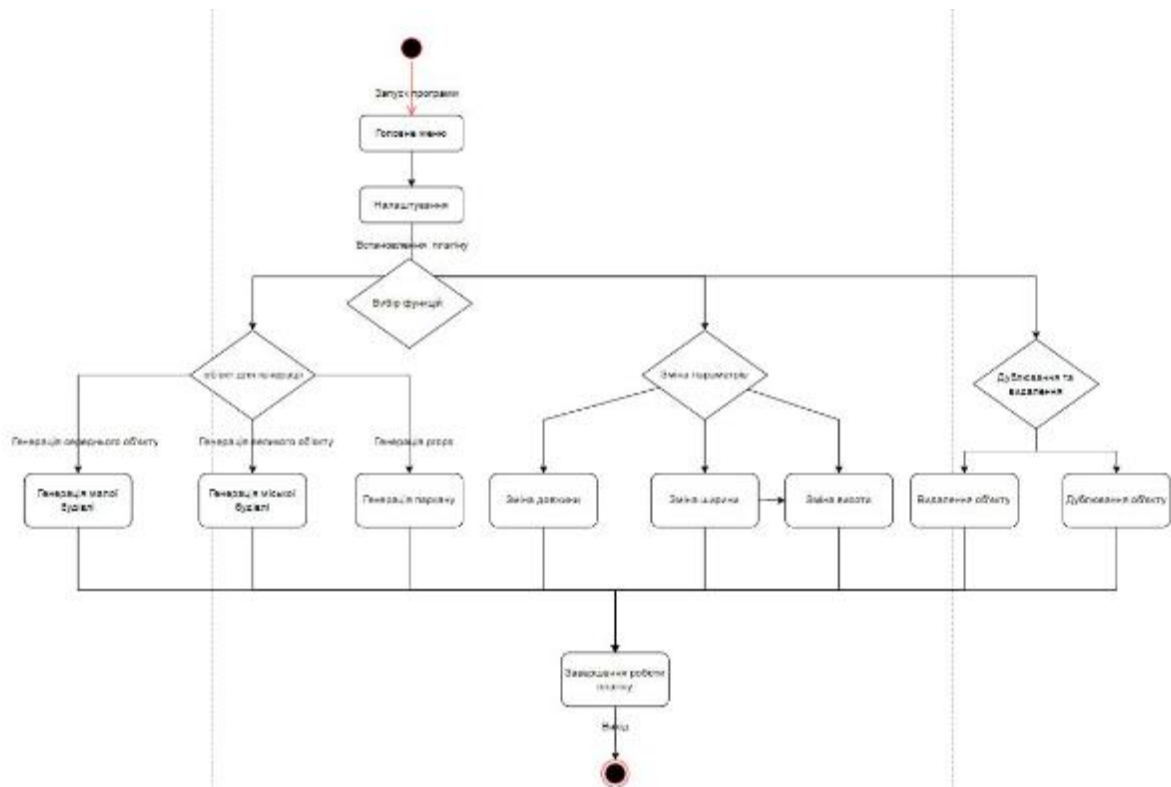


Рисунок 2.12 - Діаграма стану

Діаграма стану - це графічне уявлення стану системи, що складається з об'єктів та їх взаємодій. Ця діаграма дозволяє наочно уявити різні стани системи та переходи між ними. Діаграма станів може бути корисною при проектуванні та розробці програмного забезпечення, систем управління та інших систем зі станом. Зокрема, діаграма стану дозволяє визначити:

1. Стани системи та їхні властивості.

2. Події, що призводять до переходу між станами.

3. Дії, що відбуваються при переході між станами.

Діаграма станів є частиною узагальненої моделі системи (UML), яка є стандартом проектування програмного забезпечення. Це означає, що діаграма станів може використовуватися для опису системи, що розробляється, і для обміну загальним розумінням роботи системи між розробниками і замовниками.

Наразі плагін має декілька функцій, розглянемо кожен з них:



Рисунок 2.12 - Інтерфейс плагіну

1. Add Small House, при натисканні на першу кнопку, буде створено маленький сільський будинок. Завдяки розподілу на окремі об'єкти, модель стає модульною, завдяки чому можна усього за кілька дій змінити форму дома, роблячи майбутню сцену більш різноманітною.

2. Add Town House, друга кнопка створює будинок більш схожий на міський, який більше підходить для створення оточення міста. Як і перша модель, цей будинок також є модульним, що дозволяє легко додати або видалити додатковий поверх та змінити форму додавши ще одну основу.

3. Add Fence, остання кнопка створює паркан, який можна комбінувати з будівлями для створення заднього двору або інших ідей, все обмежується лише уявою автора.

4. Редагування розмірів створенні моделі.

5. Дублювання створенної моделі.

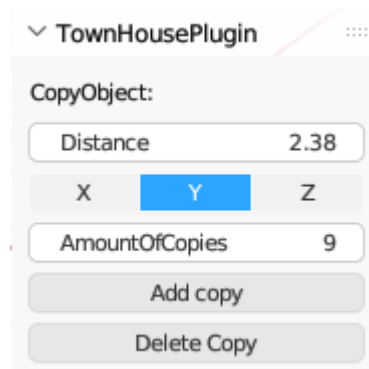


Рисунок 2.13 – Інтерфейс меню для дублювання

Нижче наведено приклад модульності одного з будинків, та як це можна використовувати для різноманітності у сцені



Рисунок 2.14 - Різний вигляд одного будинку

3. РОЗРОБКА ПЛАГІНУ

3.1. Загальний огляд засобів для розробки

Створення плагіна для Blender включає ряд етапів та використання різних інструментів та засобів розробки. У процесі розробки плагіна для Blender застосовуються такі методи та інструменти:

1. Мови програмування:

Вибір мов програмування для створення плагінів досить великий, але в даному випадку був обраний Python, оскільки він найкраще синергує з Blender.

2. Blender API: Blender має спеціальний інтерфейс програмування програм (API), що забезпечує доступ до функціональних можливостей програми.

3. Інтегроване середовище розробки (IDE): Для розробки плагінів у Blender можна використовувати потрібне інтегроване середовище розробки (IDE) для написання Python-коду. З них можна виділити PyCharm, Visual Studio Code, Sublime Text та інші.

4. Редактор тексту Blender: У Blender також передбачено вбудований редактор тексту, який полегшує написання коду плагіна прямо всередині програми.

