

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ
РОЗШИРЕННЯ ФУНКЦІОНАЛЬНИХ МОЖЛИВОСТЕЙ
СЕРЕДОВИЩА BLENDER
SOFTWARE DEVELOPMENT TO EXTEND THE
FUNCTIONALITY OF THE BLENDER ENVIRONMENT

Виконав: студент 4 курсу, групи КНТ-211сп

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

МАСЛОВ В.В.

(ПРІЗВИЩЕ та ініціали)

Керівник ПАРХОМЕНКО А.В.

(ПРІЗВИЩЕ та ініціали)

Рецензент ЗЕЛЕНЬОВА І.Я.

(ПРІЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Факультет комп'ютерних наук та технологій
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 122 Комп'ютерні науки
(код і найменування)
Освітня програма (спеціалізація) Комп'ютерні науки
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2024 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

МАСЛОВА Вадима Вячеславовича
(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка програмного забезпечення для розширення функціональних можливостей середовища Blender. Software Development to Extend the Functionality of the Blender Environment
керівник проєкту (роботи) к.т.н., доцент, ПАРХОМЕНКО Анжеліка Володимирівна,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “24” квітня 2024 року № 168

2. Строк подання студентом проєкту (роботи) 10 червня 2024 року

3. Вихідні дані до проєкту (роботи) рекомендована література, завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Проєктування програмного забезпечення. 3. Конструювання програмного забезпечення. 4. Інструкція з використання програмного забезпечення

5. Перелік графічного матеріалу (з з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)
12 слайдів презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ПАРХОМЕНКО А.В., доцент		
Нормоконтроль	БЄЛОВА А.В., асистент		

7. Дата видачі завдання «15» квітня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Срок виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка архітектури програми.	2 тиждень	Розділ 2
5	Розробка програми.	3-4 тижні	Розділ 3
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис) Вадим МАСЛОВ
(Ім'я ПРІЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Анжеліка ПАРХОМЕНКО
(Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 79 с., 7 табл., 25 рис., 2 дод., 24 джерел.

3D МОДЕЛЬ, СЕРЕДОВИЩЕ BLENDER, ТОПОЛОГІЯ, ОПТИМІЗАЦІЯ, АДДОН.

Об'єкт дослідження – процес оптимізації топології 3D моделей.

Предмет дослідження – методи та програмні засоби для оптимізації топології 3D моделей.

Метою роботи є розробка програмного забезпечення для розширення функціональних можливостей середовища Blender та оптимізації топології 3D моделей.

Матеріали, методи та технічні засоби: об'єктно-орієнтована мова програмування – Python, інтерфейс програмування Blender API, Blender 3.6.10, персональний комп'ютер з процесором AMD Ryzen 5 5600G під управлінням операційної системи Microsoft Windows 11.

Результати. Створено програмне забезпечення, яке інтегрується з Blender та підвищує ефективність роботи з 3D моделями.

Висновки. Розроблений аддон для Blender є зручним інструментом, що дозволить користувачам створювати більш складні та якісні моделі, зберігаючи при цьому оптимальні показники продуктивності та використання ресурсів.

Галузь використання – кінематограф, анімація, ігрова індустрія, віртуальна та доповнена реальність, 3D-друкування, візуалізація в архітектурі та медицині.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 79 pages, 7 tables, 25 figures, 2 appendixes, 24 sources.

3D MODEL, BLENDER ENVIRONMENT, TOPOLOGY, OPTIMIZATION, ADD-ON.

Object of study is the process of optimizing the topology of 3D models.

Subject of study are methods and software tools for optimizing the topology of 3D models.

The purpose of the work is the development of software to expand the functional capabilities of the Blender environment and increase the efficiency of working with 3D models.

Materials, methods, and technical tools: object-oriented programming language Python, Blender API programming interface, Blender 3.6.10, personal computer with AMD Ryzen 5 5600G processor running Microsoft Windows 11 operating system.

Results. Software has been created that integrates with Blender and improves the efficiency of working with 3D models.

Conclusions. The developed add-on for Blender is a convenient tool that allows users to create more complex and high-quality models while maintaining optimal performance and resource usage.

The field of use is cinema, animation, gaming industry, virtual and augmented reality, 3D printing, visualization in architecture and medicine.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	7
Вступ.....	8
1 Аналіз предметної області.....	9
1.1 Огляд та порівняння програмного забезпечення для 3D моделювання ..	9
1.2 Аналіз особливостей середовища Blender	11
1.3 Типи аддонів.....	13
1.4 Аналіз аддонів для оптимізації геометрії.....	16
1.5 Постановка задач дипломної роботи	20
2 Проєктування програмного забезпечення	22
2.1 Призначення та функціональні можливості аддона.....	22
2.2 Аналіз вимог	22
2.3 Архітектура системи.....	23
2.4 Вибір інструментарію розробки	25
2.5 Методика розробки аддона	29
2.6 Висновки до розділу 2	34
3 Конструювання програмного забезпечення	36
3.1 Схема алгоритму функціонування програмного забезпечення	36
3.2 Опис особливостей програмної реалізації.....	38
3.3 Висновки до розділу 3	42
4 Інструкція з використання програмного забезпечення.....	43
4.1 Практичне застосування аддона.....	43
4.2 Експериментальне дослідження ефективності розробленого аддона ...	54
4.3 Висновки до розділу 4	60
Висновки	61
Перелік джерел посилань	62
Додаток А Текст програми.....	64
Додаток Б Слайди презентації	73

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API	– Application Programming Interface;
BIM	– Building Information Modeling;
CLI	– Command Line Interpreter;
HDR	– High Dynamic Range;
IDE	– Integrated Development Environment;
UI	– User Interface.

ВСТУП

3D-модельовання відіграє ключову роль у сучасному світі, охоплюючи безліч галузей та сфер діяльності. Від створення фільмів та ігор до проєктування продуктів та архітектури, тривимірне модельовання є невід'ємною частиною творчого процесу та виробництва.

В умовах швидкого технологічного прогресу та появи нових завдань перед користувачами, стає очевидною необхідність постійного розвитку інструментів та програмного забезпечення. Незважаючи на широкий вибір існуючих інструментів, завжди є потреба в нових рішеннях, здатних вирішувати сучасні завдання більш ефективно та зручно.

У цьому контексті аддон для Blender залишається актуальним та важливим інструментом. Дозволяючи оптимізувати геометрію моделей та спрощувати процес роботи з ними, аддон забезпечує про користувачів 3D-модельовання засобами для вирішення сучасних завдань та підвищення продуктивності. Такий інструмент стає незамінним в умовах індустрії, що швидко розвивається, тривимірної графіки та дизайну.

Метою роботи є розробка програмного забезпечення для розширення функціональних можливостей середовища Blender та оптимізації топології 3D моделей.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд та порівняння програмного забезпечення для 3D моделювання

Autodesk Maya, ZBrush і Blender є провідними програмами у сфері 3D моделювання та візуалізації, і кожна з них займає вагому позицію в середовищі 3D користувачів. Хоча ці програми мають певну кількість спільних рис та працюють на загальних принципах, особливо у функціональності інструментів для скульптингу, вони відрізняються в ряді ключових аспектів, що надає їм унікальний характер і призначення в 3D індустрії (табл. 1.1).

Таблиця 1.1 – Порівняння програм для 3D моделювання [1]-[3]

Критерій	Blender	Autodesk Maya	ZBrush
1	2	3	4
Платформи	Windows, macOS, Linux	Windows, macOS Linux	Windows, macOS
Основні функції	Моделювання, скульптинг, анімація, рендеринг, композитинг	Моделювання, анімація, ригінг, динаміка, рендеринг	Скульптинг, текстурування, рендеринг
Інтерфейс	Гнучкий, багатофункціональний	Професійний, комплексний	Спеціалізований для скульптингу
Підтримка плагінів	Широкий вибір безкоштовних та платних аддонів	Широкий вибір плагінів	Обмежений набір плагінів
Застосування	Кіноіндустрія, відеоігри, анімація, медицина, промисловість	Кіноіндустрія, відеоігри, анімація	Створення високодеталізованих моделей та текстур

Продовження таблиці 1.1

1	2	3	4
Вартість	Безкоштовно, Open Source	Платна, підписка	Платна, ліцензія
Переваги	Безкоштовно, відкритий вихідний код, інтеграція з різними плагінами та скриптами	Широкий функціонал для анімації та рендерингу, підтримка індустрії	Найкращі інструменти для скульптингу та текстурування
Недоліки	Менше стандартів в індустрії може вимагати більше налаштування	Висока вартість, займає багато часу на освоєння	Обмежена підтримка анімації та ригінгу

ZBrush, Blender та Autodesk 3ds Max мають свої унікальні особливості та призначення в області 3D-моделювання та візуалізації. Зробивши аналіз, можна зробити висновок, що Blender є одним з найбільш привабливих інструментів для 3D моделювання та візуалізації. Відкритий код та безкоштовність дозволяють зробити його доступним для широкого кола користувачів [1].

Крім того, Blender має широкий функціонал, який включає повний набір інструментів для різних завдань від створення 3D моделей до анімації та відеомонтажу. Популярність та активна спільнота користувачів роблять Blender важливим ресурсом для навчання та підтримки у створенні проєктів. Ці фактори роблять Blender вигідним вибором для багатьох користувачів, незалежно від їхніх потреб і рівня вмінь у 3D графіці.

1.2 Аналіз особливостей середовища Blender

Blender дозволяє працювати з різними типами геометричних об'єктів, такими як базові полігональні форми (наприклад, куби, сфери, циліндри), криві, поверхні, metaballs, а також з векторними шрифтами. За допомогою перетину та зміни розмірів окремих полігонів можна створювати складніші об'єкти. Крім того, доступна функція малювання довільних кривих за допомогою олівця (Grease Pencil) [1], [4]-[5].

У Blender кожен елемент, з яким користувач взаємодіє, називається об'єктом. Це може бути полігональна модель, крива, джерело світла, камера огляду, арматура моделі тощо. Об'єкти групуються в колекції, що дозволяє краще організувати робочу область. Blender також надає можливість створювати матеріали та текстури за допомогою вузлів, які дозволяють встановлювати різноманітні параметри взаємодії об'єктів [1], [5].

Додатково, Blender пропонує інструменти для анімації, включаючи різноманітні методи деформації об'єктів, а також можливість розширення функціоналу за допомогою додатків, які підтримуються у форматах .zip або .py(Python) [1].

Blender використовує внутрішню файлову систему для зберігання сцен з усіма їхніми складовими, такими як моделі, текстури та ефекти, в єдиному файлі формату .blend. Ці файли є сумісними з різними версіями Blender і переносними між різними платформами, що дозволяє зручно переміщати та зберігати контент [1].

Користувачі можуть зберігати налаштування робочого середовища в .blend файлах, що забезпечує збереження інтерфейсу та розміщення об'єктів між сеансами роботи з програмою [1].

Додатково, у Blender є можливість прив'язувати або прикріплювати елементи зовнішніх файлів до поточного проєкту. Це дозволяє зручно використовувати різноманітні ресурси без необхідності включати їх у .blend файл [1].

Під час роботи програма автоматично створює резервні копії проєктів, що забезпечує збереження даних у випадку непередбачених ситуацій [1].

Важливо зауважити, що структура .blend файлів схожа на прямий дамп пам'яті програми, що ускладнює їх перетворення в інші формати. Тим не менше, окремі елементи, такі як моделі, можуть бути експортовані в інші формати для подальшого використання [1].

Основу інтерфейсу Blender складають горизонтальні вкладки, які розділені на різні робочі простори для зручного перемикання між функціями. Кожна вкладка містить панелі інструментів, розташовані вертикально справа, з якими користувач може працювати. Багато функцій мають свої власні комбінації клавіш, що полегшує і прискорює роботу [1].

Після того, як Blender став проєктом з відкритим вихідним кодом, до всіх функцій було додано повні контекстні меню, а використання інструментів стало логічнішим та гнучкішим. Користувацький інтерфейс підтримує різні колірні схеми оформлення, а також прозорі плаваючі елементи, що розширюють можливості Blender [1].

Для зручності, до окремих об'єктів та їхніх полігонів можна додавати нотатки, що спрощує роботу з великими проєктами [1].

Основні вкладки у користувацькому інтерфейсі Blender-а включають такі робочі простори, як "Моделювання", "Скульптинг", "Редагування розгортки", "Малювання текстури", "Шейдинг", "Анімація", "Рендеринг", "Компонування", "Геометричні вузли" та "Скриптування" [1].

У Blender доступні різні рушії для рендерингу, включаючи Workbench, Eevee та Cycles, кожен з яких має свої унікальні особливості та призначення. Також існують зовнішні рушії, які можна використовувати безкоштовно на безоплатній основі, що розширює можливості програми [1].

1.3 Типи аддонів

Blender підтримує використання аддонів, які розширюють його функціональність та надають додаткові можливості користувачам. На рисунку 1.1 наведено типи аддонів для Blender.

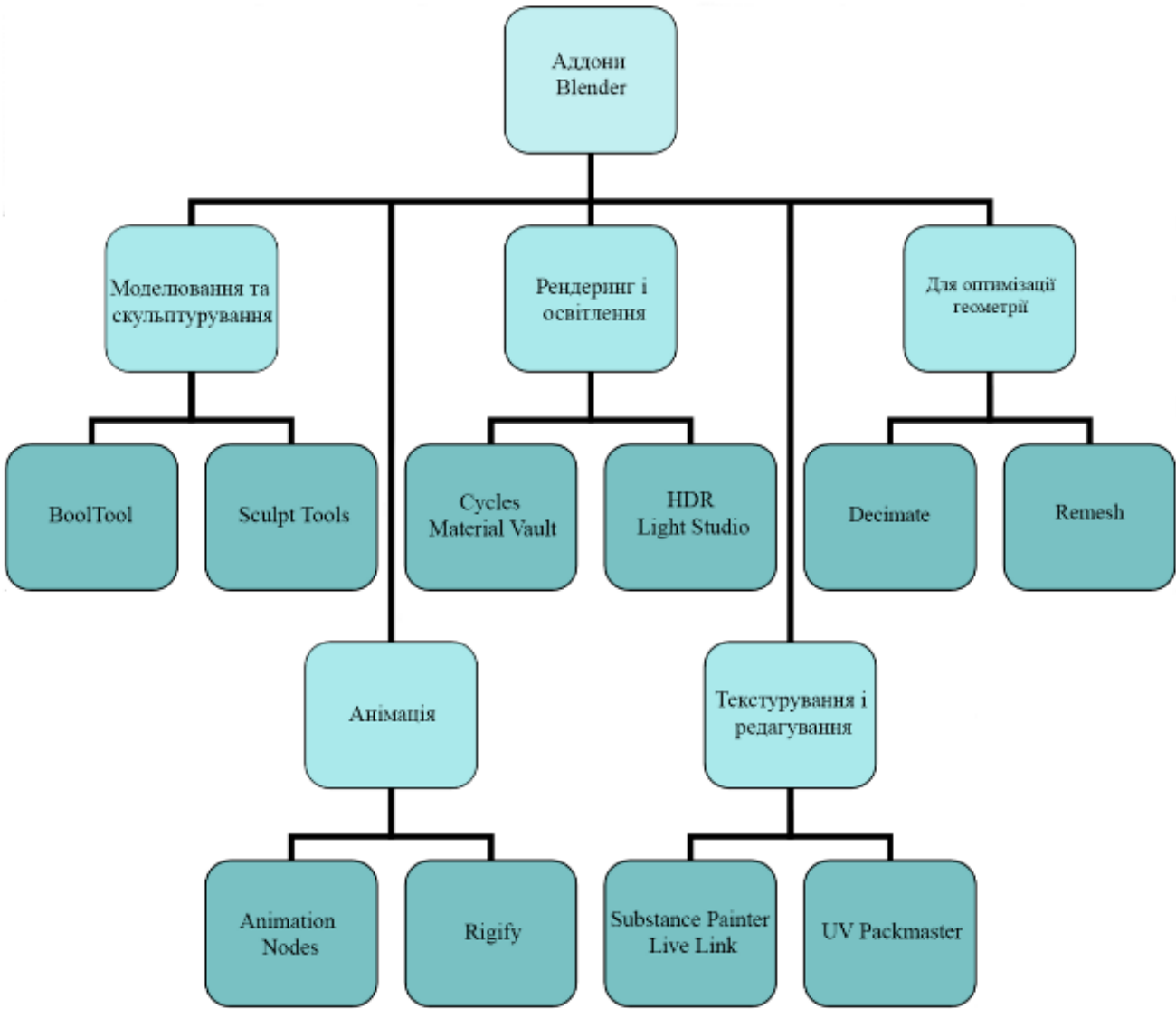


Рисунок 1.1 –Типи аддонів для Blender

Існує кілька типів аддонів, спрямованих на різні аспекти творчого процесу.

Для моделювання та скульптури.

BoolTool. Значно спрощує роботу з булевими операціями у Blender. Він дозволяє користувачам легко і швидко створювати складні геометричні форми шляхом об'єднання, віднімання та перетину об'єктів. Заощаджує час при створенні складних моделей та дозволяє уникнути ручного виконання складних операцій.

Sculpt Tools. Цей аддон покращує стандартні інструменти скульптингу у Blender, додаючи нові можливості та функції. Він надає додаткові пензлі та налаштування, які допомагають створювати більш детальні та складні моделі. Підвищує ефективність роботи з деталями, особливо корисним для створення високополігональних моделей [6]-[7].

Для анімації.

Animation Nodes. Це потужний інструмент, що дозволяє створювати складні анімації за допомогою візуального програмування. Користувачі можуть будувати анімаційні схеми, пов'язуючи вузли, що відкриває практично безмежні можливості для творчості. Полегшує створення складних анімаційних ефектів, не вимагаючи глибоких знань у програмуванні.

Rigify. Він автоматизує процес створення скелетних структур для персонажів, дозволяючи швидко та легко створювати клуні для анімації. Він надає набір готових до використання клунь, які можна адаптувати під конкретного персонажа. Скорочує час створення ригів і полегшує процес анімації персонажів [8]-[9].

Для рендерингу і освітлення.

Cycles Material Vault. Надає бібліотеку готових матеріалів для рендерингу Cycles. Це значно спрощує процес створення реалістичних сцен, пропонуючи широкий вибір матеріалів для використання у проєктах. Прискорює процес рендерингу за рахунок використання готових матеріалів, покращує якість фінальних зображень.

HDR Light Studio. Цей аддон дозволяє точно налаштовувати освітлення сцени за допомогою HDR-зображень. Він забезпечує високу якість

освітлення, що особливо важливо для створення реалістичних та візуально привабливих сцен. Покращує контроль за освітленням, підвищує реалізм фінальних рендерів [10]-[11].

Для текстуровання і редагування.

Substance Painter Live Link. Цей аддон забезпечує зв'язок між Blender та Substance Painter, дозволяючи швидко та зручно переносити текстури між програмами. Це особливо корисно для текстуровання моделей у реальному часі. Спрощує процес текстуровання, покращує інтеграцію між різними програмами.

UV Packmaster. Цей аддон автоматизує процес розміщення UV розгортки на текстурних атласах, оптимізує розміщення UV-шеллів, що допомагає уникнути перекриттів та ефективно використовувати текстурний простір. Заощаджує час при створенні UV-розгортки, покращує якість текстуровання [12].

Для оптимізації геометрії.

Decimate. Цей аддон надає широкий набір інструментів для ретопології у Blender. Він дозволяє легко створювати нову топологію поверх існуючої моделі, забезпечуючи контроль за кожним полігоном. RetopoFlow дозволяє проводити процес ретопології швидко та ефективно, з можливістю налаштування всіх аспектів створюваної геометрії [15].

Remesh. Цей аддон використовує алгоритми автоматичної ретопології для створення квадратної геометрії з існуючої моделі. Він дозволяє швидко та ефективно створювати оптимальну топологію, зберігаючи деталі та форму моделі. QuadRemesher є добрим вибором для тих, хто хоче швидко отримати якісну ретопологію без необхідності робити це вручну [16].

Аддони для Blender є важливим елементом для розширення можливостей цієї платформи. Вони спрощують робочий процес, оскільки вони пропонують нові інструменти та можливості, які можуть зробити створення складних моделей та анімацій більш ефективним. Вони також можуть допомогти підвищити реалізм та якість створюваних моделей.

Вибір відповідних аддонів залежить від конкретних потреб проєкту та переваг користувача. Наприклад, деякі аддони можуть бути орієнтовані на моделювання, інші на анімацію, а треті на створення текстур або управління освітленням. Важливо вибирати аддони, які найкраще відповідають цілям проєкту та перевагам користувачів.

1.3 Аналіз аддонів для оптимізації геометрії

Blender має вбудовані модифікатори Decimate і Remesh, які надають додаткові можливості для управління топологією моделей. Ці інструменти можуть бути корисними в різних ситуаціях, пов'язаних з оптимізацією та переробкою геометрії.

Модифікатор Decimate використовується для зменшення кількості полігонів у моделі, зберігаючи у своїй загальну форму об'єкта [15].

Модифікатор Decimate зображено на рисунку 1.2.

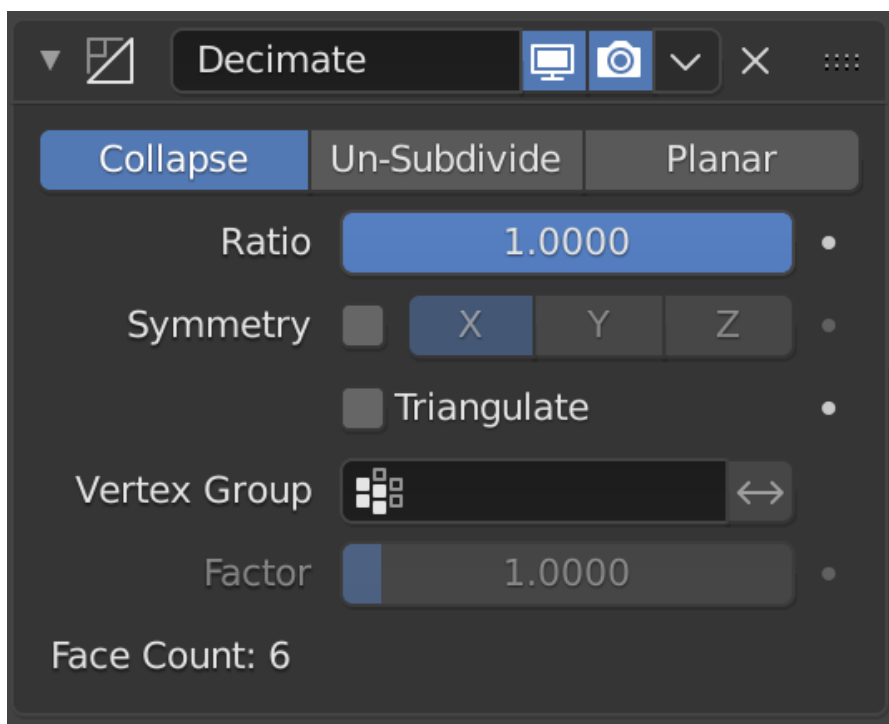


Рисунок 1.2 – Інтерфейс аддону Decimate

Це особливо корисно при оптимізації високополігональних моделей для використання в іграх, анімаціях або покращення продуктивності.

Collapse. Цей режим зменшує кількість полігонів шляхом об'єднання пар вершин. Приклад застосування зображено на рисунку 1.2. Користувачі можуть контролювати ступінь редукції за допомогою параметра Ratio, який визначає відсоток полігонів, що залишилися [15].

Un-Subdivide. Цей режим "розгортає" дії попередніх сабдивіженів (subdivision), зменшує кількість полігонів, дотримуючись їхньої початкової структури. Корисний для моделей, які були багаторазово поділені на дрібніші полігони [15].

Planar. Цей режим видаляє межі, що знаходяться в одній площини чи близькі до неї. Він корисний для спрощення геометрії плоских поверхонь, що може значно зменшити кількість полігонів без значної втрати деталей [15].

Приклад застосування зображено на рисунку 1.2.



Рисунок 1.2 – Демонстрація роботи опцій аддону Decimate

Модифікатор Remesh використовується для перебудови геометрії моделі з метою створення більш рівномірної та організованої топології [16]. Модифікатор Remesh зображено на рисунку 1.3.

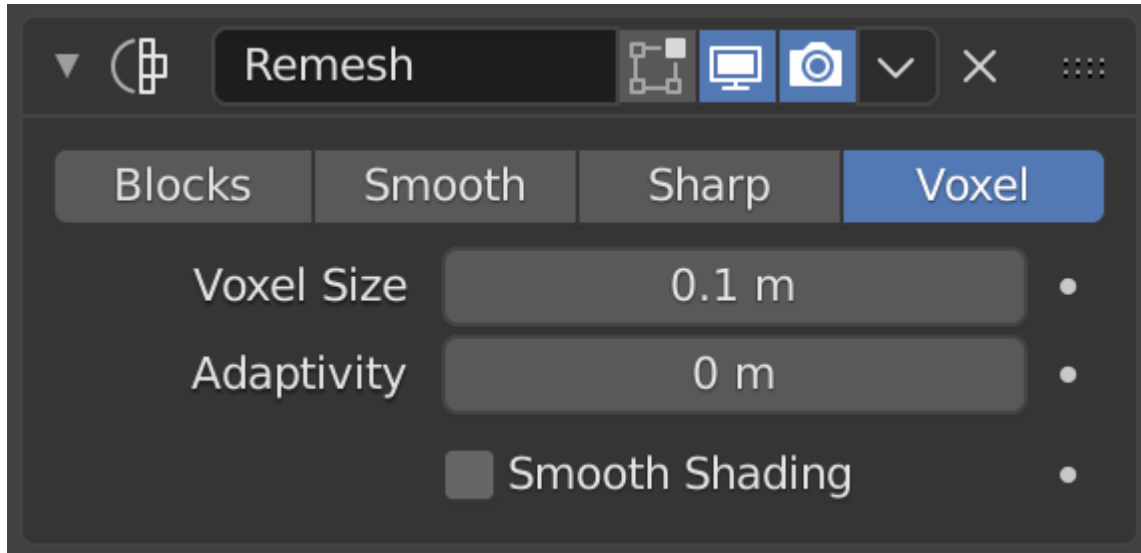


Рисунок 1.3 – Інтерфейс аддону Remesh

Аддон Remesh пропонує кілька режимів роботи.

Blocks. Цей режим створює блокову структуру, розділяючи модель на куби. Він корисний для створення стилізованих або низькополігональних моделей, а також для швидкого створення начерків [16].

Приклад застосування зображено на рисунку 1.4.

Smooth. Цей режим створює більш згладжену та рівномірну сітку, що робить його корисним для перетворення сканів або високополігональних моделей на більш керовану форму для подальшого редагування [16].

Приклад застосування зображено на рисунку 1.4.

Sharp. Цей режим зберігає гострі краї та деталі моделі, створюючи чіткішу топологію. Корисний для моделей із великою кількістю деталей, де потрібно зберегти різкість граней [16].

Приклад застосування зображено на рисунку 1.4.

Voxel. Цей режим використовує воксельну модель для створення нової сітки. Він дозволяє керувати деталізацією за допомогою параметра

Voxel Size і добре підходить для відновлення геометрії після операцій з болю або роботи з пошкодженими сітками [16].

Приклад застосування зображено на рисунку 1.4.

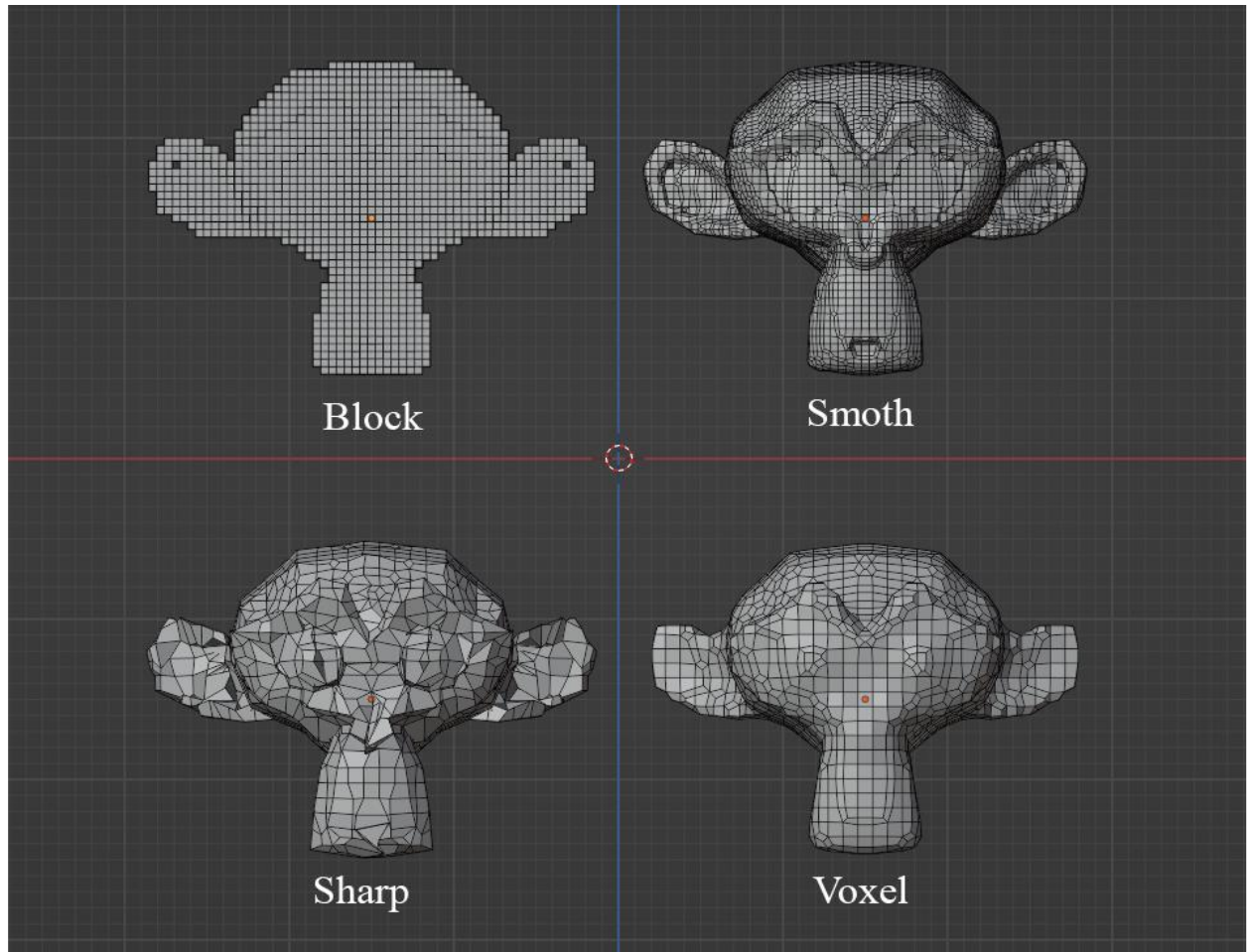


Рисунок 1.4 – Демонстрація роботи опцій аддону Remesh

Обидва модифікатори, Decimate та Remesh, надають різні підходи до оптимізації та переробки топології моделі в Blender.

Decimate чудово підходить для ситуацій, коли потрібно зменшити кількість полігонів без значної зміни загальної форми моделі. Його гнучкість у налаштуваннях дозволяє точно контролювати процес редукції.

Remesh корисний для створення більш упорядкованої та рівномірної топології. Він може бути корисний для перетворення сканів, високополігональних моделей або підготовки геометрії для подальшого скульптингу.

Використання модифікаторів Decimate та Remesh у Blender надає потужні засоби для управління топологією моделей. Вони доповнюють можливості інших інструментів, таких як RetopoFlow та QuadRemesher, пропонуючи різні підходи до вирішення задач оптимізації та переробки геометрії. Тому розробка аддону для оптимізації петель також може бути корисним разом з цими модифікаторами, щоб надати користувачам Blender ще більше можливостей для роботи з моделями.

1.5 Постановка задач дипломної роботи

В результаті аналізу функціональності та особливостей програмного пакету Blender, можна зробити висновок, що Blender є потужним інструментом для створення тривимірних моделей та анімацій. Завдяки своїй гнучкості та різноманітним можливостям, Blender користується популярністю серед 3D-художників та аніматорів. В Особливості, такі як робота з різними типами об'єктів та організація робочої області, роблять його зручним та ефективним інструментом для реалізації найрізноманітніших проєктів – від створення анімаційних фільмів до розробки ігор.

На даний момент існує ряд аддонів та інструментів, які значно покращують функціональність Blender у різних галузях, таких як моделювання, анімація, рендеринг, текстурування та організація проєкту. Вони надають додаткові можливості та інструменти для роботи з програмою, що робить процес створення тривимірних моделей більш ефективним та зручним для користувачів. У сучасному світі, де вимоги до якості та продуктивності тривимірних моделей постійно зростають, актуальність роботи аддону для оптимізації полігональної сітки у Blender важко переоцінити.

Таким чином, аналіз існуючих інструментів та їх функціональності наголошує на важливості створення аддону для оптимізації полігональної сітки в Blender. На відміну від Decimate та Remesh, які виконують загальну

оптимізацію геометрії, розробка аддона буде надавати більш детальне та контрольоване рішення для керування петлями. Це особливо корисно для моделлерів, які потребують точного керування конкретними аспектами топології, такими як кути між суміжними гранями та збереження важливих деталей у певних областях моделі.

Аддон стане важливим доповненням до існуючих інструментів оптимізації в Blender, пропонуючи спеціалізовані можливості, які можуть бути особливо корисними для професіоналів, які потребують точного контролю над топологією своїх моделей.

Створення такого аддону потребує глибокого розуміння алгоритмів оптимізації та роботи з топологією 3D моделей. Однак його розробка значно спростить процес оптимізації полігональних мереж, зробить його більш інтуїтивним та ефективним. В результаті користувачі зможуть швидше та якісніше працювати з високополігональними моделями, покращуючи продуктивність та скорочуючи час, необхідний для підготовки моделей до рендерингу чи анімації.

Тому розробка аддону для швидкої та зручної оптимізації полігональної сітки об'єкта у Blender є актуальною та потребує подальшого дослідження та реалізації.

Метою роботи є розробка програмного забезпечення для розширення функціональних можливостей середовища Blender та оптимізації топології 3D моделей.

Для досягнення мети необхідно розв'язати наступні задачі:

- провести аналіз існуючих аддонів для оптимізації полігональних сіток у Blender та визначити актуальність створення власного аддону ;
- розробити алгоритм функціонування програмного забезпечення, що враховує особливості роботи аддона в Blender;
- реалізувати аддон, який інтегруватиметься з Blender і надасть можливість оптимізації полігональних сіток моделей;
- провести тестування аддону та оцінити його ефективність.

2 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Призначення та функціональні можливості аддона

Розроблюваний аддон для Blender призначений для оптимізації полігональних сіток та видалення надлишкових елементів.

Аддон для Blender має забезпечувати наступні можливості:

- автоматичне видалення непотрібних ребер зниження кількості полігонів без шкоди для форми моделі;
- автоматичне покращення структури петель для підвищення якості деформації моделі під час анімації, а також інструменти для аналізу та візуалізації поточного стану топології моделі;
- автоматизація процесу ретопології, що робить моделі більш керованими та легкими для текстурування та анімації;
- можливість налаштування мінімального та максимального кута для видалення та оптимізації петель;
- можливість зберігати певні петлі та вибирати лише ті, які мають бути оптимізовані;
- зручний інтерфейс з можливістю включення та вимикання накладень для кращого візуального контролю за процесом оптимізації.

2.2 Аналіз вимог

Аддон для Blender повинен відповідати ряду важливих критеріїв, щоб забезпечити ефективну і зручну оптимізацію геометрії 3D моделей. Аддон розроблений з метою оптимізації петель у мішах, надаючи користувачам можливість вибирати або видаляти ребра відповідно до заданих параметрів кутів та інших критеріїв. Нижче наведено функціональні та нефункціональні вимоги для аддону FunkyLoop Optimizer для Blender.

Аддон повинен відповідати наступним функціональним вимогам:

- оптимізовувати виділені петлі в меші, видаляючи чи виділяючи ребра;
- користувач повинен мати можливість встановлювати мінімальний та максимальний кути для видалення петель;
- аддон повинен ігнорувати петлі, які з'єднані менш ніж з двома іншими петлями;
- після оптимізації петель користувач повинен мати можливість автоматично виділити їх для подальшого редагування;
- повина бути панель керування в 3D View, де користувач може налаштувати параметри та запустити оптимізацію;
- редагування повинно функціонувати в режимі редагування меша (Edit Mode).

Нефункціональні вимоги:

- інтерфейс має бути інтуїтивно зрозумілим та легким у використанні. Він має бути доступний через меню та панель інструментів для зручності користувачів;
- аддон повинен бути сумісним із останніми версіями Blender, забезпечуючи його доступність та використання в актуальних робочих процесах;
- аддон повинен запобігати помилкам та некоректній поведінці, забезпечуючи стабільну роботу та безпеку даних користувача;
- код аддону має бути гнучким та легко розширюваним для додавання нових функціональностей без значних змін.

2.3 Архітектура системи

Архітектура системи визначає, як аддон взаємодіє з користувачем, як виконує операції та як використовує дані Blender для досягнення своїх цілей (рис. 2.1).