5. Створення інтерфейсу користувача: В Blender є можливість створювати інтерфейс (UI) для плагінів. Для створення плагінів можна використовувати інструменти, такі як бібліотека PyQT або вбудований модуль bpy.types у Blender, для створення інтерактивних елементів керування, включаючи кнопки, поля введення, меню та ін.

У наступних розділах буде детальніше розглянуто кожен із засобів розробки.

3.2. Blender API

Blender API — це програмний інтерфейс, який надає розробникам доступ до функцій програми Blender. Blender API — це потужний інструмент, який дозволяє створювати плагіни, сценарії та розширювати можливості Blender.[17]

Архітектура Blender API базується на об'єктно-орієнтованому підході. Він

пропонує різні модулі та класи, які розробники можуть використовувати для взаємодії з різними аспектами Blender, такими як моделювання, анімація, візуалізація тощо.

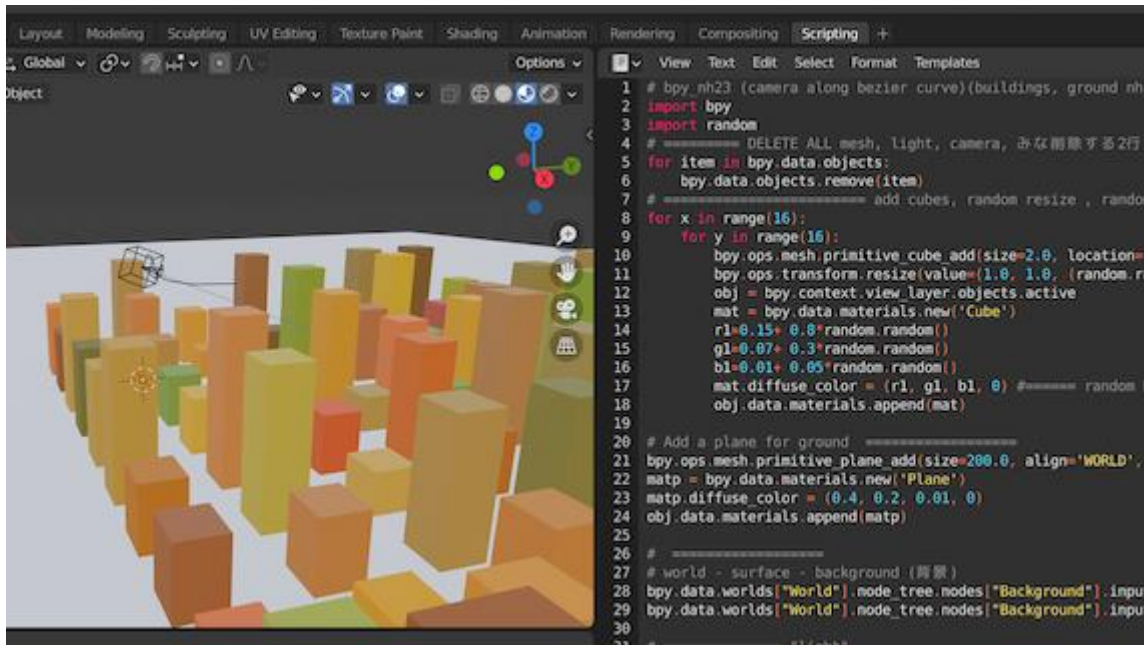


Рисунок 3.1 – Програмування з використанням Blender API

Бібліотека Blender API повністю реалізована на мові програмування Python. Python було обрано як основну мову для розробки плагінів і сценаріїв для Blender через його простоту, популярність і широке використання в спільноті розробників.

Основні компоненти Blender API включають:

1. Модуль bpy: Цей модуль є основним інтерфейсом API Blender. Він забезпечує доступ до різноманітних функцій Blender, таких як робота з об'єктами, матеріалами, сценами, анімацією тощо. Модуль bpy також містить класи, які описують різні об'єкти та елементи в Blender.
2. Класи та методи: Blender API пропонує багато класів і методів, які дозволяють вам програмно керувати різними аспектами Blender. Класи представляють такі сутності, як об'єкти, матеріали, камери, освітлення тощо, а методи дозволяють виконувати операції над цими сутностями, наприклад створювати, змінювати, видаляти тощо.
3. Маніпулювання даними: API Blender надає можливість маніпулювати

такими даними, як вершини, ребра, багатокутники та текстури. Розробники можуть використовувати API для читання та запису даних моделі, а також для виконання різних геометричних операцій, таких як перетворення, створення та редагування сіток та інших елементів сцени.

4. Робота з матеріалами та текстурами: API Blender надає можливість створювати, редагувати та застосовувати матеріали та текстури. Розробники можуть контролювати такі властивості матеріалу, як колір, прозорість, відображення та інші параметри, а також налаштовувати текстури та їх координати.
5. Анімація та тимчасові дані: API Blender дозволяє програмувати анімацію та працювати з тимчасовими даними в Blender. Розробники можуть створювати ключові кадри, керувати кривими анімації, налаштовувати параметри часу та шкали часу та виконувати інші операції, пов'язані з анімацією.
6. Взаємодія інтерфейсу користувача: API Blender надає можливість створювати інтерфейс користувача (UI) для плагінів. Розробники можуть створювати та налаштовувати елементи керування, такі як кнопки, поля введення, меню та інші інтерактивні компоненти, щоб полегшити взаємодію користувача з плагінами.

Blender API надає багатий набір функцій і можливостей, які дозволяють розробникам повністю контролювати та розширювати Blender відповідно до своїх потреб. Детальна документація та ресурси для API Blender доступні на офіційному веб-сайті Blender, надаючи поглиблені знання та використання API.

3.3 Python API

Python API для Blender — це інтерфейс прикладного програмування, наданий Blender, провідним програмним забезпеченням для 3D-моделювання та анімації. API Python надає розробникам можливість розширювати та налаштовувати функціональні можливості Blender шляхом написання сценаріїв і плагінів мовою програмування Python.

Основою для роботи з Python API в Blender є модуль bpy. Модуль bpy надає вам доступ до всіх функцій Blender, включаючи створення та редагування об'єктів, роботу з матеріалами та текстурами, керування анімацією та камерами та багато інших функцій.