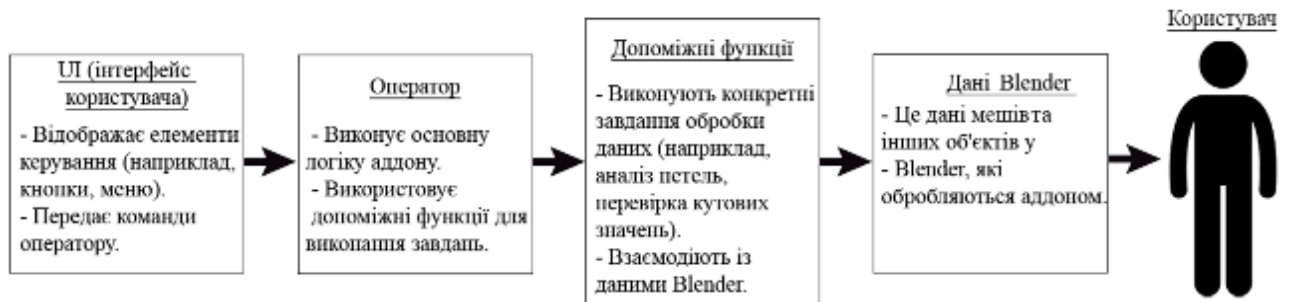


Рисунок 2.1 –Архітектура системи

Розглянемо основні компоненти архітектури та їх роль у процесі роботи з Blender:

- UI аддону є важливою частиною його функціоналу, він надає користувачеві зручні елементи керування, такі як кнопки, меню та поля введення, дозволяючи їм легко взаємодіяти з функціями аддону, Добре спроектований та інтуїтивно зрозумілий інтерфейс допомагає користувачам швидко освоїтися з новим функціоналом та ефективно використовувати його для своїх цілей;

- оператори є ключовими компонентами аддону, оскільки вони відповідають за виконання конкретних операцій або завдань, які користувач запитує, також оператори можуть містити складні алгоритми і логіку, дозволяючи аддону виконувати різноманітні завдання, такі як оптимізація моделей, створення анімацій або генерація текстур;

- допоміжні функції надають додаткову функціональність виконання операцій операторів, вони можуть включати алгоритми обробки даних, перевірки умов або взаємодію із зовнішніми джерелами даних, допоміжні функції допомагають розділити логіку аддона на більш дрібні частини, що перевикористовуються, що робить код чистішим і підтримуваним;

- для виконання своїх завдань аддони взаємодіють із даними Blender, такими як геометрія моделей, налаштування сцен та параметри анімації, ці дані надають аддонам доступ до вмісту сцен та об'єктів у Blender,

дозволяючи їм змінювати та покращувати моделі, анімації та інші аспекти проєктів користувачів;

- користувач отримує результат при взаємодії з аддоном.

Правильно спроектований інтерфейс та ефективна логіка виконання операцій роблять аддон зручним інструментом для користування.

2.4 Вибір інструментарію розробки

Для написання програми обрано мову програмування Python та середовище швидкої розробки додатків Blender API.

Середовищем функціонування програмного продукту обрана операційна система Windows 11 сімейства Windows NT корпорації Microsoft [20]. На сьогоднішній день – Windows найбільш поширена операційна система виробництва Microsoft. Основними причинами вибору саме цієї операційної системи є:

- стабільність роботи системи;
- зручний інтерфейс;
- режим мультизадачної роботи;
- популярність використання.

Також необхідно відмітити, що Windows 11 має точки відновлення, тому в разі збою в роботі системи можливе повне відновлення програм, документів та інших даних, система є багатофункціональною та при встановленні має стандартний пакет драйверів [20].

Python — високорівнева мова програмування загального призначення з динамічною строгою типізацією та автоматичним управлінням пам'яттю, орієнтована підвищення продуктивності розробника, читання коду та його якості, і навіть забезпечення переносимості написаних у ньому програм. [21].

Мова є повністю об'єктно-орієнтованим у цьому плані, що це об'єктами. Незвичайною особливістю мови є виділення блоків коду

відступами. Синтаксис ядра мови мінімалістичний, рахунок чого практично рідко виникає необхідність звертатися до документації [21].

Сама ж мова відома як інтерпретована і використовується в тому числі для написання скриптів. Недоліками мови є найчастіше нижча швидкість роботи і більш високе споживання пам'яті написаних нею програм проти аналогічним кодом, написаним компілюваних мовами, як-от С чи С++.

Python відомий своїм простим та читаним синтаксисом. Він мінімалістичний і інтуїтивно зрозумілим. Основні принципи Python полягають у тому, щоб код був легким і зрозумілим [21].

С, одна з найстаріших мов програмування, відома своєю простотою і низькорівневим управлінням пам'яттю. Його синтаксис лаконічний, але вимагає від розробника уважного ставлення до деталей, особливо управління пам'яттю. С широко використовується для системного програмування, створення операційних систем та вбудованих систем [22].

С++ є розширенням мови С і підтримує як процедурне, так і об'єктно-орієнтоване програмування. Його синтаксис складний і потужний, що дозволяє створювати як високорівневі, і низькорівневі програми. С++ широко використовується для системного програмування, розробки ігор та високопродуктивних додатків [23].

Python інтерпретована мова, що робить його повільнішим порівняно з компілюваними мовами програмування. Він не підходить для завдань, які потребують високої продуктивності, таких як графічні програми або системи реального часу. Однак завдяки бібліотекам, написаним на С та С++, Python може бути використаний для обробки даних та наукових обчислень. С компілюється безпосередньо в машинний код, що забезпечує високу продуктивність. Це робить його ідеальним вибором для системного програмування та систем. С++ також компілюється в машинний код і відомий своєю високою продуктивністю, що робить його популярним для розробки ігор та інших програм, що потребують високої швидкості

виконання. Python універсальний і широко використовується в різних галузях: веб-розробка (Django, Flask), наукові обчислення та обробка даних (NumPy, pandas, SciPy), машинне навчання (TensorFlow, PyTorch), автоматизація та скриптинг, розробка ігор (Pygame), що вбудовуються системи (MicroPython). С застосовується в системному програмуванні (операційні системи, драйвери), системах, що вбудовуються, розробці мікроконтролерів і критично важливих додатках, де важлива передбачувана продуктивність. C++ використовується у тих галузях, де потрібна висока продуктивність: системне програмування, розробка ігор (Unreal Engine), фінансове програмування (трейдингові системи), вбудовані системи, обробка графіки (OpenGL, DirectX) [21]-[23].

Python підтримується безліччю інструментів та інтегрованих середовищ розробки (IDE), таких як PyCharm, VS Code, Jupyter Notebook та багато інших. Ці інструменти значно полегшують процес розробки, налагодження та тестування програм. Завдяки їм розробники можуть працювати більш ефективно, використовуючи різноманітні функції, такі як автодоповнення, налагодження на льоту та візуалізація даних [24].

Що стосується мови програмування C, існує безліч компіляторів та інструментів, таких як GCC, Clang, Visual Studio та Code::Blocks. Ці інструменти надають розробникам потужні засоби для написання, налагодження та оптимізації коду мовою C [22].

Так само, для мови C++ існує безліч IDE та компіляторів, включаючи Visual Studio, CLion, GCC та інші. Ці інструменти також мають різноманітні функції, які допомагають розробникам у створенні, налагодженні та оптимізації програмного коду [23].

Загалом, різноманітність інструментів для роботи з Python, C і C++ дозволяє розробникам вибирати ті, які найкраще відповідають їхнім потребам та перевагам, полегшуючи процес розробки програмного забезпечення.

Таблиця 2.1 – Порівняння мов програмування [21]-[23]

Критерій	Python	C	C++
Синтаксис	Простий та легкий для читання, використовує відступи для блоків коду	Короткий, вимагає управління пам'яттю, використовує фігурні дужки	Складний, підтримує процедурне та об'єктно-орієнтоване програмування, використовує фігурні дужки
Продуктивність	Інтерпретований, повільніше компілюваних мов	Висока, компілюється в машинний код	Висока, компілюється в машинний код
Області застосування	Веброзробка, наукові обчислення, машинне навчання, автоматизація, розробка ігор, системи, що вбудовуються	Системне програмування, вбудовані системи, розробка мікроконтролерів	Системне програмування, розробка ігор, фінансове програмування, вбудовані системи, обробка графіки
Інструментальні засоби	PyCharm, VS Code, Jupyter Notebook	GCC, Clang, Visual Studio, Code::Blocks	VisualStudio, CLion, GCC
Затребуваність на ринку праці	Висока в областях аналізу даних, машинного навчання та веб-розробки	Висока в областях аналізу даних, машинного навчання та веб-розробки	Висока у розробці ігор, системному програмуванні, вбудованих системах та фінансовому секторі

Для створення аддонів для Blender, Python є найбільш підходящою мовою з кількох причин. По-перше, сам Blender надає вбудовану підтримку Python, що дозволяє розробникам легко інтегрувати свої скрипти та розширення. Python має простий і читаний синтаксис, що прискорює процес розробки та налагодження аддонів. Завдяки широкій стандартній бібліотеці та безлічі сторонніх модулів, Python спрощує виконання складних завдань, таких як обробка даних, мережна взаємодія та робота з різними форматами файлів. Нарешті, активна спільнота Python-розробників та безліч доступних

ресурсів полегшують отримання підтримки та обмін знаннями, що особливо важливо для розробки якісних та функціональних аддонів для Blender.

2.5 Методика розробки аддона

У розробці аддону для Blender кожен крок відіграє ключову роль у створенні функціонального та зручного інструменту для користувачів. Від створення файлу та імпортування необхідних модулів до визначення основної функції та реєстрації оператора в інтерфейсі Blender – кожен етап потребує уважного підходу та глибокого розуміння процесу. У даному контексті розробка аддону для Blender, який автоматизує процес оптимізації поліганльної сітки шляхом видалення зайвих петель на основі кутового порогу, є важливим дослідженням у галузі програмування та 3D-моделювання.

На рисунку 2.2 наведено методику створення такого аддону, починаючи зі створення файлу та імпортування необхідних модулів, і закінчуючи реєстрацією та інтеграцією аддону в інтерфейс Blender. Розглянемо детальніше структурна схема створення кода для аддона.

Етап 1. Створення файлу аддону. Для початку створіть новий файл у каталозі addons Blender з розширенням .py, цей файл міститиме весь код вашого аддону. а на самому початку файлу потрібно додати метадані, такі як назва аддону, ім'я автора, версія та інформація про сумісність з різними версіями Blender, це важливо для того, щоб Blender міг розпізнати і правильно обробити ваш аддон.

Етап 2. Імпорт необхідних модулів. Після додавання метаданих імпортуйте необхідні модулі для роботи з Blender та його API, важливо імпортувати модулі bpy (основний модуль Blender Python API) та bmesh (для роботи з мешами), щоб отримати доступ до всіх необхідних функцій та класів.

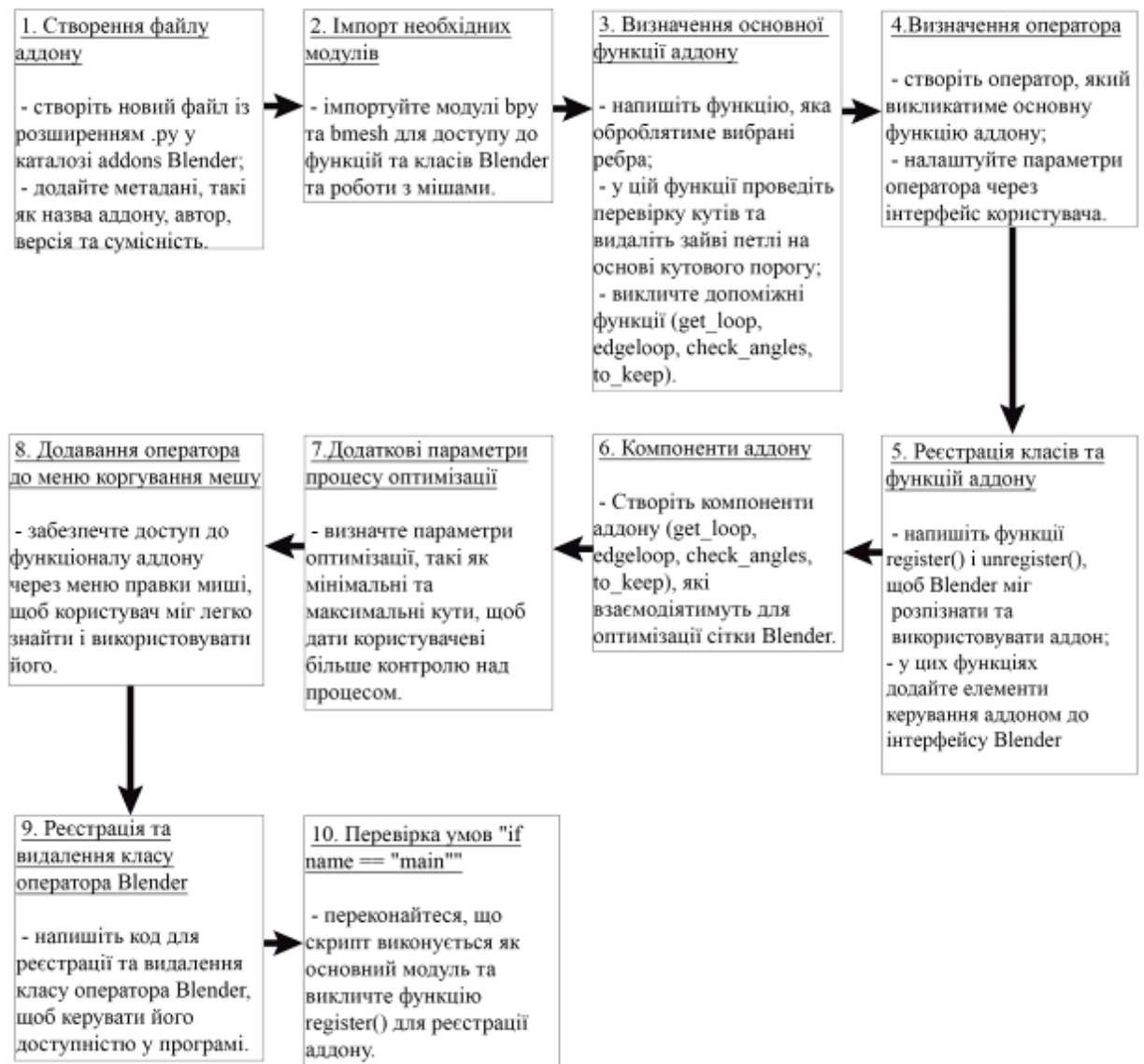


Рисунок 2.2 – Методика створення аддона

Етап 3. Визначення основної функції аддону. Створіть основну функцію аддону, яка виконуватиме основні операції. Саме у цій функції відбуватиметься обробка вибраних ребер: перевірка кутів та видалення зайвих петель на основі кутового порога. Викликайте допоміжні функції, які допоможуть в обробці меша.

Етап 4. Визначення оператора. Створіть оператор Blender, який викликатиме вашу основну функцію аддону. Оператор повинен мати параметри, які можна налаштовувати через інтерфейс, щоб користувачі могли керувати процесом оптимізації.

Етап 5. Реєстрація класів на функції аддону. Напишіть функції `register()` та `unregister()`, які будуть реєструвати та видаляти класи та функції вашого аддону, це дозволить Blender розпізнати та використовувати ваш аддон, а також додасть елементи управління в інтерфейс програми.

Етап 6. Компоненти аддону. Такі функції як роботи з петлями ребер і перевірки кутів. Ці компоненти взаємодіятимуть між собою для оптимізації сітки в Blender.

Етап 7. Додаткові параметри процесу оптимізації. Визначте додаткові параметри оптимізації, такі як мінімальні та максимальні кути, які допоможуть вам контролювати процес видалення ребер.

Етап 8. Додавання оператора до меню корегування мешу. Щоб зробити аддон більш доступним для користувачів, додайте його в меню редагування мешу, це зробить його легко доступним і дозволить користувачам використовувати його у потрібний момент.

Етап 9. Реєстрація та видалення класу оператора Blender. Напишіть код для реєстрації та видалення класу оператора Blender. Це дозволить вам керувати доступністю вашого оператора в програмі та забезпечить його правильну роботу.

Етап 10. Перевірка умов `"if name == 'main'"`. Переконайтеся, щоб ваш аддон правильно реєструється під час запуску, використовуйте умову `"if name == 'main'"` для цієї перевірки та виклику функції `register()`.

Це основні кроки, які ви повинні виконати для розробки вашого аддону у Blender. Переконайтеся, що кожен крок виконується уважно, щоб ваш аддон працював коректно і легко доступний для користувачів.

Окрім цього, можливо, вам знадобляться додаткові модулі, такі як `math` для роботи з кутами та `bpy.props` для створення налаштувань аддону.

Ключові функції:

- ініціалізація: в цій частині коду ви створюєте класи операторів та панелей, які будуть використовуватись в аддоні;

- клас оператора: цей клас відповідає за виконання основної логіки оптимізації мешу;
- панель інтерфейсу: цей клас відповідає за створення користувацького інтерфейсу в Blender, де користувач зможе взаємодіяти з аддоном;
- в кінці файлу необхідно зареєструвати всі створені класи, щоб Blender міг використовувати їх у своїй роботі.

Цей аддон дозволяє оптимізувати меш, видаляючи зайві петлі на основі кутового порогу. За допомогою метаданих, імпорту модулів, створення класів операторів та панелей, а також їх реєстрації, ви можете створити ефективний інструмент для роботи з мешами в Blender.

Щоб Blender розпізнав аддон, необхідно зареєструвати всі класи та функції аддону. Це робиться за допомогою функції `register()`, а скасування реєстрації – за допомогою функції `unregister()`. Ці функції також додають елементи керування аддоном в інтерфейс Blender, наприклад меню редагування мешу.

Аддон складається з кількох компонентів, що взаємодіють для оптимізації сітки Blender. Інші компоненти, такі як `get_loop`, `edgeloop`, `check_angles`, `to_keep`, виконують допоміжні функції, необхідні виявлення і обробки петель в меші. Ці компоненти взаємодіють між собою ефективного виконання завдання оптимізації.

Функція `to_keep`, ймовірно, призначена для майбутніх покращень, коли будуть додані додаткові умови для збереження ребер. Функція `deselect` знімає виділення з усіх вершин об'єкта в меші, забезпечуючи, що наступні операції будуть виконані на невиділеній сітці.

Функція `get_loops` знаходить усі петлі ребер у меші, що відповідають заданим умовам. Вона обходить всі вибрані ребра, перевіряючи їхню відповідність умовам і кутовим порогам, а потім збирає петлі за допомогою функції `get_loop`. Функція `get_neighbours` визначає сусідні петлі кожної петлі

у списку петель. Вона використовує грані визначення сусідніх ребер і, відповідно, петель.

Функція `skiploop` додає петлю та її сусідів до списків для подальшої обробки. Вона обходить сусідні петлі, щоб зібрати всі пов'язані петлі для фінальної обробки. Функція `optiloops` є основною функцією, яка виконує оптимізацію сітки. Вона оновлює нормалі сітки, перемикає режим виділення на ребра, знаходить усі ребра, які мають бути видалені на основі заданих кутових порогів, обробляє петлі, перевіряє кути, визначає сусідів та залежно від налаштувань вибирає чи видаляє петлі.

Клас `OPTILOOPS_OT_operator` є основним оператором, який викликає функцію `optiloops`. Він містить атрибути та методи налаштування та оптимізації. Метод `invoke` викликається при запуску оператора, перевіряє приховані властивості та запускає оптимізацію. Метод `execute` виконує основну дію оператора, викликаючи `optiloops`. Метод `draw` відображає інтерфейс користувача в панелі інструментів, дозволяючи користувачу налаштовувати параметри оптимізації.

Додаткові властивості оператора служать для точного настроювання процесу оптимізації моделі. Наприклад, мінімальні та максимальні кути можуть бути задані для визначення того, які петлі слід видалити, а які залишити. Це дозволяє користувачеві гнучкіше контролювати геометрію моделі відповідно до його потреб. Крім того, облік петель із менш ніж двома сусідами допомагає виключити з розгляду небажані елементи моделі під час оптимізації.

Щоб зробити оператор доступним у меню редагування миші, використовується функція `optiloops_menu`, яка автоматично додає оператора у відповідне місце інтерфейсу Blender. Це полегшує користувачеві доступ до функціонала та робить його більш інтуїтивно зрозумілим для використання.

Функції `register` та `unregister`, у свою чергу, забезпечують реєстрацію та видалення класу оператора у Blender. Це дозволяє ефективно керувати його доступністю в програмі, роблячи його доступним для використання в

потрібний момент і приховуючи, коли він не потрібний, що допомагає оптимізувати інтерфейс Blender для користувача. Умова `"if name == 'main'"` використовується для перевірки, чи виконується скрипт як основний модуль, і при необхідності викликає функцію `register` для реєстрації аддону. Цей підхід дозволяє тестувати аддон, запускаючи його без необхідності встановлення у Blender.

Загалом цей аддон надає користувачам потужні інструменти для оптимізації сіток, спрощуючи процес видалення зайвих ребер та підтримання чистоти геометрії моделі в Blender. Після реалізації аддону важливо ретельно протестувати його, щоб переконатися, що всі функції та оператори працюють коректно у різних сценаріях використання. Тестування допоможе виявити та виправити можливі помилки, а також покращити загальну продуктивність та зручність використання аддону.

Крім того, для покращення аддону можна розглянути додавання додаткових функцій, таких як можливість збереження оптимізації налаштувань для подальшого використання або інтеграція з іншими інструментами Blender. Регулярне оновлення аддону з урахуванням зворотного зв'язку від користувачів також допоможе підтримувати його актуальність та відповідність сучасним вимогам та стандартам.

2.6 Висновки до розділу 2

Отже, після ретельного аналізу розділу 2, можна зробити висновок, що розробка аддону для Blender, спрямоване на оптимізацію полігонів в 3D моделей, орієнтований на автоматизацію різних аспектів роботи з 3D моделями, таких як:

- ретопології;
- оптимізації петель;
- структуризації моделей.

У цьому аддоні варто виділити декілька ключових функцій, які відповідають за реалізацію зазначених завдань. Серед них можна виокремити автоматичне видалення непотрібних ребер, яке допомагає спростити модель та зменшити її об'єм, що є важливим для оптимізації продуктивності обробки. Також до ключових функцій відноситься покращення структури петель, яке спрямоване на збільшення ефективності роботи з геометрією.

Однією з головних переваг аддону є можливість оптимізувати ретопології, що є важливим етапом в роботі з 3D моделями. Також користувачеві доступні інші налаштування параметрів оптимізації, які дозволяють забезпечити більш точну та індивідуальну обробку моделі залежно від конкретних потреб.

Аддон розроблено мовою програмування Python з використанням Blender API. Вибір саме цих інструментів обумовлений їхньою ефективністю та популярністю серед розробників. Python володіє простим синтаксисом та широким спектром функціоналу, що сприяє швидкому розвитку програм та дозволяє легко взаємодіяти з Blender API для створення потрібного функціоналу.

3 КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Схема алгоритму функціонування програмного забезпечення

На рисунку 3.1 наведено схему алгоритму функціонування програми.

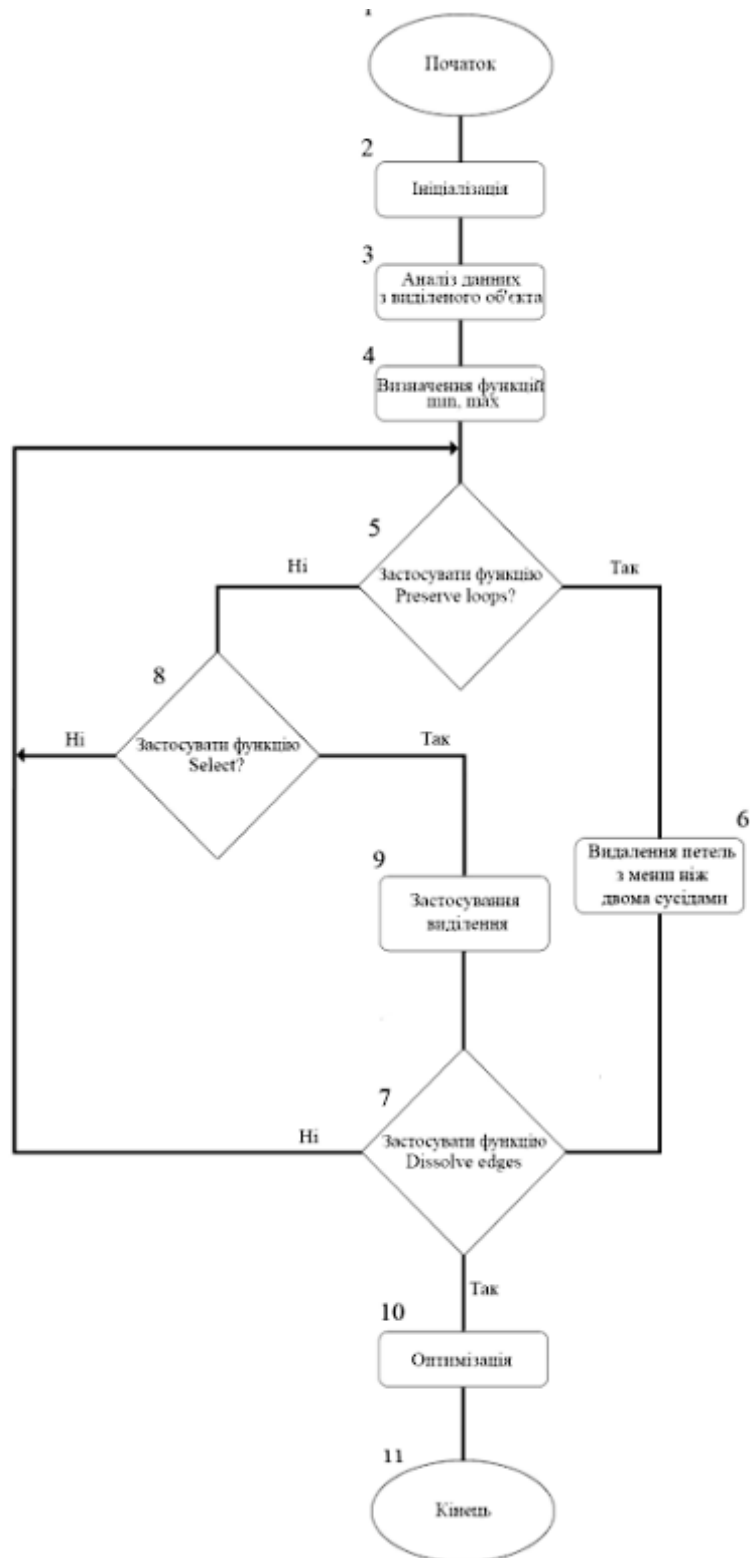


Рисунок 3.1 – Схема алгоритму функціонування програми

Функціонування аддона розпочинається з ініціалізації, де завантажується та реєструється аддон. Потім програма отримує дані про активний об'єкт і його сітку, готуючи bmesh для редагування.

Блок 1. Початок. Цей блок являє собою початок виконання програми або скрипта.

Блок 2. Ініціалізація. У цьому блоці відбувається ініціалізація необхідних змінних, об'єктів та параметрів перед початком виконання основного алгоритму.

Блок 3. Аналіз даних із виділеного об'єкта. Тут відбувається аналіз даних, які відносяться до виділеного об'єкта у середовищі Blender. У разі представленого скрипту це, ймовірно, означає аналіз даних міша (геометрії), які будуть оптимізовані.

Блок 4. Визначення функцій min і max. Цей блок включає визначення функцій min і max, ймовірно, для встановлення мінімального та максимального кутів, які будуть використовуватися в процесі оптимізації.

Блок 5. Застосувати функцію Preserve Loops. Цей блок включає рішення про те, чи використовуватиметься функція Preserve Loops (Зберегти петлі). У контексті скрипту це, ймовірно, опція, яка вказує, чи буде збережено петлі з певними характеристиками.

Блок 6. Видалення петель з менш ніж двома сусідніми. У цьому блоці виконується видалення петель, які мають менше двох сусідніх петель. Це може бути частиною оптимізації для видалення "ізолюваних" петель.

Блок 7. Застосувати функцію Dissolve Edges. Цей блок включає застосування функції Dissolve Edges (Розчин ребер), що, ймовірно, означає видалення деяких ребер у меші для оптимізації його структури.

Блок 8. Застосувати функцію Select. Цей блок є застосуванням функції Select (Виділення), що, ймовірно, означає виділення певних елементів міша після застосування оптимізації.

Блок 9. Застосування виділення. У цьому блоці застосовується виділення, яке було встановлене в попередньому блоці, що, ймовірно, означає оновлення виділення в інтерфейсі користувача Blender.

Блок 10. Оптимізація. Цей блок є основним процесом оптимізації, який виконується відповідно до попередніх кроків і параметрів, встановлених у попередніх блоках.

Блок 11. Кінець. Цей блок являє собою завершення виконання програми або скрипта.

3.2 Опис особливостей програмної реалізації

Процес починається з ініціалізації, де аддон завантажується та реєструється у Blender. Це включає кілька кроків: спочатку відбувається завантаження файлів аддону, потім здійснюється реєстрація класів і операторів, необхідних для його роботи. Після цього налаштовується інтерфейс користувача, включаючи додавання нових елементів, таких як панелі, кнопки та меню.

Процес оптимізації сітки в коді починається з отримання доступу до даних сітки активного об'єкта Blender. Спочатку відбувається ініціалізація аддону, де завантажується та реєструється код аддону. Цей процес включає визначення необхідних класів і операторів, а також налаштування інтерфейсу для зручної взаємодії з аддоном.

Коли аддон ініціалізований, відбувається вибір активного об'єкта сітки, який потрібно оптимізувати. Далі сітка об'єкта перетворюється на об'єкт BMesh з використанням модуля bmesh, що дозволяє проводити операції на рівні ребер та вершин.

Потім розпочинається процес оптимізації сітки. Функція `get_loops` використовується для пошуку петель ребер сітки. Кожна петля ребер є групою ребер, пов'язаних між собою. Це дозволяє виділити області сітки, які можуть бути оптимізовані.

Після отримання петель ребер сітки відбувається перевірка кутів між ребрами за допомогою функції `check_angles`. Для кожної петлі ребер сітки перевіряється, чи всі ребра знаходяться в межах кутів, заданих користувачем. Якщо кути відповідають заданим параметрам, петля ребер вважається підходящою для оптимізації.

Потім відбувається оптимізація сітки. Ребра, які відповідають умовам оптимізації можуть бути розчинені за допомогою функції `dissolve_edges`. Це призводить до зменшення кількості ребер у сітці та покращує її продуктивність. Після завершення оптимізації сітки відбувається оновлення сцени для відображення змін. Область сітки може автоматично виділятися після оптимізації, якщо вибрано відповідну опцію. Весь цей процес автоматично оптимізує сітку моделі відповідно до заданих параметрів, що сприяє покращенню продуктивності та якості моделювання у Blender.

Для більш глибокого розуміння процесу оптимізації сітки у Blender, розглянемо основні функції, оператори та змінні, що використовуються у коді. Для того щоб систематизувати наше розуміння, давайте розглянемо основні функції та методи, які застосовуються в коді аддону у таблиці 3.1.

Таблиця 3.1 – Таблиця опису функцій

Функції	Опис
1	2
<code>OPTIMIZE_MESH_OT_OptimizeMesh</code>	Оператор Blender виконує оптимізацію сітки.
<code>execute</code>	Метод, який викликається під час виконання оператора. Запускає процес оптимізації із викликом функції <code>optiloops</code> .
<code>optiloops</code>	Основна функція оптимізації сітки. Отримує доступ до сітки, шукає петлі ребер та проводить оптимізацію.

Продовження таблиці 3.1

1	2
get_loops	Функція пошуку петель сітки.
skiploop	Функція пропуску оптимізованих петель.
skiploop	Функція пропуску оптимізованих петель.
to_keep	Функція визначення, чи потрібно зберігати певні ребра у процесі оптимізації.
deselect	Функція для зняття виділення з ребер та вершин.
get_loop	Допоміжна функція отримання петлі ребер.
check_angles	Функція перевірки кутів між ребрами сітки.
get_neighbours	Функція визначення сусідів петель ребер.

Змінні відіграють важливу роль передачі даних, зберіганні інформації та управлінні процесом оптимізації. Розглянемо основні змінні, які використовуються в процесі оптимізації сітки в Blender в Таблиці 3.2.

Таблиця 3.2 – Таблиця опису змінних

Змінні	Опис
1	2
dissolve_edges	Бульова змінна для визначення, чи слід розчиняти ребра у процесі оптимізації.

Продовження таблиці 3.2

1	2
select_after_dissolve	Бульова змінна для визначення, чи потрібно автоматично виділяти область сітки після оптимізації.
save_form	Бульова змінна визначення, чи потрібно зберігати форму моделі у процесі оптимізації.
min_angle	Змінна для завдання мінімального порогу при оптимізації ребер..
max_angle	Змінна для завдання максимального порога кута під час оптимізації ребер.
sel	Безліч ребер сітки, які були вибрані для оптимізації.
sel_copy	Копія вибраних ребер сітки.
final_loops	Список оптимізованих петель ребер.
skip_loops	Список пропущених петель ребер.

Насамкінець, аддон демонструє застосування різних функцій, методів та змінних для ефективного виконання оптимізації сітки, що важливо для забезпечення високої продуктивності та якості роботи з моделями.

Глибоке розуміння роботи цього аддону дозволяє моделювальникам покращити свої навички та зробити процес моделювання більш ефективним та зручним.

3.3 Висновки до розділу 3

Після ретельного аналізу розділу 3 можна зробити висновок щодо функціонування аддону для Blender, спрямованого на оптимізацію сіток 3D моделей.

Було проаналізовано та вивчено послідовністю кроків побудову коду аддона що зображену на рисунку 3.1. Кожен блок виконує певну функцію у процесі оптимізації, та використовуються для оптимізації геометрії об'єкта в середовищі Blender.

Цей матеріал детально освітлює алгоритм роботи аддону, починаючи з ініціалізації та реєстрації його в середовищі Blender, а закінчуючи процесом оптимізації сіток. Розглянуті ключові етапи алгоритму включають аналіз даних об'єкта, визначення функцій та застосування різних операцій для оптимізації сіток.

Компоненти коду, які були представлені у таблиці 3.1 та таблиці 3.2, розкривають основні функції та методи, використовувані під час оптимізації сіток. Ці компоненти включають в себе оператори, методи та змінні, які активно використовуються для обробки та оптимізації геометрії моделей.

Аналіз цього матеріалу дозволяє глибше зрозуміти принципи роботи аддону, його основні функції та роль у покращенні процесу моделювання у Blender.

Насамкінець, аддон демонструє застосування різних функцій, методів та змінних для ефективного виконання оптимізації сітки, що важливо для забезпечення високої продуктивності та якості роботи з моделями.

4 ІНСТРУКЦІЯ З ВИКОРИСТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Практичне застосування аддона

Для того, щоб застосувати аддон у Blender, необхідно виконати кілька кроків.

Запустіть програму Blender на комп'ютері. Це можна зробити, двічі клацнувши на ярлику програми або здійснити пошук у меню "Пуск" (Windows) або в Launchpad (MacOS).

Коли Blender запущено, ви побачите верхню панель з різними меню. Знайдіть та клацніть по меню "Edit", яке зазвичай розташоване у лівому верхньому кутку екрана. Це відкриє меню, що випадає, з різними опціями.

Наочно побачити меню "Edit" можна на рисунку 4.1.

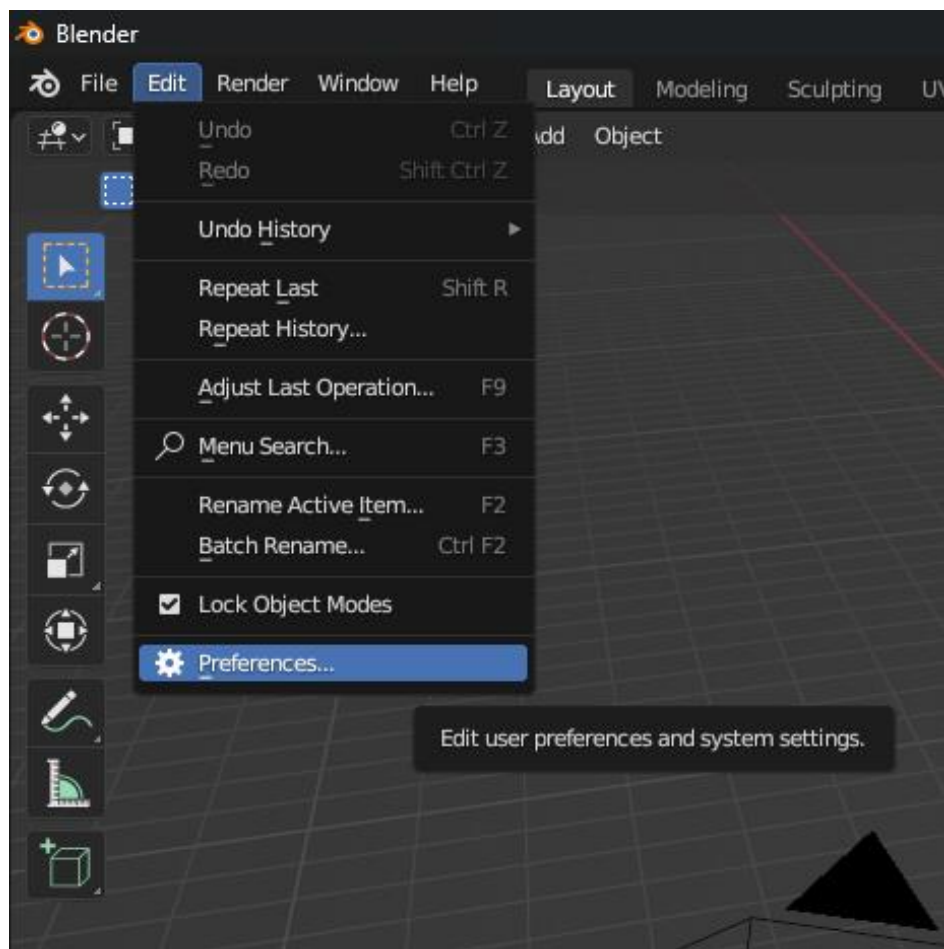


Рисунок 4.1 – Меню "Edit"

У меню "Edit" знайдіть опцію "Preferences" і клацніть по ній. "Preferences" зображено на рисунку 4.2. Далі перейдіть на вкладку "Addons" і натисніть кнопку "Встановити" (Install). У діалоговому вікні виберіть архів з вашим аддоном і натисніть (Install Add-on from File). Зазвичай він знаходиться у лівій частині вікна. Цей розділ містить список усіх доступних аддонів для Blender.