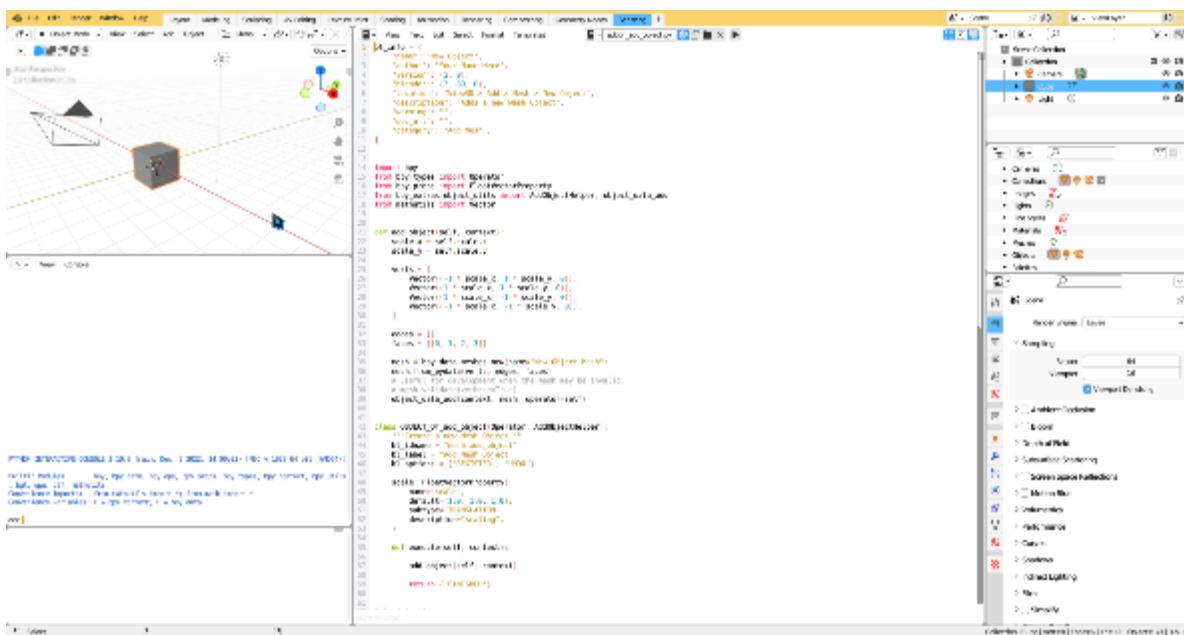


Рисунок 3.2 – Створення об'єкту за допомогою плагіна

API Python у Blender має розширену документацію, яка включає описи класів, методів, властивостей і приклади використання. Ця документація доступна на офіційному веб-сайті Blender, надаючи програмістам усе необхідне для розробки плагінів і сценаріїв, які налаштовують Blender відповідно до їхніх потреб.[18]

За допомогою API Python розробники можуть створювати інструменти для автоматизації завдань, розширювати функціональні можливості Blender,

створювати інтерфейс користувача, інтегрувати Blender з іншими програмами тощо. API Python є одним із ключових факторів, які роблять Blender гнучким і розширюваним інструментом для створення 3D-контенту.

Однією з переваг використання API Python для Blender є широка підтримка та активна спільнота розробників, які діляться знаннями, досвідом і створюють багато бібліотек і розширень для Blender на основі API Python.

Загалом, Python API Blender є потужним інструментом для розробників, щоб впроваджувати інновації та розширювати можливості Blender для задоволення унікальних потреб і викликів користувачів.

3.4 Процес розробки плагіну

Розробка плагіна почалася з імпорту бібліотеки bpy, другим кроком було моделювання, кожна дія якого запам'ятовувалася та виводилося у вигляді коду. Далі було створено функцію `smallhouse`, куди було перенесено всі дії при моделюванні у вигляді коду.

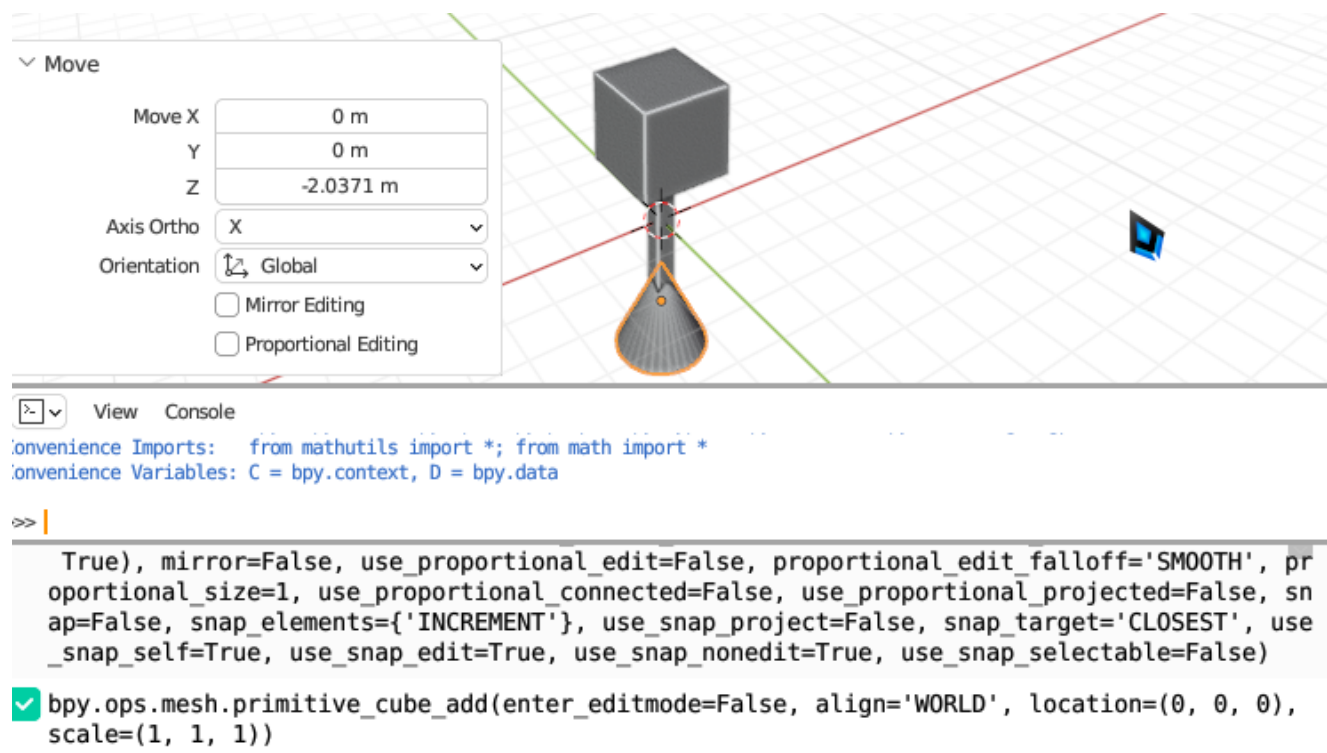


Рисунок 3.3 – Кожна дія при моделюванні переводиться в код

Далі був створений оператор `SimpleOperator`, котрий під час виклику

виконуватиме функцію `smallhouse`, яка у свою чергу створить першу модель у плагіні.

```

691 |
692 class SimpleOperator(bpy.types.Operator):
693     """Tooltip"""
694     bl_idname = "object.simple_operator"
695     bl_label = "Simple Object Operator"
696
697     def execute(self, context):
698         smallhouse(context)
699         return {'FINISHED'}
700