Після цього аддон з'явиться у списку встановлених аддонів, і його можна буде увімкнути, поставивши галочку поряд із назвою.

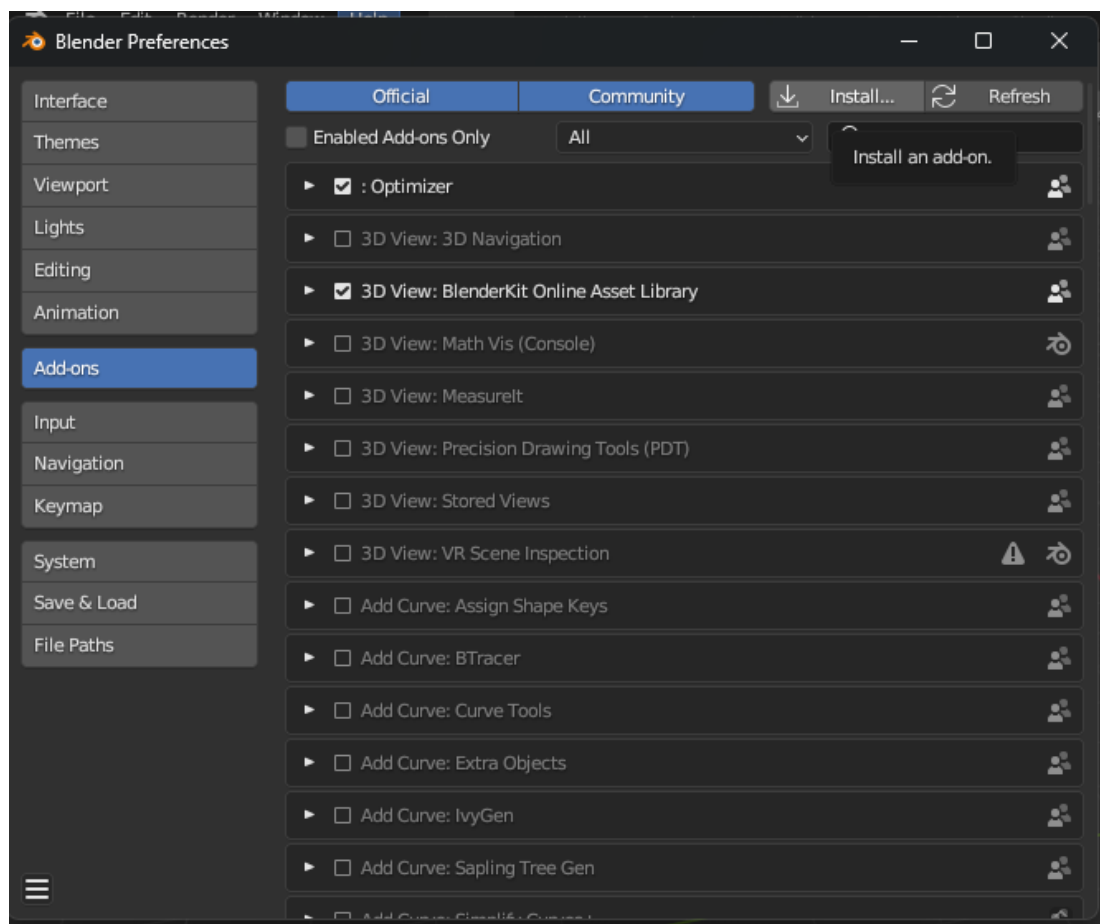


Рисунок 4.2 – Відкрите вікно розділу "Preferences"

Тепер, після встановлення аддону знайдіть його у списку доступних аддонів у розділі "Add-ons". Переконайтеся, що поруч із назвою аддону стоїть галочка, що означає його активацію. Якщо галочка не встановлена, просто клацніть на ній.

Після активації аддону закрийте вікно налаштувань, натиснувши кнопку "Save Preferences" у правому нижньому кутку вікна. Тепер аддон успішно встановлений і готовий до використання у Blender.

Нижче наведено послідовний план дій, який допоможе вам оптимізувати полігони у вашому проєкті з використанням Blender:

Запустіть Blender і створіть новий файл проєкту. Це можна зробити, запустивши Blender і вибравши опцію "New" або "Open " у меню "File".

На рисунку 4.3 зображено меню "File".

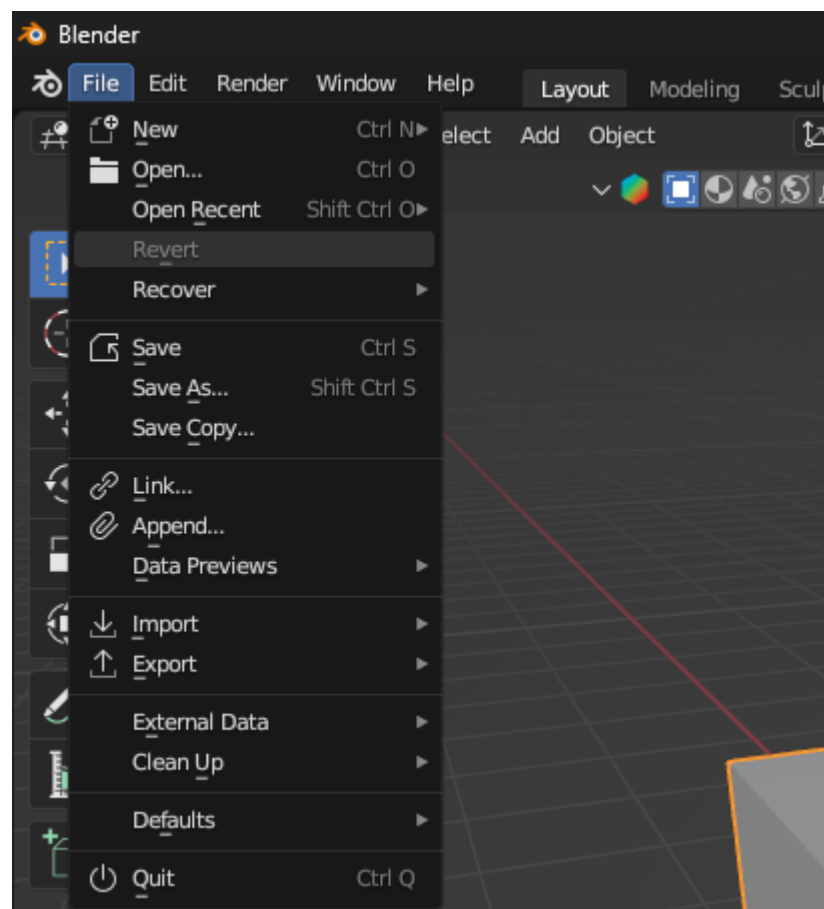


Рисунок 4.3 – Меню "File"

Після створення нового проєкту додайте об'єкт, який ви будете оптимізувати, натиснувши комбінацію клавіш Shift + A або виберіть "Add" (що зображено на рисунку 4.4) у верхньому меню, потім оберіть "Mesh", а після цього "Cube", щоб створити куб, який буде використовуватися для оптимізації.

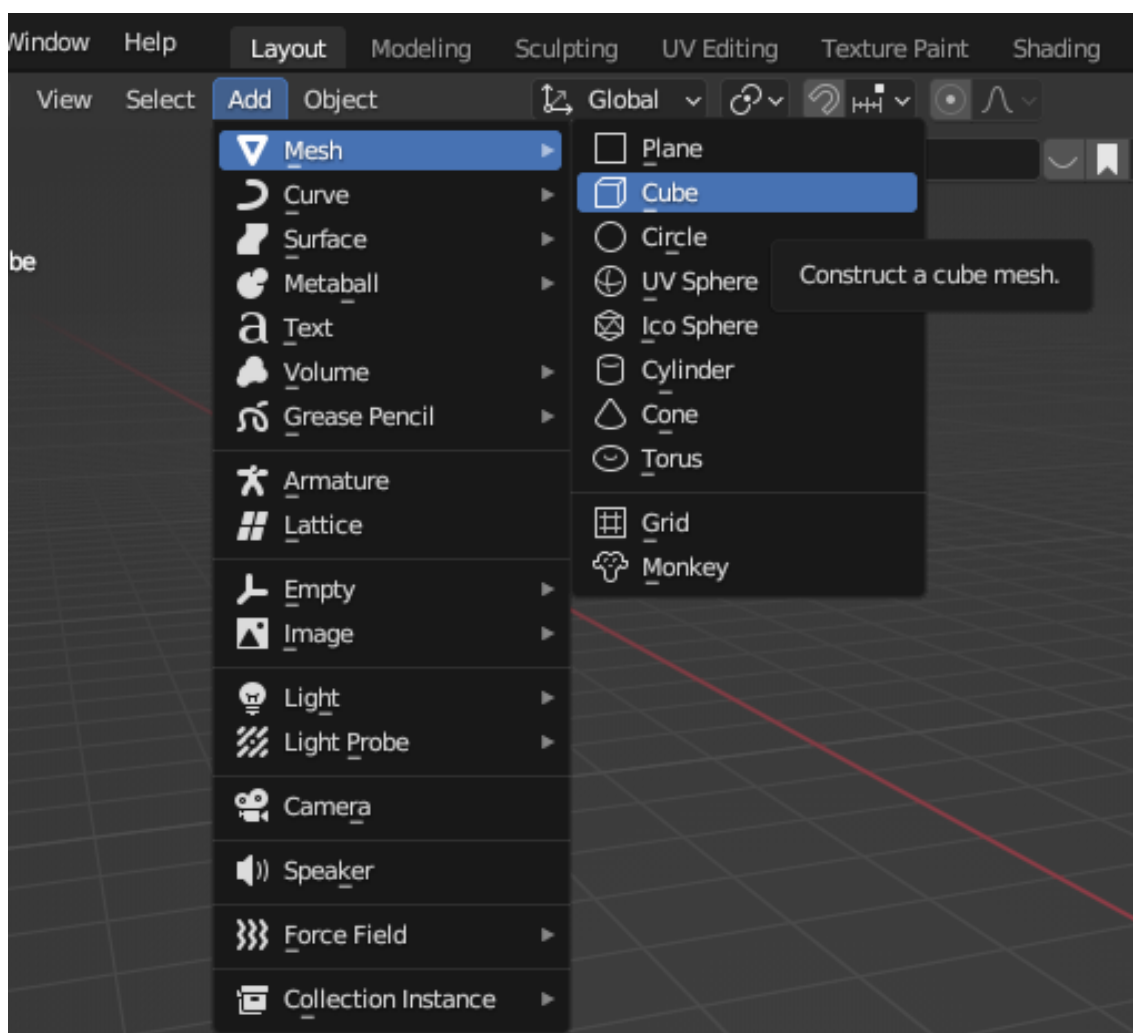


Рисунок 4.4 – Вкладка "Add"

Для переходу в режим редагування в Blender. У вікні 3D Viewport виберіть куб, який ви хочете редагувати. Після вибору об'єкта перейдіть до режиму редагування. Це можна зробити двома способами

натисніть клавішу "Tab" на клавіатурі – це швидкий спосіб перемикання між режимами об'єкта (об'єктний режим і режим редагування). Або виберіть "Edit Mode" у верхньому лівому куті вікна Blender. Це робить те саме - переходить в режим редагування обраного об'єкта.

Після виконання цих кроків ви будете перебувати в режимі редагування обраного об'єкта і зможете застосовувати аддон

Щоб викликати аддон для початку натисніть клавішу "N", щоб відкрити бічну панель, якщо вона ще не відкрита. У боковій панелі перейдіть

на вкладку "Optimizer". Далі побачите панель з назвою "Loop Optimizer". Як саме виглядає "Loop Optimizer" можна побачити на рисунку 4.5.

Усередині цієї панелі має бути кнопка "Optimize". Далі натисніть кнопку, щоб активувати аддон і оптимізувати вибрану сітку.

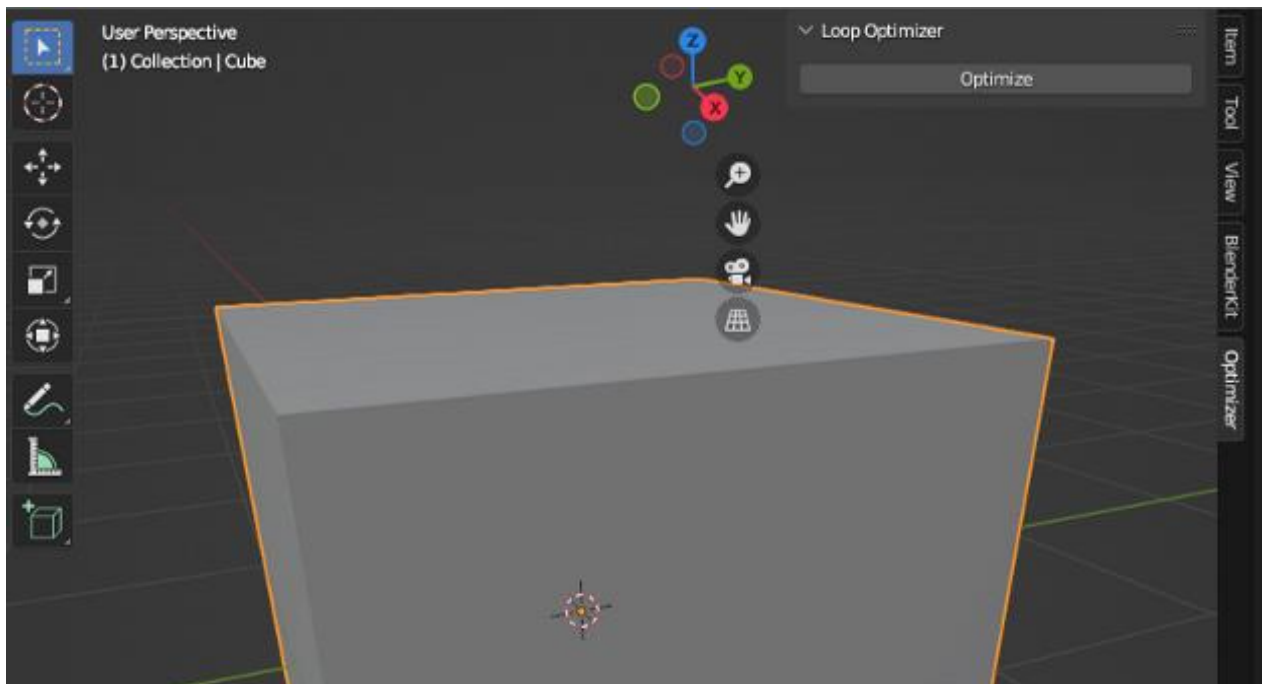


Рисунок 4.5 – Меню виклика аддона "Loop Optimizer"

Після застосування аддон, вам потрібно буде налаштувати його параметри. Це можна зробити через відповідний інтерфейс в лівому нижньому кутку. Інтерфейс можна подивитись на рисунку 4.6.

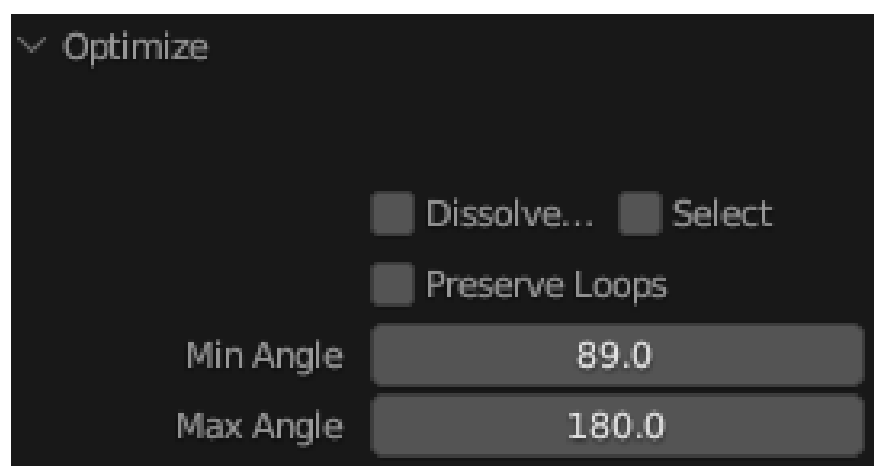


Рисунок 4.6 – Меню аддона " Optimize"

Це був загальний огляд того, як викликати меню аддона(рисунок 4.7) у Blender для оптимізації полігонів. Тепер давайте розберемо, як працює аддон для Blender, спрямований на оптимізацію полігонів. Його функціонал дозволяє автоматизувати процес оптимізації моделей, включаючи зведення полігонів, видалення непотрібних деталей.