```

Рисунок 3.4 – клас `SimpleOperator`

Так само як і з першою моделлю, були створені інші моделі та функції з класами для них. Закінчивши з цим, наступним кроком буде створення інтерфейсу для взаємодії з плагіном.

```

class PanelOne(bpy.types.Panel):
    """Creates a Panel in the scene context of the properties editor"""
    bl_label = "Town maker addon"
    bl_idname = "SCENE_PT_panel_one"
    bl_space_type = 'PROPERTIES'
    bl_region_type = 'WINDOW'
    bl_context = "scene"

    def draw(self, context):
        layout = self.layout

        # Create some UI elements for panel one
        layout.label(text="Choose your model")
        layout.operator("object.simple_operator", text="Add Small House")
        layout.operator("object.simple_operator2", text="Add Town House")
        layout.operator("object.simple_operator3", text="Add Fence")

```

Рисунок 3.5 – Створення інтерфейсу

Цей код створює інтерфейс, який відобразиться в редакторі властивостей Blender, пропонуючи користувачеві вибрати модель і додати відповідні об'єкти в сцену за допомогою операторів "Add Small House", "Add Town House" і "Add Fence".

В результаті розробки був створений плагін, що створює повноцінні моделі в один натиск:

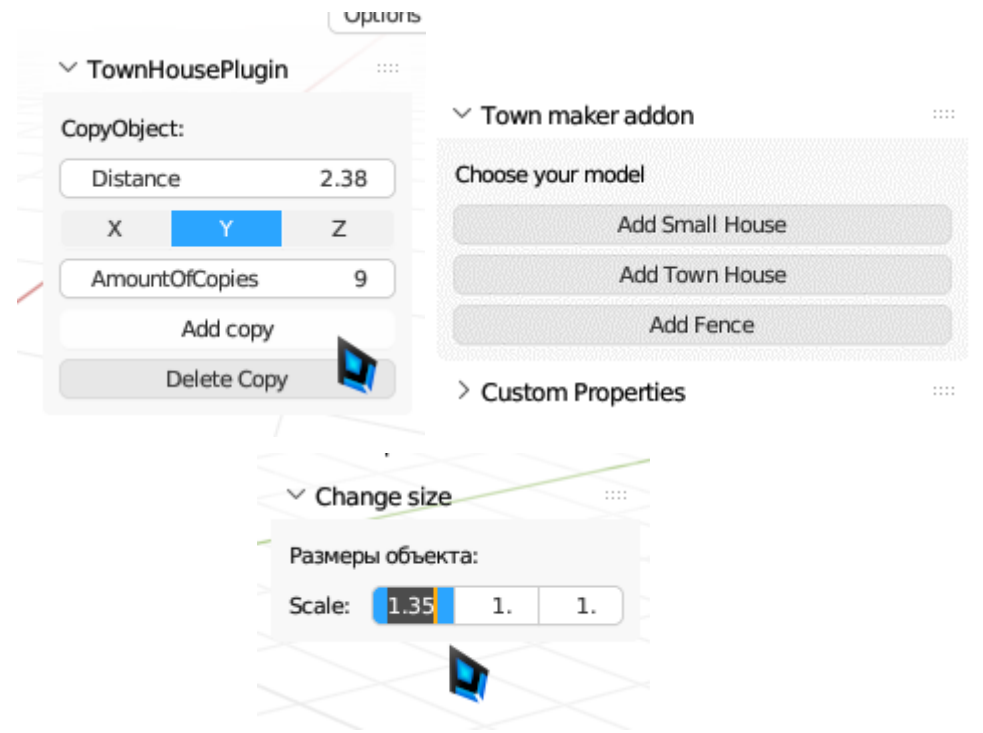


Рисунок 3.6 – Інтерфейс плагіну

В результаті розробки був створений плагін, що створює повноцінні моделі в один натиск, та дозволяє редагувати його характеристики:

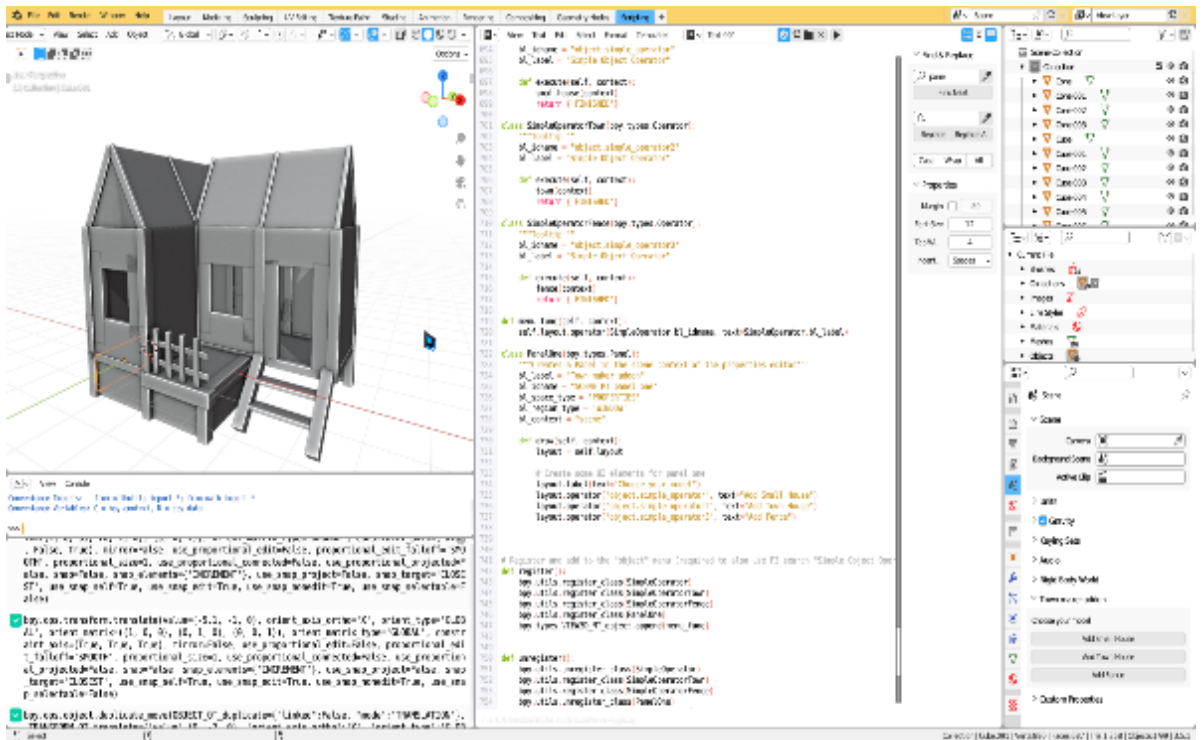


Рисунок 3.7 – Приклад роботи плагіну

ВИСНОВКИ

Згідно результатів даної дипломної роботи було розроблено плагін для спрощеного створення 3D моделей мовою Python з використанням Blender API.

В ході виконання дипломної роботи було виконано наступні ключові пункти:

1. Проведено аналіз галузі розробки плагінів для спрощеного створення 3D моделей. Її актуальність та ключові проблеми, які можна виділити. Визначено основну ключову аудиторію.

2. Розглянуто аналоги плагіну, які можуть бути використано для спрощення створення 3D моделей. Ключовими недоліками існуючих систем є велика ціна та складність у використанні, також обмеженість у функціоналі. Таким чином, плагін під час розробки покладався на усунення цих недоліків.

3. Проаналізовано засоби розробки програмного забезпечення. обрано мову програмування Python, що забезпечує високу надійність та простоту у використанні. Також, Python пропонує багатий вибір інструментів і бібліотек для розроблення плагінів, що забезпечує швидке розроблення та підтримку плагінів. Окрім Python був використан Blender API, який чудово взаємодіє з Python. Взаємодія між Blender API і Python надає розробникам потужний і гнучкий інструментарій для впровадження інновацій і розширення можливостей Blender для задоволення унікальних потреб та завдань користувачів.

Створений плагін має значний потенціал щодо його покращення в подальшому та додавання іншого функціоналу в залежності від розвитку.

ПЕРЕЛІК ПОСИЛАНЬ

1. Heather. The History of the Sketchpad Computer Program [Електронний ресурс] / Heather. – 2022. – Режим доступу до ресурсу: <https://history-computer.com/sketchpad-guide/>.
2. The history of 3D modeling and animation [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://inspirationtuts.com/the-history-of-3d-modeling-and-animation/>.
3. How 3D Modeling is Useful in Advertising [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://c-istudios.com/how-3d-modeling-is-useful-in-advertising/#:~:text=3D%20modeling%20can%20be%20used,create%20animations%20or%20short%20films>
4. Kumbharkar S. Return of 3D Graphics in Web and Mobile UI [Електронний ресурс] / Sanjivani Kumbharkar. – 2022. – Режим доступу до ресурсу: <https://calsoftinc.com/blogs/2022/12/return-of-3d-graphics-in-web-and-mobile-ui.html>. Kyla B.
5. 5 Ways to use Interactive 3D Models in Healthcare [Електронний ресурс] / Ball Kyla. – 2021. – Режим доступу до ресурсу: <https://www.thinglink.com/blog/5-ways-to-use-interactive-3d-models-in-healthcare/>
6. Bondarenko A. 3D Modeling in the Film Industry [Електронний ресурс] / Aleksandra Bondarenko. – 2023. – Режим доступу до ресурсу: <https://www.selfcad.com/blog/how-3d-modeling-is-used-in-the-film-industry>.
7. 3D MODELING IN THE GAMING INDUSTRY [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://retrostylegames.com/blog/3d-modeling-in-gaming-industry/>.
8. 9 BEST PLUGINS FOR BLENDER IN 2023 [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://kitbash3d.com/a/blog/best-plugins-for-blender-in-2023>. Autodesk Maya [Електронний ресурс] – Режим доступу до ресурсу: <https://www.autodesk.com/products/maya/overview>.

9. History of Maya [Электронный ресурс]. – 2021. – Режим доступа до ресурсу: <https://inspirationtuts.com/what-is-maya-history/>.
10. Failes I. A VISUAL HISTORY OF 3DS MAX [Электронный ресурс] / Ian Failes. – 2020. – Режим доступа до ресурсу: <https://beforesandafters.com/2020/06/04/a-visual-history-of-3ds-max/>.
11. What is Cinema 4D? [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: <https://www.nobledesktop.com/learn/cinema-4d/what-is-cinema-4d>.
12. ZBrush Movies VFX [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: <https://www.maxon.net/en/zbrush-movies-vfx>.
13. Blender's History [Электронный ресурс] – Режим доступа до ресурсу: <https://www.blender.org/about/history/>.
14. Garofalo E. How to Animate in Blender [Электронный ресурс] / Emma Garofalo. – 2022. – Режим доступа до ресурсу: <https://www.makeuseof.com/how-to-animate-in-blender/>.
15. WHAT 3D MODELING SOFTWARE IS USED FOR VIDEO GAMES [Электронный ресурс]. – 2022. – Режим доступа до ресурсу: <https://retro-stylegames.com/blog/3d-modeling-software-for-video-games/>.
16. Mein K. Tips for Coding Blender [Электронный ресурс] / Kent Mein – Режим доступа до ресурсу: https://wiki.blender.org/wiki/Tools/Tips_for_Coding_Blender.
17. Blender 3.5 Python API Documentation [Электронный ресурс] // Blender Foundation. – 2023. – Режим доступа до ресурсу: <https://docs.blender.org/api/current/index.html>.
18. API Overview [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: https://docs.blender.org/api/current/info_overview.html

ДОДАТОК А

```
bl_info = {
    "name": "Town maker",
    "author": "Kosenko",
    "version": (1, 0),
    "blender": (2, 80, 0),
    "location": "View3D > Add > Mesh > New Object",
    "description": "Adds a new Mesh Object",
    "warning": "",
    "doc_url": "",
    "category": "Add Mesh",
}

import bpy

def smallhouse(context):

    bpy.ops.mesh.primitive_plane_add(size=2, enter_editmode=False, align='WORLD',
location=(0, 0, 0), scale=(1, 1, 1))

def town(context):

    bpy.ops.mesh.primitive_cube_add(enter_editmode=False, align='WORLD', loca-
tion=(0, 0, 0), scale=(1, 1, 1))

def fence(context):
```

```
bpy.ops.mesh.primitive_cube_add(size=1, enter_editmode=False, align='WORLD',  
location=(0, 0, 0), scale=(1, 1, 1))
```

```
class SimpleOperator(bpy.types.Operator):
```

```
    """Tooltip"""
```

```
    bl_idname = "object.simple_operator"
```

```
    bl_label = "Simple Object Operator"
```

```
    def execute(self, context):
```

```
        smallhouse(context)
```

```
        return {'FINISHED'}
```

```
class SimpleOperatorTown(bpy.types.Operator):
```

```
    """Tooltip"""
```

```
    bl_idname = "object.simple_operator2"
```

```
    bl_label = "Simple Object Operator"
```

```
    def execute(self, context):
```

```
        town(context)
```

```
        return {'FINISHED'}
```

```
class SimpleOperatorFence(bpy.types.Operator):
```

```
    """Tooltip"""
```

```
    bl_idname = "object.simple_operator3"
```

```
    bl_label = "Simple Object Operator"
```

```
    def execute(self, context):
```

```
        fence(context)
```

```
        return {'FINISHED'}
```

```
def menu_func(self, context):
    self.layout.operator(SimpleOperator.bl_idname, text=SimpleOperator.bl_label)
```

```
class PanelOne(bpy.types.Panel):
    """Creates a Panel in the scene context of the properties editor"""
    bl_label = "Town maker addon"
    bl_idname = "SCENE_PT_panel_one"
    bl_space_type = 'PROPERTIES'
    bl_region_type = 'WINDOW'
    bl_context = "scene"

    def draw(self, context):
        layout = self.layout

        # Create some UI elements for panel one
        layout.label(text="Choose your model")
        layout.operator("object.simple_operator", text="Add Small House")
        layout.operator("object.simple_operator2", text="Add Town House")
        layout.operator("object.simple_operator3", text="Add Fence")
```

```
class MyAddonPanel(bpy.types.Panel):
    bl_idname = "OBJECT_PT_my_addon"
    bl_label = "Мой плагин"
    bl_space_type = 'VIEW_3D'
    bl_region_type = 'UI'
    bl_category = 'My Addon'
```

```
def draw(self, context):
    layout = self.layout
```

```
obj = context.active_object
```

```
if obj is not None and obj.type == 'MESH':
```

```
    layout.label(text="copy:")
```

```
    layout.prop(context.scene, "copy_distance", text="dist")
```

```
    layout.prop(context.scene, "copy_axes", text="axis")
```

```
    layout.prop(context.scene, "copy_count", text="amountcopy")
```

```
    layout.operator("object.add_copy_operator")
```

```
    layout.operator("object.remove_copy_operator")
```

```
class AddCopyOperator(bpy.types.Operator):
```

```
    bl_idname = "object.add_copy_operator"
```

```
    bl_label = "addcopy"
```

```
    def execute(self, context):
```

```
        obj = context.active_object
```

```
        distance = context.scene.copy_distance
```

```
        axes = context.scene.copy_axes
```

```
        count = context.scene.copy_count
```

```
        array_modifier = obj.modifiers.new(name="Array", type='ARRAY')
```

```
        array_modifier.count = count
```

```
        array_modifier.use_relative_offset = True
```

```

array_modifier.relative_offset_displace = (0.0, 0.0, 0.0)

for axis in axes:
    if axis == 'X':
        array_modifier.relative_offset_displace[0] = distance
    elif axis == 'Y':
        array_modifier.relative_offset_displace[1] = distance
    elif axis == 'Z':
        array_modifier.relative_offset_displace[2] = distance

return {'FINISHED'}

```

```

class RemoveCopyOperator(bpy.types.Operator):
    bl_idname = "object.remove_copy_operator"
    bl_label = "delcopy"

    def execute(self, context):
        obj = context.active_object

        array_modifier = obj.modifiers.get("Array")
        if array_modifier is not None:
            obj.modifiers.remove(array_modifier)

        return {'FINISHED'}

```

```

def register():
    bpy.utils.register_class(MyAddonPanel)

```

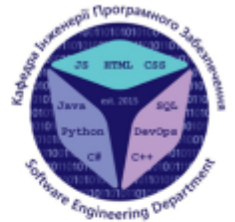
```
bpy.utils.register_class(AddCopyOperator)
bpy.utils.register_class(RemoveCopyOperator)
bpy.types.Scene.copy_distance = bpy.props.FloatProperty(name="dist", default=1.0)
bpy.types.Scene.copy_axes = bpy.props.EnumProperty(
    name="axis",
    items=[('X', "X", ""), ('Y', "Y", ""), ('Z', "Z", "")],
    options={'ENUM_FLAG'}
)
bpy.types.Scene.copy_count = bpy.props.IntProperty(name="Copy", default=1,
min=1, max=100)
```

```
def unregister():
    bpy.utils.unregister_class(MyAddonPanel)
    bpy.utils.unregister_class(AddCopyOperator)
    bpy.utils.unregister_class(RemoveCopyOperator)
    del bpy.types.Scene.copy_distance
    del bpy.types.Scene.copy_axes
    del bpy.types.Scene.copy_count
```