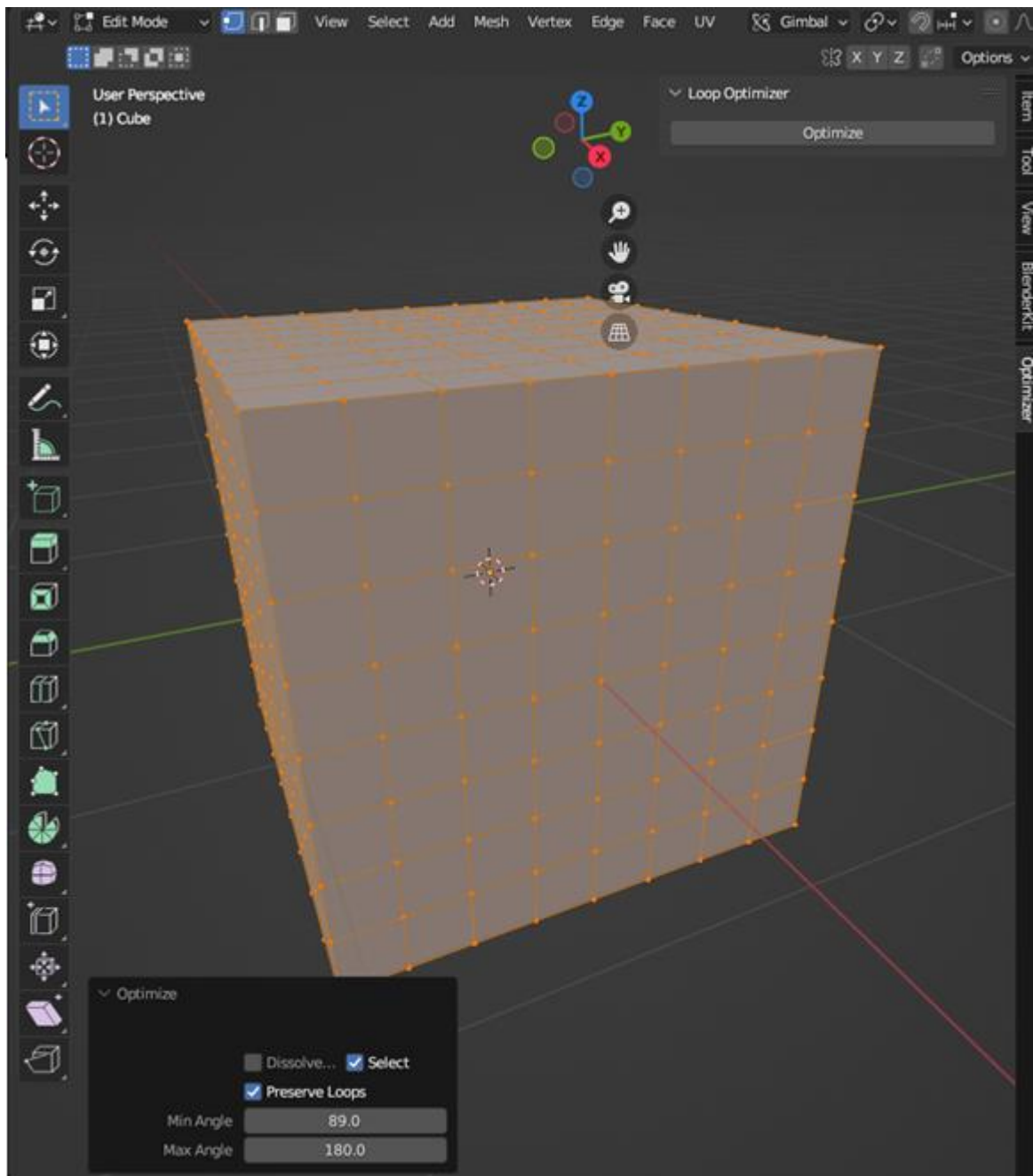


Рисунок 4.7 – Меню виклика аддона

Функція "Dissolve edges" управляє тим, що станеться зі знайденими петлями у вашій 3D моделі. Якщо вона включена, знайдені петлі будуть видалені. Якщо вона вимкнена, знайдені петлі просто виділяться, але залишаться на місці. Це дає можливість або відразу видалити непотрібні петлі, або спочатку перевірити, які петлі будуть видалені, і прийняти рішення пізніше. Застосування функції "Dissolve edges" можна побачити на рисунку 4.8.

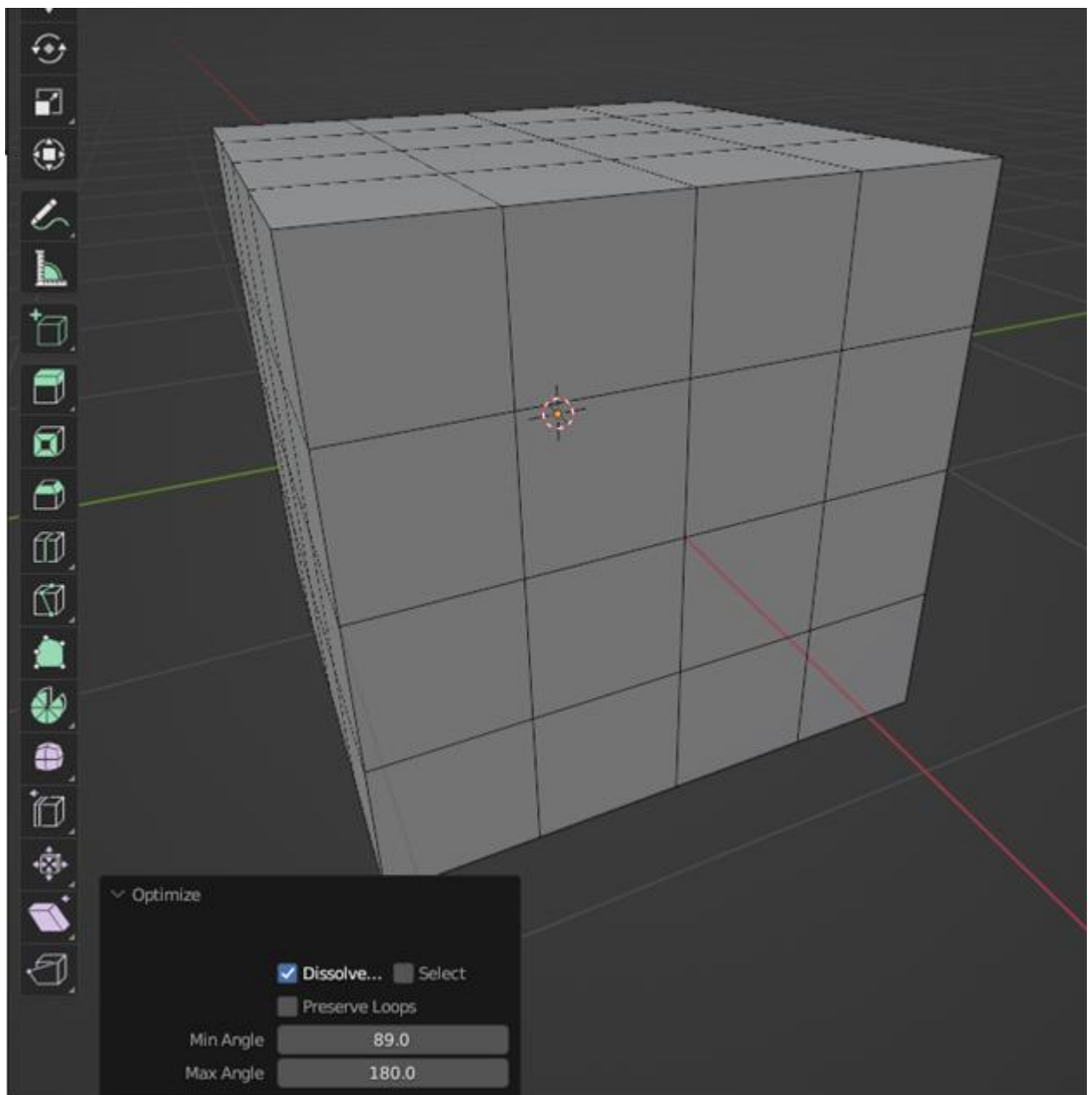


Рисунок 4.8 – Застосування функції "Dissolve edges"

Функція "Select" керує тим, чи будуть виділені петлі, які залишилися після виконання скрипта. Застосування функції "Select" на рисунку 4.9. Якщо вона включена то всі петлі, які були видалені, будуть виділені. Якщо її вимкнено, то петлі залишаться невиділеними.

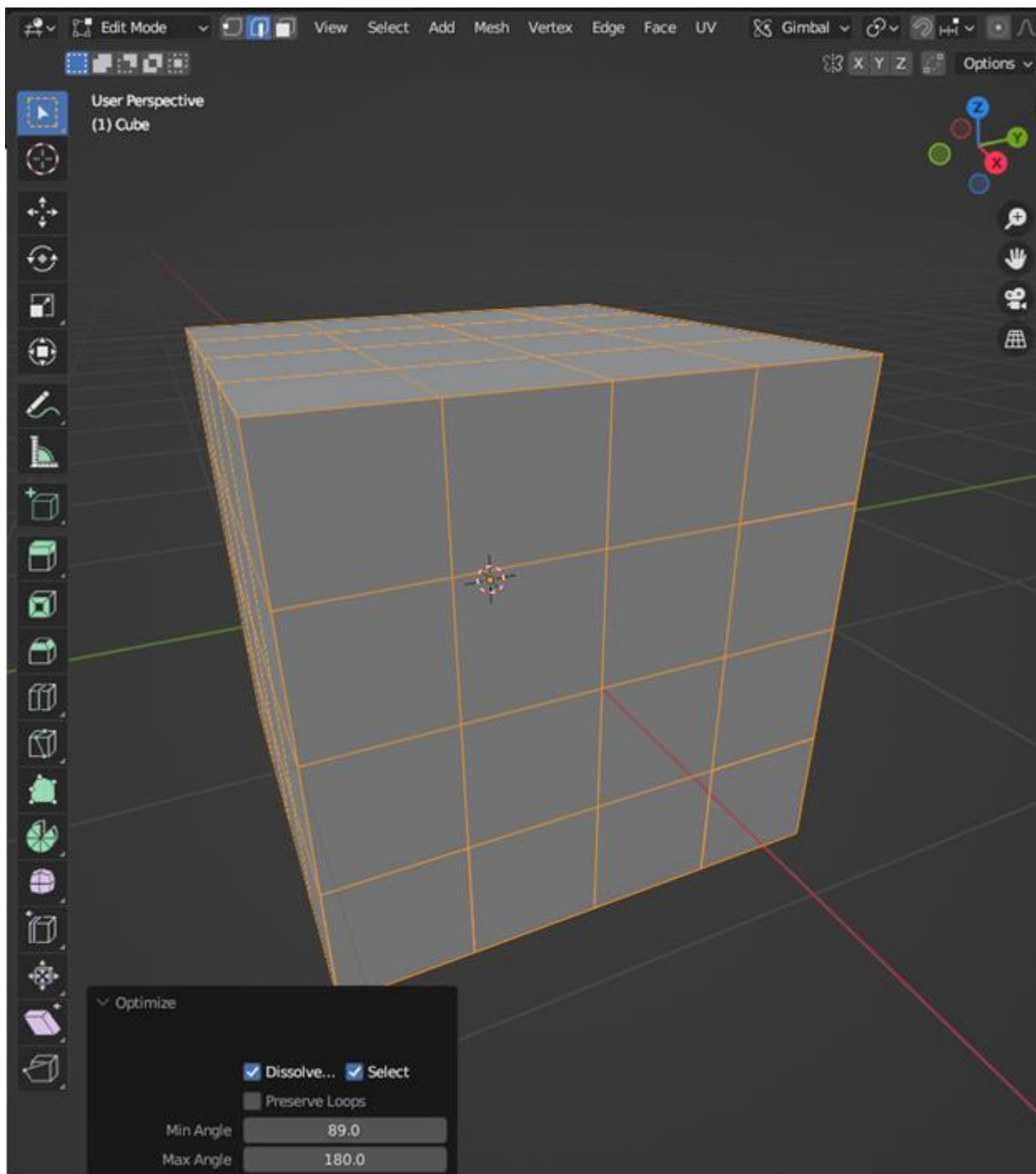


Рисунок 4.9 – Застосування функції "Select"

Параметр "Preserve Loops" визначає, чи мають бути видалені лише ті петлі, які мають менше двох сусідніх петель. Наглядно як працює функція "Preserve Loops" зображено на рисунку 4.10. Якщо він увімкнений, петлі, що не впливають на сусідні петлі (тобто ті, у яких менше двох сусідніх петель), будуть виключені з видалення. Це дозволяє зберегти структурні елементи моделі, які можуть бути важливими для її цілісності.

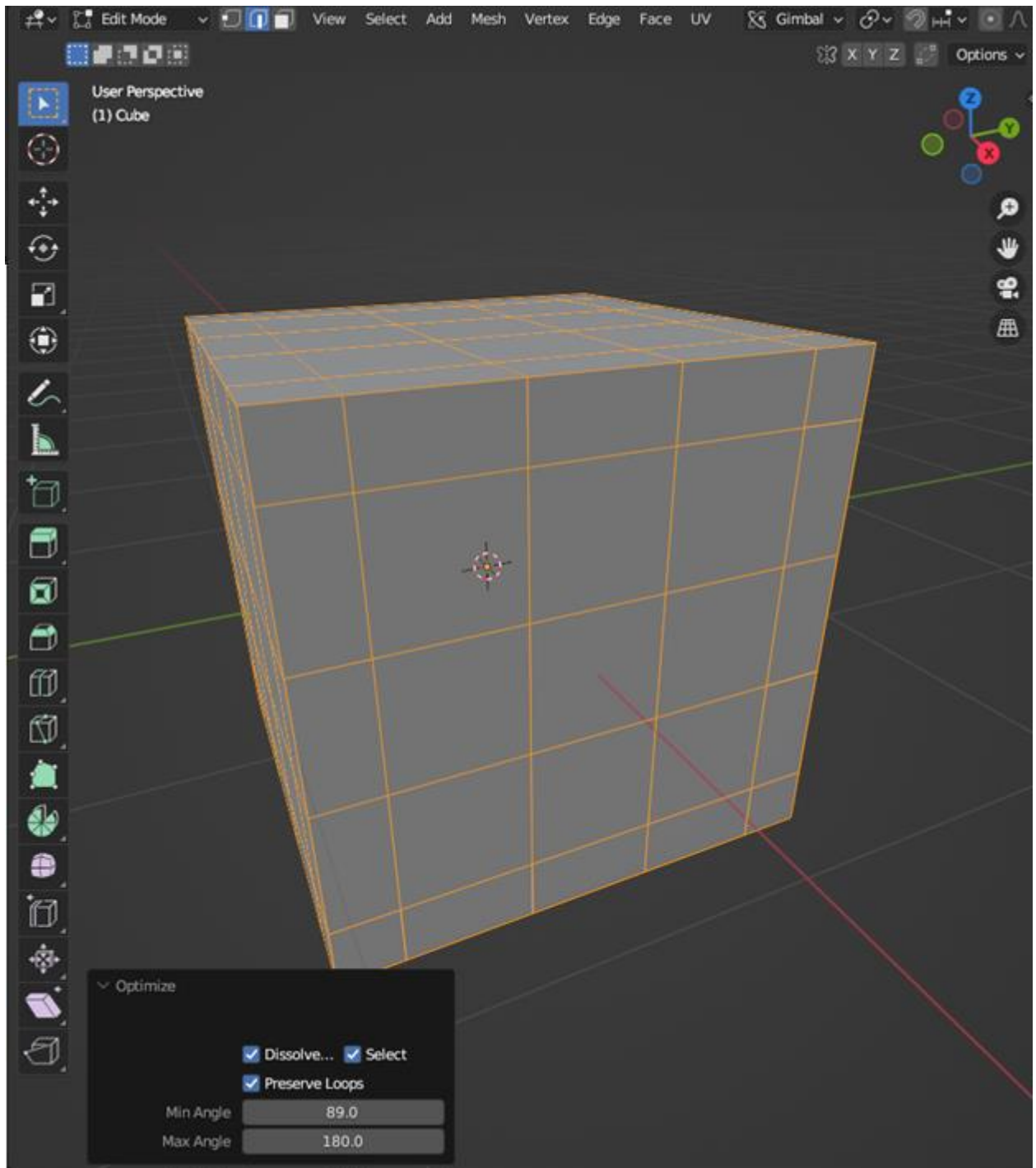


Рисунок 4.10 – Застосування функції "Preserve Loops"

Функції "Min Angle" та "Max Angle" керують кутовими значеннями, які визначають, які петлі будуть оброблені:

- "Min Angle" це мінімальний кутовий поріг, петлі, що утворюють кути, менші від цього значення, будуть обрані для обробки. Значення задається у градусах і може змінюватись від -0.0001 до 180 градусів;
- "Max Angle" це максимальний кутовий поріг, петлі, що утворюють кути, що перевищують це значення, також будуть вибрані для обробки. Значення задається у градусах і може змінюватись від 0.0001 до 180 градусів.

Ці два параметри задають діапазон кутів, в який повинні попадати петлі, щоб бути обробленими (виділеними або віддаленими):

- якщо кут петлі більший за "Min Angle" і менший за "Max Angle", то петля буде оброблена;
- кути, що знаходяться поза цим діапазоном, будуть проігноровані.

Таким чином, ви можете точно налаштувати, які петлі будуть обрані для обробки, ґрунтуючись на їхніх кутах. Тож ці функції дозволяють контролювати процес роботи з петлями, керуючи їх видаленням та виділенням на вашу думку.

Функції "Min Angle" та "Max Angle" зображено на рисунку 4.11.

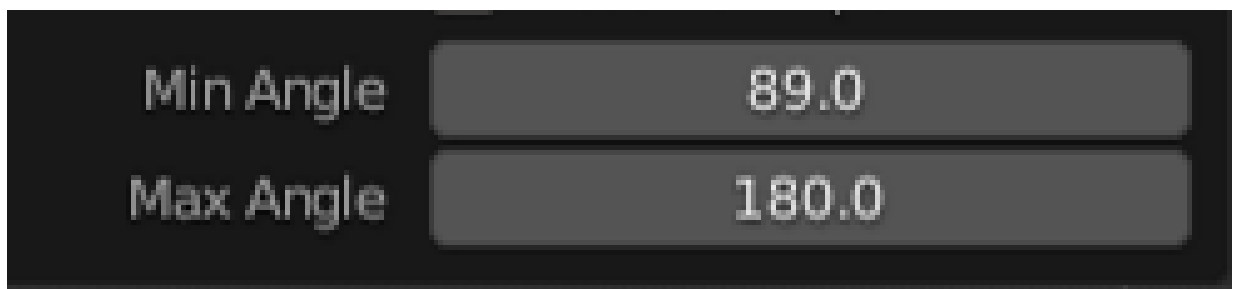


Рисунок 4.11 – Застосування функції "Min Angle" та "Max Angle"

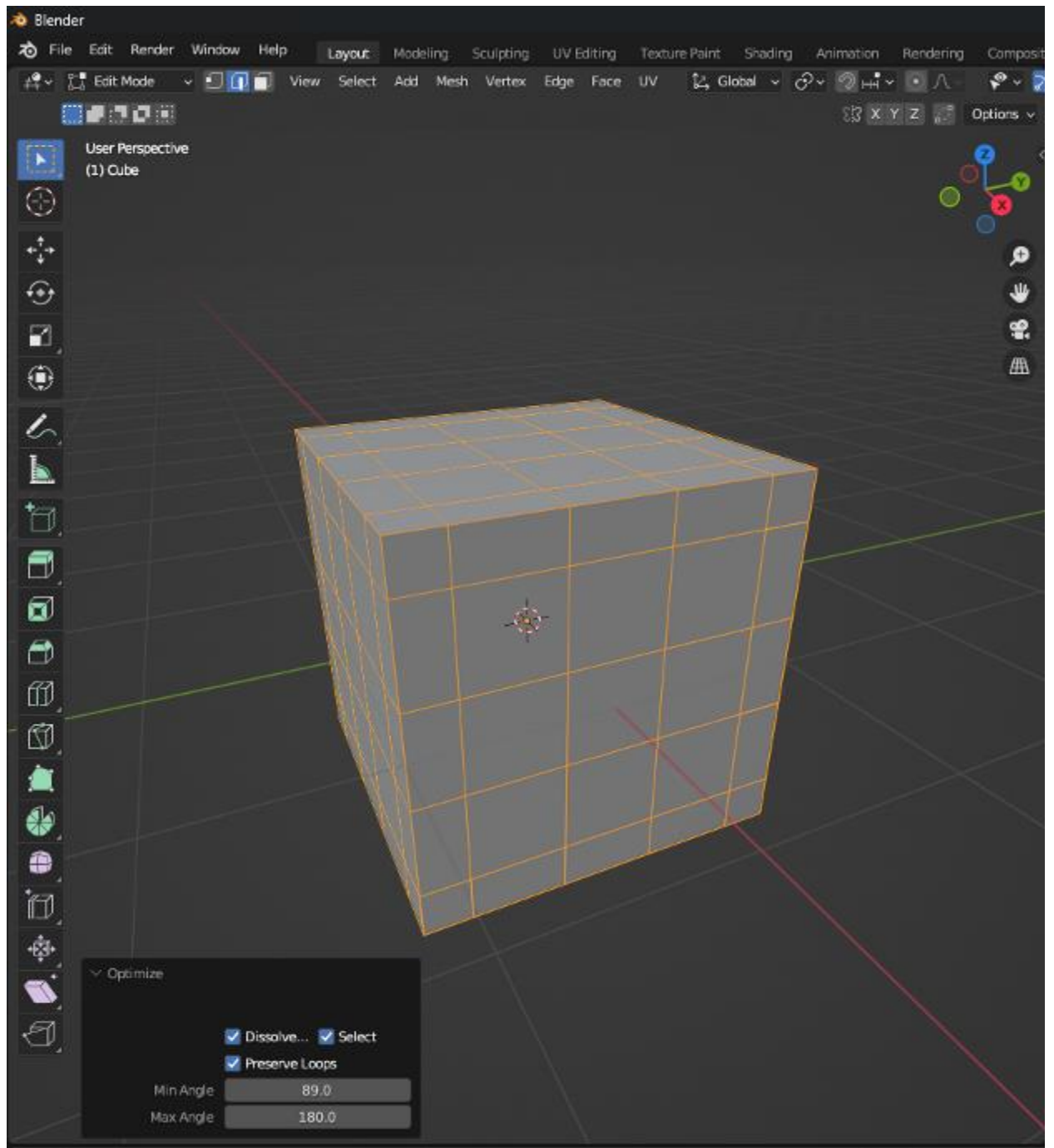


Рисунок 4.12 – Демонстрація застосування всіх функцій одразу

Дотримуючись цих інструкцій, можливо ефективно використовувати аддон для оптимізації полігональної сітки у Blender, видаляючи зайві ребра та підтримуючи чистоту геометрії моделей.

4.2 Експериментальне дослідження ефективності розробленого аддона

Аддон для оптимізації полігональної сітки у Blender був розроблений для покращення топології 3D моделей шляхом спрощення та очищення сіток. Мета цього експериментального дослідження полягає в оцінці ефективності цього аддону шляхом порівняння топології моделей до та після його застосування. У дослідженні розглянемо:

- низькополігональну модель;
- модель середньої складності;
- високополігональну модель.

Об'єкти для експериментального дослідження зображено на рисунку 4.13.

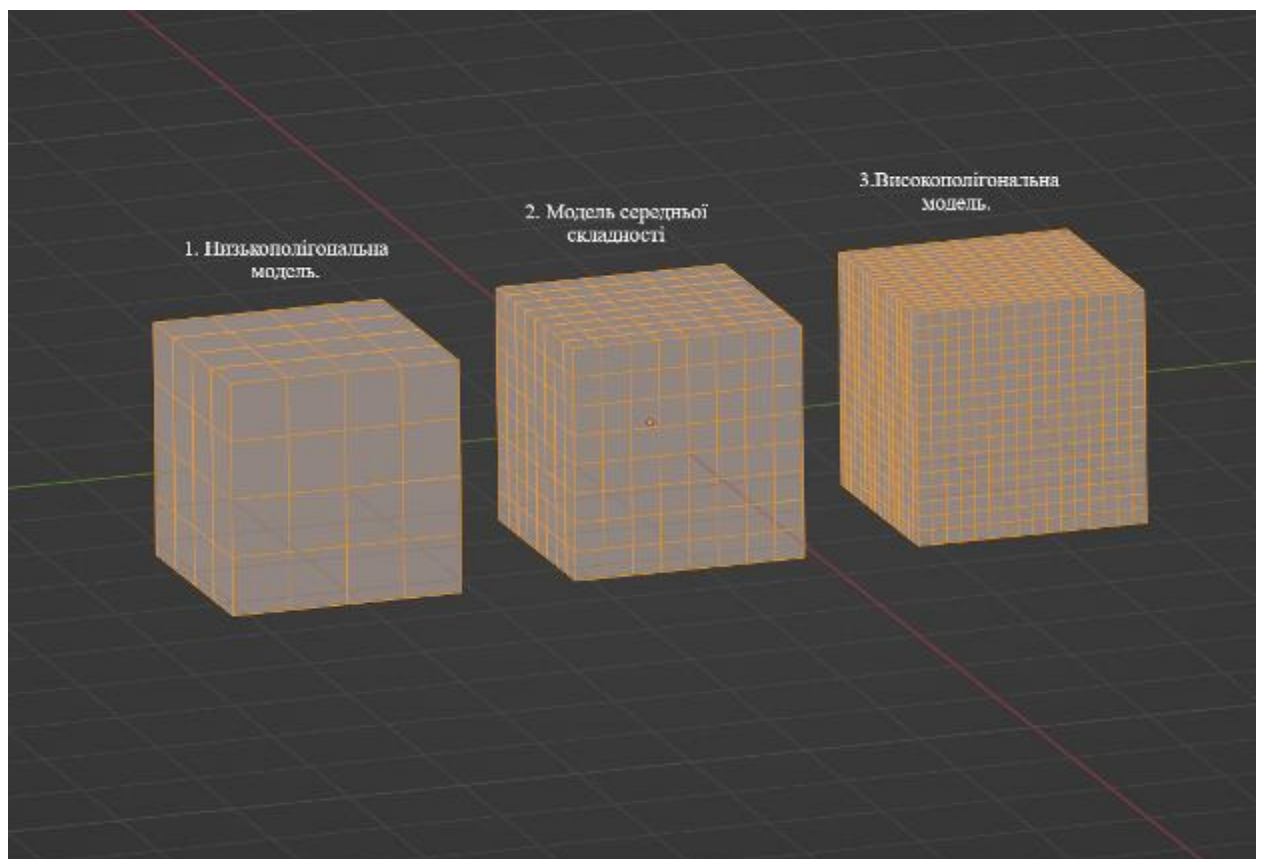


Рисунок 4.13 – Скріншот моделей до використання аддону

Для кожної моделі буде проведено аналіз вихідної топології, що включає, кількість вершин, ребер, граней та візуальні зміни топології.

У таблиці 4.1 наведено кількісні показники трьох моделей до оптимізації з використанням розробленого аддону. Показано такі дані:

- об'єкт (низькополігональна, середньої складності, високополігональна);
- вершини (кількість вершин моделі до оптимізації);
- ребра (кількість ребер у моделі до оптимізації);
- грані (кількість граней у моделі до оптимізації).

Таблиця 4.1 – Порівняння кількості вершин та ребер до оптимізації

Об'єкт	Вершини	Ребра	Грані
Низькополігональна модель	98	192	96
Модель середньої складності	386	768	384
Високополігональна модель	1538	3072	1536

Для кожної моделі застосуємо аддон із використанням стандартних налаштувань Dissolve edges та Select (рис. 4.14). У процесі оптимізації фіксується кількість вершин, кількість ребер, візуальні зміни топології.



Рисунок 4.14 – Застосування функції "Dissolve Edges" та "Select"

Було проведено візуальні порівняння моделей до та після оптимізації. Скріншоти моделей в режимі до (рис. 4.13) та після (рис. 4.15) використання аддону показують, що кількість трикутників та інших топологічних артефактів суттєво зменшилась.

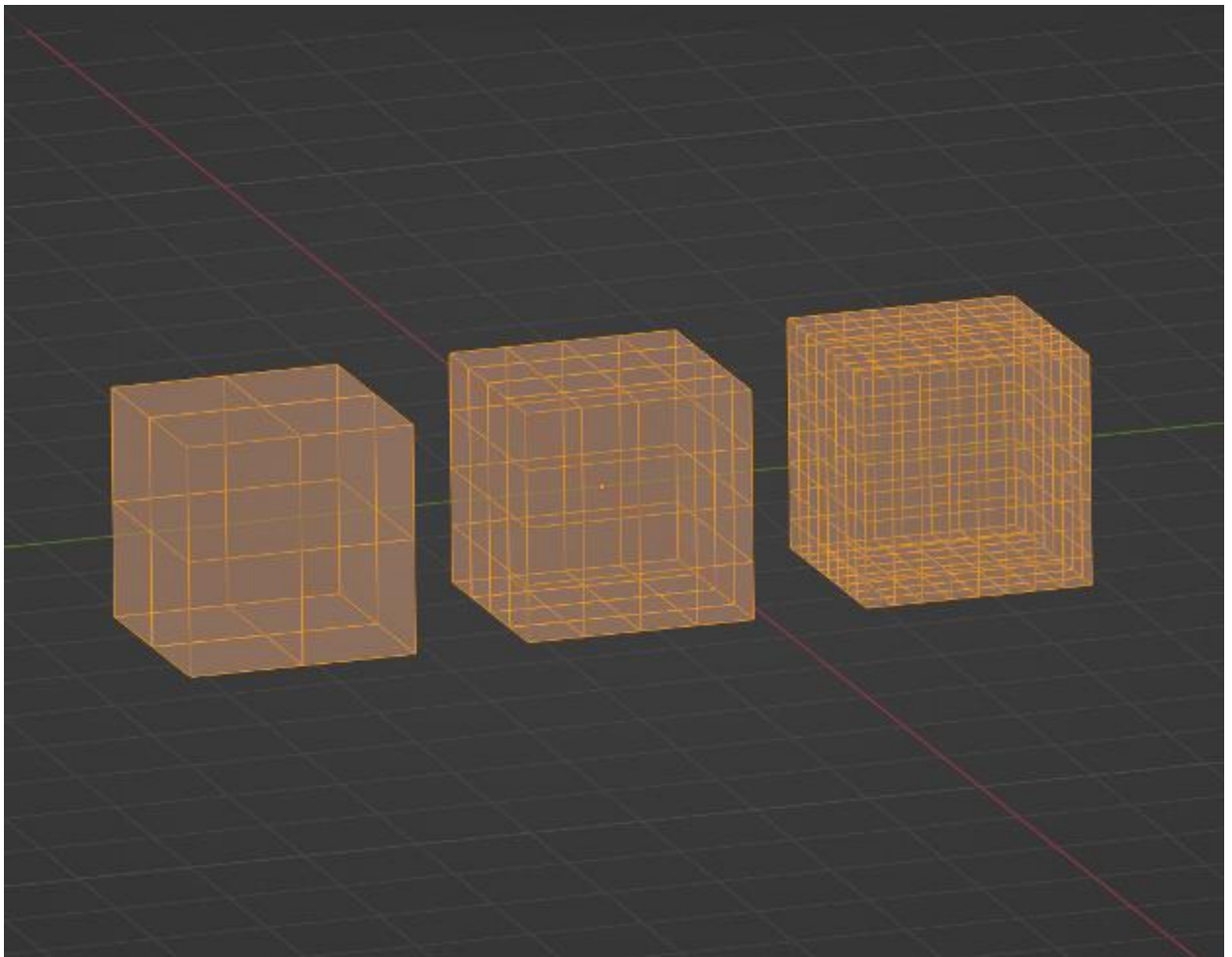


Рисунок 4.15 – Скріншот моделей після використання аддону

Результати кількісного аналізу представлені в таблиці, де вказано кількість вершин і ребер після оптимізації і після оптимізації, а також час, витрачений на оптимізацію для кожної моделі.

У таблиці 4.2 наведено кількісні показники трьох моделей після оптимізації з використанням розробленого аддону.

Кількісні показники показують значне скорочення вершин і ребер після оптимізації. Якісні покращення включають зменшення кількості трикутників, зірочок та інших артефактів, що покращує загальну топологію моделі.

Таблиця 4.2 – Порівняння кількості вершин та ребер після оптимізації

Об'єкт	Вершини	Ребра	Грані
1	2	3	4
Низькополігональна модель	26	48	24
Модель середньої складності	98	192	96
Високополігональна модель	386	768	384

Дослідження показало, що аддон для оптимізації петель у Blender успішно виконує свою функцію, покращуючи топологію моделей та видаляючи зайві елементи. Подальший розвиток та налаштування цього аддону можуть зробити його ще більш ефективним та універсальним.

Тепер скористаємось опцією аддону "Preserve Loops" разом з "Dissolve Edges", "Select". Застосування всіх функцій можна побачити на рисунку 4.16.

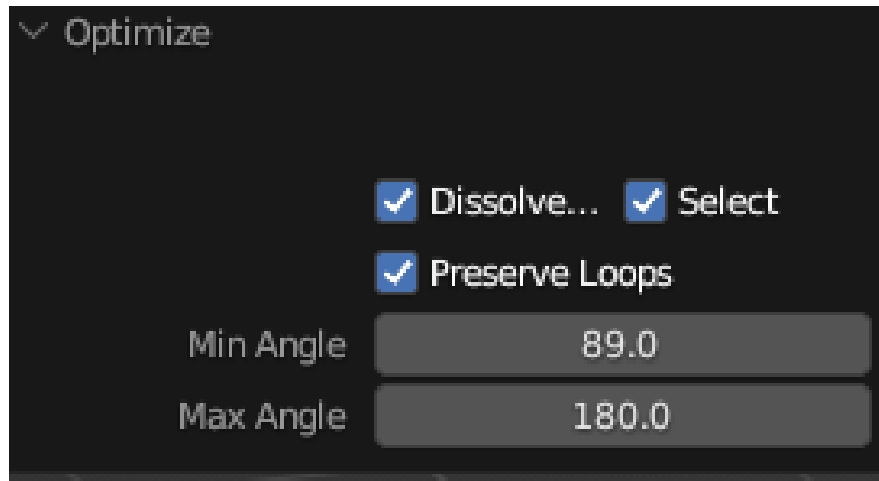


Рисунок 4.16 – Застосування "Dissolve Edges", "Select" та "Preserve Loops"

Якщо використовувати опцію Preserve Loops, то оптимізація буде проводитися з урахуванням збереження петель у топології моделі. Наглядно можна побачити на рисунку 4.17.

Це означає, що оптимізація ігноруватиме відкриті петлі, тобто ті петлі, які з'єднані з менш ніж двома іншими петлями. У процесі оптимізації фіксується кількість вершин, кількість ребер, візуальні зміни топології.

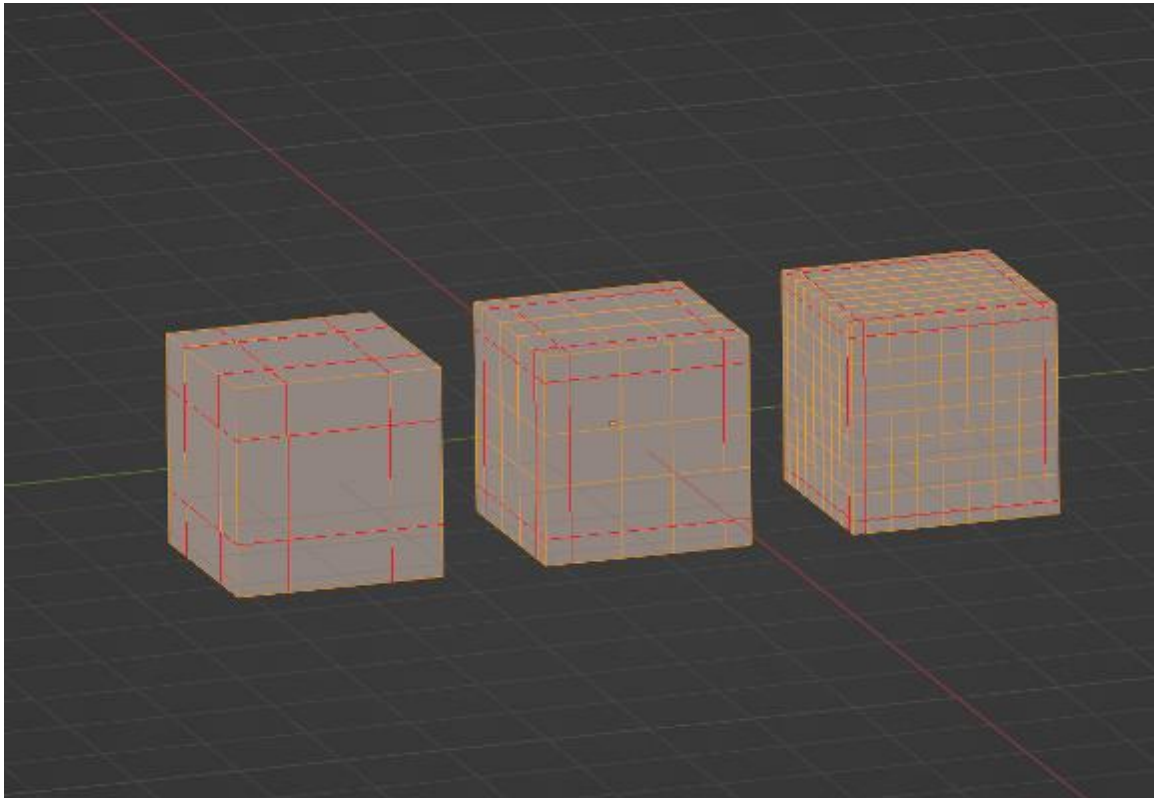


Рисунок 4.17 – Застосування "Dissolve Edges", "Select" та "Preserve Loops"

В результаті використання опції "Preserve Loops":

- петлі в моделі, які є частиною більшої структури або топологічно важливі, не будуть змінені або видалені;
- відкриті петлі, які є частиною будь-якої більшої структури, будуть проігноровані у процесі оптимізації.

Це може бути корисним у випадках, коли оптимізація не повинна втручатися в основні структури або форми моделі, зберігаючи її важливі деталі та особливості.

У таблиці 4.3 наведено кількісні показники трьох моделей після оптимізації з застосуванням "Dissolve Edges", "Select" та "Preserve Loops".

Таблиця 4.3 – Таблиця опис функцій

Об'єкт	Вершини	Ребра	Грані
Низькополігональна модель;	56	108	54
Модель середньої складності	152	300	150
Високополігональна модель	488	972	486

В результаті проведеного експерименту було встановлено, що аддон ефективно зменшує кількість вершин та ребер у моделях, покращуючи при цьому топологію. Час, витрачений на оптимізацію, варіюється залежно від складності моделі, проте процес займає незначний час.

Проведено порівняння візуальних характеристик моделей до та після оптимізації, а також оцінка наявності чи відсутності топологічних артефактів.

Деякі моделі можуть мати складну топологію, для якої аддон може не спрацювати оптимально. Можливо, буде потрібно подальше налаштування аддону для специфічних типів моделей.

Дослідження показало, що аддон для оптимізації сітки у Blender успішно виконує своє завдання, покращуючи топологію моделей та видаляючи надлишкові елементи. Подальший розвиток та налаштування аддону можуть зробити його ще більш ефективним та універсальним.

4.3 Висновки до розділу 4

Отже, після ретельного аналізу розділу 4, можна зробити висновок, що представлений посібник охоплює процес встановлення та використання аддону в Blender для оптимізації полігонів моделей.