ДОДАТОК Б



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



РОЗРОБКА ПЛАГІНУ ДЛЯ СПРОЩЕНОГО СТВОРЕННЯ 3D МОДЕЛЕЙ У ПРОГРАМІ BLENDER МОВОЮ PYTHON

Виконав студент 4 курсу
Групи ПД-44
Косенко Денис Максимович
Керівник роботи

К.п.н, доц, професор кафедри ТЦР Корецька Вікторія Олександрівна

Київ – 2023

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

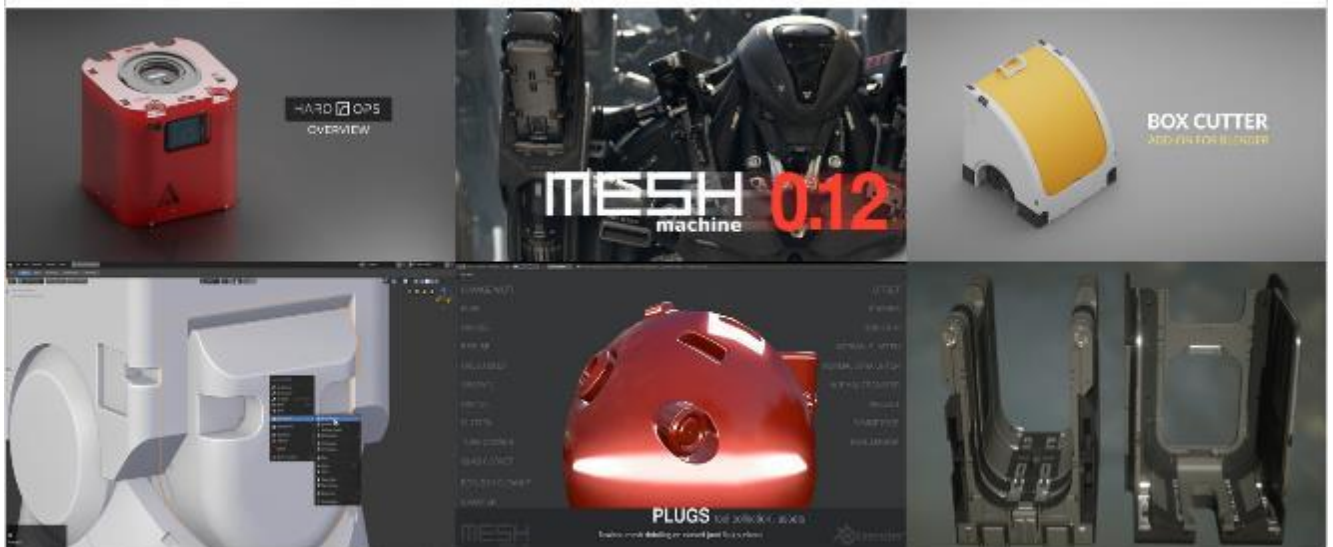
- **Мета роботи:** Спрощення процесу створення 3D-моделей у Blender за допомогою розробленого плагіну.
- **Об'єкт дослідження:** процес створення 3D моделей.
- **Предмет дослідження:** плагіни для спрощеного створення 3D моделей.

ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

1. Аналіз існуючих методів створення 3D моделей у Blender та виявлення їх недоліків.
2. Аналіз існуючих плагінів для спрощеного створення 3D моделей.
3. Реалізація плагіну мовою Python для програми Blender.
4. Тестування розробленого плагіну.
5. Оцінка ефективності та порівняння плагіну з існуючими рішеннями.

3

АНАЛІЗ АНАЛОГІВ



4

АНАЛІЗ АНАЛОГІВ



АНАЛІЗ АНАЛОГІВ

	HardOps	MeshMachine	BoxCutter	RetopoFlow	KITOPS	TownMaker
Кількість функцій	5	3	4	5	2	5
Створення моделей	-	-	+	-	-	+
Ретопологія	-	-	-	+	-	-
Зміна параметрів об'єктів	+	+	+	-	-	+
Підтримка гарячих клавіш	+	+	-	+	-	-
Дублювання та видалення моделей	-	-	-	-	+	+

ВИМОГИ ДО ПЛАГІНУ

1. Модель сільського будинку.
2. Модель міського будинку.
3. Модель паркану.
4. Панель для створення моделей.
5. Панель для редагування параметрів моделей.
6. Модульність моделей.
7. Комбінування об'єктів між собою.