Починається з детальних інструкцій щодо встановлення аддону через стандартне меню налаштувань програми, після чого послідовно переходить до пояснення кроків підготовки моделі.

Наголошується на важливості переходу в режим редагування моделі та надаються докладні вказівки щодо застосування аддону, включаючи рекомендації щодо налаштування параметрів для досягнення найкращих результатів.

Ілюстративні зображення доповнюють текст, забезпечуючи наочне уявлення кожного етапу процесу та уточнюючи функціонал аддону.

Також було проведено експериментальне дослідження керівництво що доведено, що аддон підтверджу його ефективність у оптимізації полігонів моделей у Blender.

ВИСНОВКИ

В результаті виконання роботи було розроблено аддон для Blender, який забезпечує оптимізацію топології 3D моделей. У ході роботи над проєктом було проведено аналіз предметної області, що дозволило визначити основні потреби та завдання користувачів у галузі оптимізації геометрії 3D моделей. А саме було виявлено ключові аспекти, які потребують оптимізації при моделюванні:

- видалення зайвих ребер, які збільшують кількість полігонів та складність моделі, що може негативно позначитися на продуктивності та якості візуалізації;
- оптимізація петель, оскільки неправильна їхня структура може призвести до проблем при текстуруванні та анімації;
- автоматизація процесу ретопології, оскільки ручна ретопологія трудомістка і потребує значних часових витрат.

Аддон інтегрується в Blender і надає користувачеві інтуїтивно зрозумілі інструменти для оптимізації топології моделей. У 3D View доступна панель керування аддоном, де можна налаштувати параметри оптимізації та запустити процес. Користувач може встановити параметри мінімального та максимального кута для видалення петель, а також вибрати режим роботи (видалення або вибір петель). При активації функції аддон автоматично аналізує вибрані ребра та оптимізує топологію, видаляючи надлишкові петлі. Користувач також може зберегти певні петлі, які повинні бути видалені в процесі оптимізації.

Аддон надає потужні та інтуїтивно зрозумілі інструменти, які будуть корисні як користувачам-початківцям, так і досвідченим професіоналам, які прагнуть підвищити ефективність своєї роботи.

Результати роботи доповідалися на щорічній науково-практичній конференції викладачів, науковців, молодих учених, аспірантів та студентів «Тиждень науки-2024».

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Blender [Електрон. ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Blender>.
2. Autodesk Maya [Електрон. ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/Autodesk_Maya.
3. ZBrush [Електрон. ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/ZBrush>.
4. Grease Pencil [Electronic resource]. – Access mode: https://web.archive.org/web/20210613084211/https://docs.blender.org/manual/en/latest/grease_pencil/index.html.
5. Типи об'єктів – Object Types [Електрон. ресурс]. – Режим доступу: https://docs.blender.org/manual/uk/2.82/scene_layout/object/types.html.
6. Bool Tool [Electronic resource]. – Access mode: <https://blender-addons.org/bool-tool-addon/>.
7. Sculpt Tools v2.0.0.6 [Electronic resource]. – Access mode: <https://blenderartists.org/t/sculpt-tools-v2-0-0-6/1247700>.
8. Animation Nodes [Electronic resource]. – Access mode: https://github.com/JacquesLucke/animation_nodes.
9. Rigify [Electronic resource]. – Access mode: <https://docs.blender.org/manual/en/2.81/addons/rigging/rigify.html>.
10. Cycles Material Vault [Electronic resource]. – Access mode: <https://cyclesmaterialvault.gumroad.com/l/Yrhkx>.
11. HDR Light Studio [Електрон. ресурс]. – Режим доступу: <https://www.lightmap.co.uk/>.
12. Substance Painter Live Link [Electronic resource]. – Access mode: <https://xolotlstudio.gumroad.com/l/fTRFN>.
13. UV Packmaster [Electronic resource]. – Access mode: <https://uvpackmaster.com/>.

14. Blender Asset Management [Electronic resource]. – Access mode: <https://pitiwazou.artstation.com/pages/asset-management>.
15. Decimate [Електрон. ресурс]. – Режим доступа: <https://docs.blender.org/manual/ru/dev/modeling/modifiers/generate/decimate.html>
16. Remesh [Електрон. ресурс]. – Режим доступа: <https://docs.blender.org/manual/ru/2.93/modeling/modifiers/generate/remesh.html>.
17. Addons [Електрон. ресурс]. – Режим доступа: <https://docs.blender.org/manual/ru/3.6/addons/index.html>.
18. Rendering [Electronic resource]. – Access mode: <https://web.archive.org/web/20210613084359/https://docs.blender.org/manual/en/latest/render/index.html>.
19. Interface [Електрон. ресурс]. – Режим доступа: <https://docs.blender.org/manual/uk/dev/editors/preferences/interface.html>.
20. Windows 11 [Електрон. ресурс]. – Режим доступа: https://uk.wikipedia.org/wiki/Windows_11.
21. Python [Електрон. ресурс]. – Режим доступа: <https://uk.wikipedia.org/wiki/Python>.
22. C [Електрон. ресурс]. – Режим доступа: [https://uk.wikipedia.org/wiki/C_\(%D0%BC%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F\)](https://uk.wikipedia.org/wiki/C_(%D0%BC%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)).
23. C++ [Electronic resource]. – Access mode: <http://surl.li/ulbbn>.
24. Середовище розробки для Python: що це, які вони бувають і як їх використовувати [Електрон. ресурс]. – Режим доступа: <https://foxminded.ua/seredovyshche-rozrobky-python/>.

ДОДАТОК А
Текст програми

```

import bpy
import bmesh
from bpy.props import BoolProperty, FloatProperty
from math import radians

class OPTIMIZE_MESH_OT_OptimizeMesh(bpy.types.Operator):
    bl_idname = "object.optimize_mesh"
    bl_label = "Optimize Mesh"
    bl_options = {'REGISTER', 'UNDO'}

    dissolve_edges: BoolProperty(
        name="Dissolve Edges",
        description="If enabled, edges will be dissolved. Otherwise, loops will
only be selected.",
        default=True,
    )

    select_after_dissolve: BoolProperty(
        name="Select After Dissolve",
        description="If enabled, the selection will be applied after dissolving
edges.",
        default=False,
    )

    save_form: BoolProperty(
        name="Save Form",
        description="Ignore open loops, those which are connected to less
than 2 other loops.",
        default=False,
    )

```

```

min_angle: FloatProperty(
    name="Min Angle",
    description="Minimum angle threshold for edge removal.",
    default=0.0001,
    min=0,
    max=180,
    step=1,
)

```

```

max_angle: FloatProperty(
    name="Max Angle",
    description="Maximum angle threshold for edge removal.",
    default=180,
    min=0.0001,
    max=180,
    step=1,
)

```

```

def execute(self, context):
    optiloops(self, context)
    return {'FINISHED'}

```

```

def get_loop(self, bm, edge0, sel):
    verts2check = edge0.verts[:]
    checkedverts = []
    loop_edges = [edge0]
    sel.discard(edge0)

    while verts2check:
        v = verts2check.pop()

```

```

checkedverts.append(v)
if len(v.link_edges) == 4:
    estart = None
    for e in v.link_edges:
        if e in loop_edges:
            estart = e
            break

    for e in v.link_edges:
        if estart and not any((f in estart.link_faces) for f in e.link_faces):
            if len(e.link_faces) < 2 or to_keep(self, e, bm):
                sel.discard(e)
                return []

        if self.save_form and e in loop_edges:
            sel.discard(e)
            return []

    else:
        sel.discard(e)
        loop_edges.append(e)
        for v in e.verts:
            if v not in checkedverts and v not in verts2check:
                verts2check.append(v)

if self.save_form and len(set(loop_edges)) == len(loop_edges):
    return []

return list(set(loop_edges))

```

```

class edgeloop:
    def __init__(self):
        self.edges = []
        self.neighbours = set()

def check_angles(edges, Min_angle, Max_angle):
    for edge in edges:
        a = edge.calc_face_angle()
        if not (Min_angle <= a <= Max_angle):
            return False
    return True

def to_keep(self, edge, bm):
    return False

def deselect(bm):
    for v in bm.verts:
        v.select_set(False)
    bm.select_flush(False)

def get_loops(self, bm, sel, Max_angle, Min_angle):
    loops = []
    while sel:
        eligible = True
        last = sel.pop()
        if len(last.link_faces) < 2 or to_keep(self, last, bm):
            eligible = False
            sel.discard(last)
            _loop = []
        else:

```



```

):
    loop.neighbours.add(loop1)

```

```

def skiploop(final_loops, skip_loops, loop):
    final_loops.append(loop)
    checkneighbours = loop.neighbours
    checked = [loop]
    while checkneighbours:
        neighbour = checkneighbours.pop()
        checked.append(neighbour)
        if neighbour not in checked:
            skip_loops.append(neighbour)

    for n in neighbour.neighbours:
        if n not in checked:
            final_loops.append(n)
            checked.append(n)
            for n1 in n.neighbours:
                if n1 not in checked:
                    checkneighbours.add(n1)
                    checked.append(n1)
                    skip_loops.append(n1)

def optiloops(self, context):
    cao = bpy.context.active_object
    self.hidden = context.object.name
    bm = bmesh.from_edit_mesh(cao.data)
    bm.normal_update()
    bpy.ops.mesh.select_mode(
        use_extend=False, use_expand=False, type='EDGE')

```

```

Max_angle = radians(self.max_angle)
Min_angle = radians(self.min_angle)
sel = {e for e in bm.edges if e.select}
if not sel:
    sel = set(bm.edges[:])

sel_copy = sel.copy()

deselect(bm)

loops = get_loops(self, bm, sel, Max_angle, Min_angle)

get_neighbours(loops)

final_loops = []
skip_loops = []

for loop in loops:
    if loop not in skip_loops:
        skiploop(final_loops, skip_loops, loop)

if self.dissolve_edges:
    for loop in final_loops:
        for e in loop.edges:
            e.select = True
        bpy.ops.mesh.dissolve_edges()

if self.select_after_dissolve:
    for e in sel_copy:
        if e in bm.edges:

```

```
e.select = True
```

```
bmesh.update_edit_mesh(cao.data)
```

```
class OPTIMIZE_MESH_PT_OptimizeMeshPanel(bpy.types.Panel):
```

```
    bl_label = "Mesh Optimization"
```

```
    bl_idname = "PT_OptimizeMeshPanel"
```

```
    bl_space_type = 'VIEW_3D'
```

```
    bl_region_type = 'UI'
```

```
    bl_category = 'Optimize Mesh'
```

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

```
        layout = self.layout
```

```
        scene = context.scene
```

```
        layout.prop(scene, "min_angle")
```

```
        layout.prop(scene, "max_angle")
```

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

```
    def register():
```

```
        bpy.utils.register_class(OPTIMIZE_MESH_OT_OptimizeMesh)
```

```
        bpy.utils.register_class(OPTIMIZE_MESH_PT_OptimizeMeshPanel)
```

```
    def unregister():
```

```
        bpy.utils.unregister_class(OPTIMIZE_MESH_OT_OptimizeMesh)
```

```
        bpy.utils.unregister_class(OPTIMIZE_MESH_PT_OptimizeMeshPanel)
```

```
        del bpy.types.Scene.min_angle
```

```
        del bpy.types.Scene.max_angle
```

```
if __name__ == "__main__":
```

```
    register()
```

ДОДАТОК Б
Слайди презентації



Рисунок Б.1 - Слайд 1



Рисунок Б.2 - Слайд 2

МЕТА ТА ЗАВДАННЯ РОБОТИ

Метою роботи є розробка програмного забезпечення для розширення функціональних можливостей середовища Blender та оптимізації топології 3D моделей.

Завдання роботи:

- провести аналіз існуючих аддонів для оптимізації полігональних сіток у Blender та визначити актуальність створення власного аддону ;
- розробити алгоритм функціонування програмного забезпечення, що враховує особливості роботи аддона в Blender;
- реалізувати аддон, який інтегруватиметься з Blender і надасть можливість оптимізації полігональних сіток моделей;
- провести тестування аддону та оцінити його ефективність.

Рисунок Б.3 - Слайд 3

АНАЛІЗ ВИМОГ

- | | |
|---|--|
| – користувач повинен мати можливість автоматично виділити петлі для подальшого редагування; | інтерфейс має бути інтуїтивно зрозумілим та легким у використанні. |
| – оптимізовувати виділені петлі в меші | – аддон повинен бути сумісним із останніми версіями Blender, |
| – аддон повинен ігнорувати петлі, які з'єднані менш ніж з двома іншими петлями; | – аддон повинен запобігати помилкам та некоректній поведінці, |
| – повинна бути панель керування в 3D View, де користувач може налаштувати параметри та запустити оптимізацію; | – код аддону має бути гнучким та легко розширюваним |
| – редагування повинно функціонувати в режимі редагування меша (Edit Mode). | |

Рисунок Б.4 - Слайд 4

АРХІТЕКТУРА СИСТЕМИ

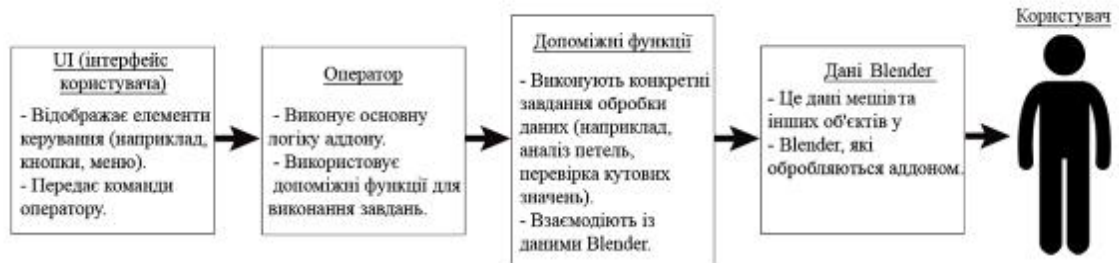


Рисунок Б.5 - Слайд 5

ВИБІР ІНСТРУМЕНТАРІЮ РОЗРОБКИ

Критерій	Python	C++	C#
Синтаксис	Простий та легкий для читання, використовує відступи для блоків коду	Короткий, немає управління потоком, використовує фігурні дужки	Складний, підтримує процедурне та об'єктно-орієнтоване програмування, використовує фігурні дужки
Продуктивність	Інтерпретований, повільніше компілюється в машинний код	Висока, компілюється в машинний код	Висока, компілюється в машинний код
Області застосування	Веб-розробка, науки обчислення, машинне навчання, автоматизація, розробка ігор, системи, що будуються	Системне програмування, будова систем, розробка мікроконтролерів	Системне програмування, розробка ігор, фінансове програмування, будова систем, обробка графіки
Інструментальні засоби	PyCharm, VS Code, Jupyter Notebook	GCC, Clang, Visual Studio, Code::Blocks	Visual Studio, CLion, GCC
Затребуваність на ринку праці	Висока в області аналізу даних, машинного навчання та веб-розробки	Висока в області аналізу даних, машинного навчання та веб-розробки	Висока у розробці ігор, системному програмуванні, будованих системах та фінансовому секторі

Рисунок Б.6 - Слайд 6



Рисунок Б.7 - Слайд 7



Рисунок Б.8 - Слайд 8



Рисунок Б.9 - Слайд 9

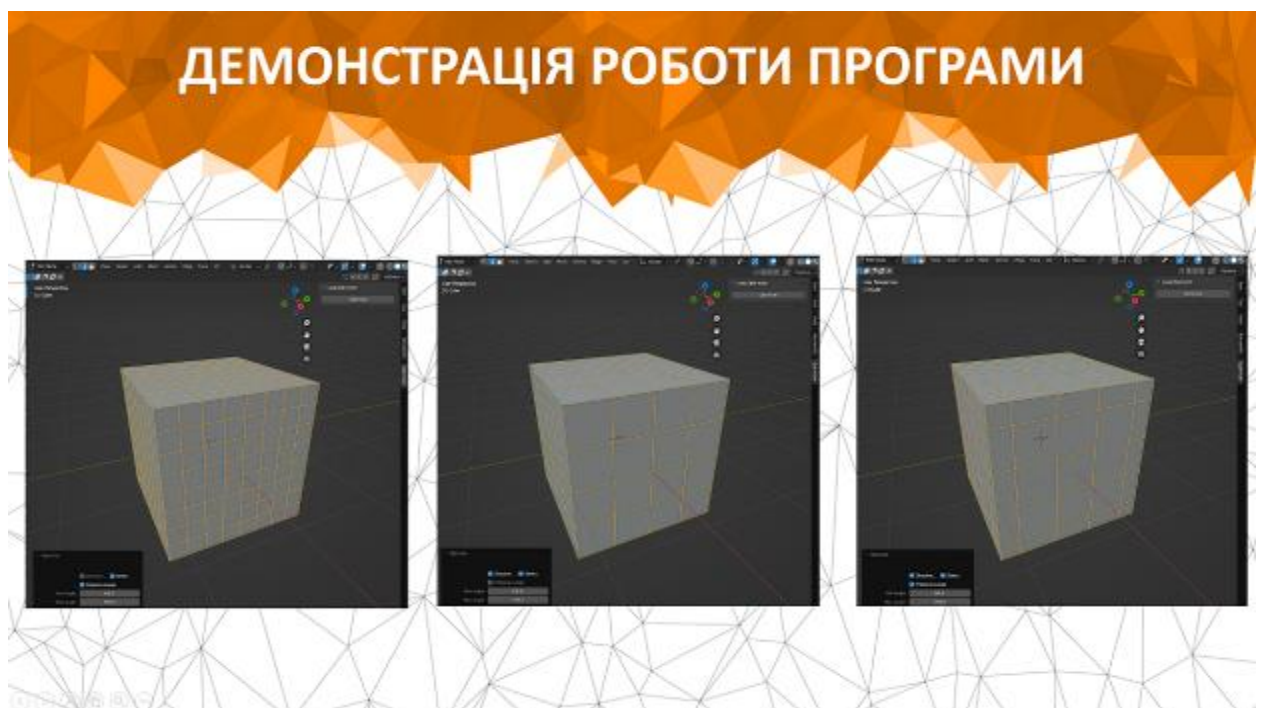


Рисунок Б.10 - Слайд 10



Рисунок Б.11 - Слайд 11

ВИСНОВКИ

В результаті виконання роботи було розроблено аддон для Blender, який забезпечує оптимізацію топології 3D моделей.

Аддон інтегрується в Blender і надає користувачеві інтуїтивно зрозумілі інструменти для оптимізації топології моделей. У 3D View доступна панель керування аддоном, де можна налаштувати параметри оптимізації та запустити процес. Користувач може встановити параметри мінімального та максимального кута для видалення петель, а також вибрати режим роботи (видалення або вибір петель). При активації функції аддон автоматично аналізує вибрані ребра та оптимізує топологію, видаляючи надлишкові петлі. Користувач також може зберегти певні петлі, які повинні бути видалені в процесі оптимізації.

Результати роботи доповідалися на щорічній науково-практичній конференції викладачів, науковців, молодих учених, аспірантів та студентів «Тиждень науки-2024»

Рисунок Б.12 - Слайд 12