7

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



Blender Python API



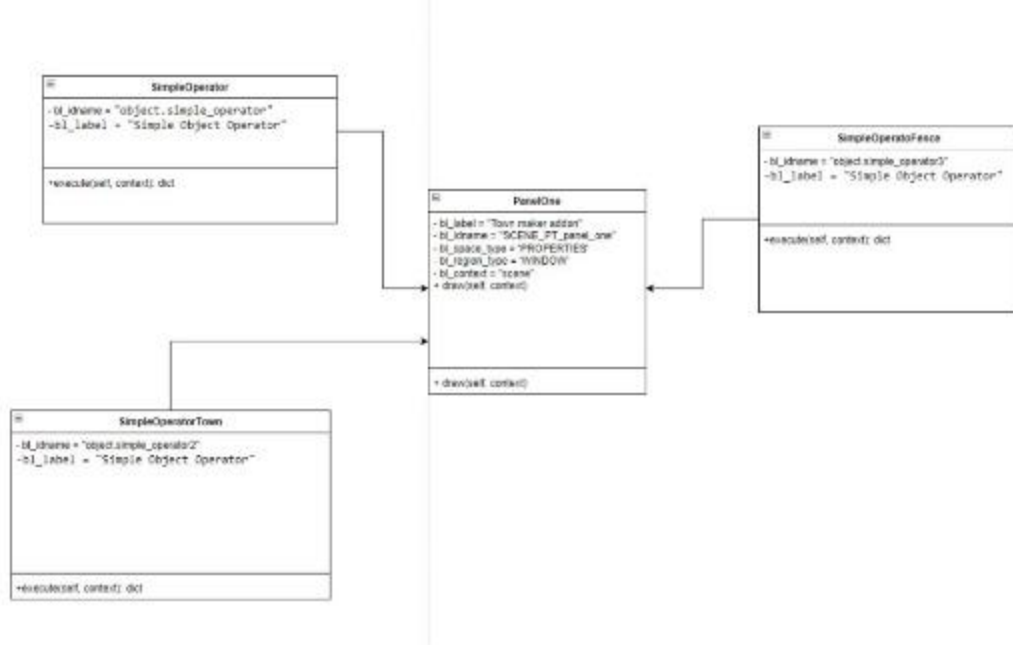
8

ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ



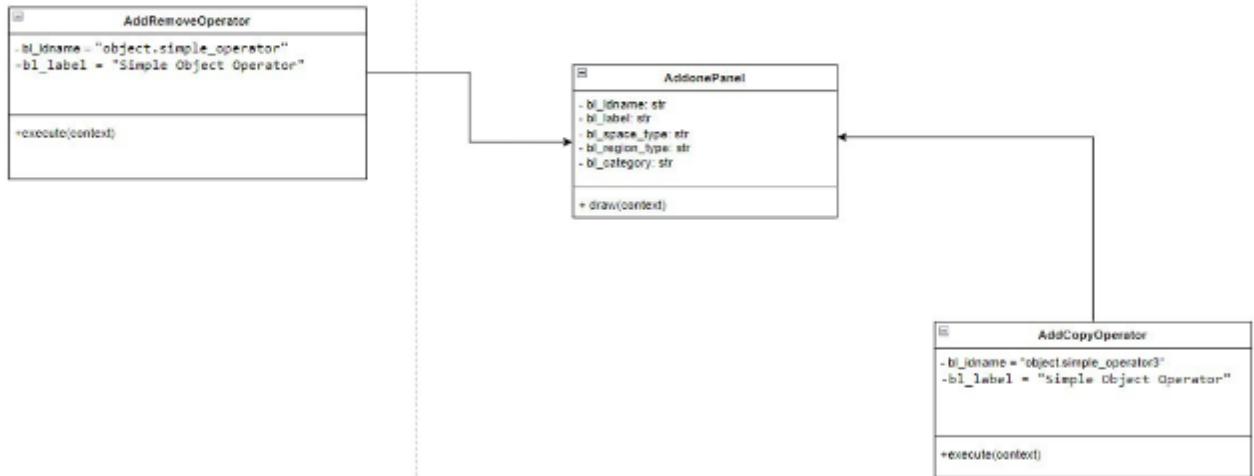
9

ДІАГРАМА КЛАСІВ ДЛЯ СТВОРЕННЯ МОДЕЛЕЙ



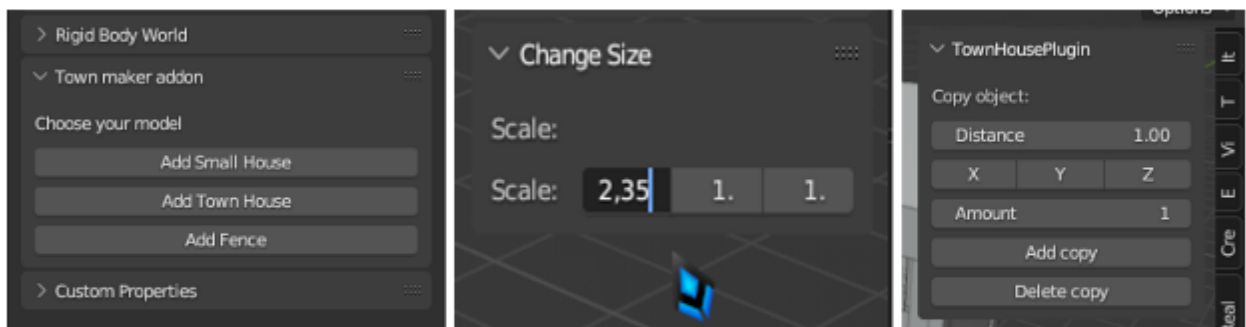
10

ДІАГРАМА КЛАСІВ ДЛЯ ДУБЛЮВАННЯ ТА ВИДАЛЕННЯ МОДЕЛЕЙ



11

ЕКРАННІ ФОРМИ



Головний екран плагіну

Інтерфейс зміни розмірів

Інтерфейс дублювання моделі

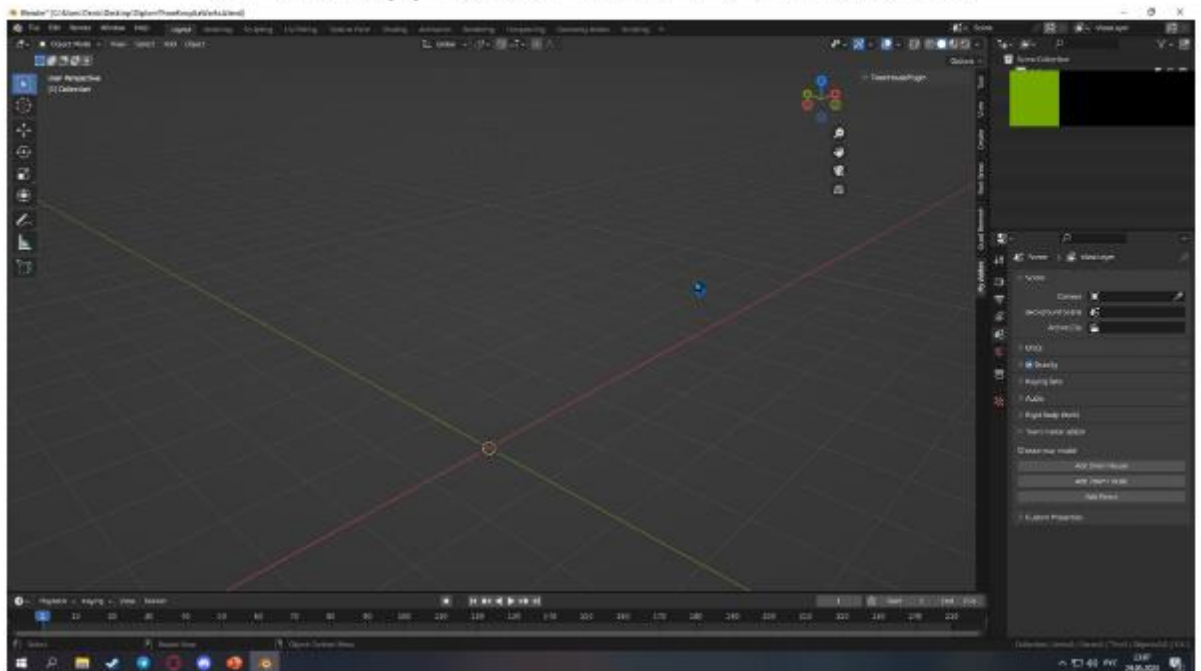
12

ПРИКЛАД ВИКОРИСТАННЯ ПЛАГІНУ



13

ПРИКЛАД ВИКОРИСТАННЯ ПЛАГІНУ



14

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

1. Косенко Д.М. Оптимізація процесу моделювання в комп'ютерній інженерії з використанням плагінів /Корсцька В.О, Косенко Д.М// Сучасні аспекти діджиталізації та інформатизації в програмній та комп'ютерній інженерії. 01-03 червня 2023р., ДУТ, м. Київ. – К: ДУТ, 2023. Подано до друку.
2. Косенко Д.М. Розробка плагіну для спрощеного створення 3D моделей у програмі Blender мовою Python / Корсцька В.О, Косенко Д.М //Сучасний етап та перспективи розвитку IoT. С. 211-212. 07 квітня 2023р., ДУТ, м. Київ – К: ДУТ, 2023.

15

ВИСНОВКИ

1. Проведено аналіз існуючих методів створення 3D моделей у Blender. Їх актуальність, та ключові проблеми, які можна виділити. Визначено основну ключову аудиторію.
2. Розглянуто аналоги плагіну, які можуть бути використані, для спрощення процесу створення 3D моделей, та розроблена таблиця порівняння аналогів.
3. Розроблено плагін для спрощеного створення моделей. Реалізовано панель, для створення моделей, зміни їх параметрів, та комбінування між собою за допомогою модульної системи.
4. Проведене функціональне тестування плагіну продемонструвало відсутність помилок.

16