

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Комп'ютерних наук і технологій
(повне найменування факультету)

Комп'ютерні системи та мережі
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалаврський
(ступінь вищої освіти)

на тему: РОЗРОБКА ТА ОПТИМІЗАЦІЯ ШЕЙДЕРІВ ДЛЯ СТВОРЕННЯ
СТИЛІЗОВАНИХ ТА ПЕРЕКОНЛИВИХ СЦЕН

Виконав(ла): студент(ка) 4 курсу,
групи КНТ-511

Спеціальності
123 Комп'ютерна інженерія
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Комп'ютерна інженерія
(назва освітньої програми (спеціалізації))

ГЕТМАН С. В.
(ПРИЗВИЩЕ та ініціали)

Керівник КИРИЧЕК Г. Г.
(ПРИЗВИЩЕ та ініціали)

Рецензент ЄФИМЕНКО М. В.
(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет Комп'ютерних наук і технологій
Кафедра комп'ютерних систем та мереж
Ступінь вищої освіти бакалаврський
Спеціальність 123 Комп'ютерна інженерія
(код і найменування)
Освітня програма (спеціалізація) Комп'ютерна інженерія
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри КУДЕРМЕТОВ Р.К.

«14» квітня 2025 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ГЕТМАНА Сергія Валерійовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка та оптимізація шейдерів для створення стилізованих та переконливих сцен

керівник проєкту (роботи) к.т.н., доцент, КИРИЧЕК Г. Г.,

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

затверджені наказом закладу вищої освіти від «08» квітня 2025 року № 151

2. Строк подання студентом проєкту (роботи) 01.06.2025 р.

3. Вихідні дані до проєкту (роботи) опис предметної області, інформація щодо PBR, Noise Textures, техніки оптимізації 3-D сцен, знання вузлів в Geometry nodes.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1) Аналіз предметної області;

2) Розробка базових 3-D сцен

3) Створення шейдерів

4) Налаштування сцен і оптимізація

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

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4	КИРИЧЕК Г. Г.		
Нормоконтроль	ПОЛЬСЬКА О.В.		

7. Дата видачі завдання «14» квітня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Аналіз предметної області	до 18.04.2025	
2	Визначення та аналіз вимог до програмного забезпечення	до 22.04.2025	
3	Розробка базових 3-D сцен	до 24.04.2025	
4	Розробка indoor-lit сцени	до 28.04.2025	
5	Розробка outdoor-lit сцени	до 01.05.2025	
6	Налаштування геометрії в сцені.	до 05.05.2025	
7	Розробка систем Geometry nodes	до 12.05.2025	
8	Розробка шейдерів	до 22.05.2025	
9	Оптимізація та заміри продуктивності	до 25.05.2025	
10	Проходження нормоконтролю	до 01.06.2025	
11	Перевірка на наявність академічного плагіату	до 03.06.2025	
12	Проходження рецензування	до 10.06.2025	

Студент(ка)

(підпис)

Сергій ГЕТМАН

(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис)

Галина КИРИЧЕК

(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
74 с., 3табл., 53 рис., 20 джерел.

СТИЛІЗОВАНЕ ОСВІТЛЕННЯ, ШЕЙДЕРИ, РЕНДЕРИНГ У РЕАЛЬНОМУ ЧАСІ, СИСТЕМА НОДІВ, GEOMETRY NODES, SHADER EDITOR, BLENDER, VULKAN.

Об'єкт розробки – шейдери та тривимірні сцени, створені у програмному середовищі Blender.

Мета роботи – розробка та оптимізація шейдерів для створення стилізованих та переконливих сцен у Blender, з метою покращення візуального сприйняття без значних втрат продуктивності.

У ході роботи було проаналізовано технічне завдання та визначено вимоги до художнього стилю й технічної реалізації сцен. Проведено проєктування візуальних ефектів, що включало створення прототипів сцен, підбір освітлення, матеріалів і розробку власних шейдерів за допомогою вузлової системи (Shader Editor) у Blender. Особливу увагу приділено нелінійним, стилізованим матеріалам, які імітують художні техніки, такі як cel-shading, ефект контуру, текстурне зафарбовування тощо.

Розробка шейдерів здійснювалася із врахуванням оптимізації – зменшено кількість обчислень, використано процедурні текстури, мінімізовано навантаження на GPU. Було проведено тестування сцен у режимах Eevee та Cycles для оцінки якості рендерингу та продуктивності на різних системах.

У результаті реалізовано набір оптимізованих шейдерів та створено декілька демонстраційних сцен, які поєднують стилізовану графіку з технічно ефективними рішеннями.

ABSTRACT

Explanatory note for the bachelor's qualification thesis: 74 pages, 3 tables, 53 figures, 20 references.

STYLIZED LIGHTING, SHADERS, REAL-TIME RENDERING, NODE SYSTEM, GEOMETRY NODES, SHADER EDITOR, BLENDER, VULKAN

Object of development – shaders and 3D scenes created within the Blender software environment.

Objective of the thesis – the development and optimization of shaders aimed at creating stylized and visually engaging scenes in Blender, while maintaining high performance and minimizing computational overhead.

During the project, the technical requirements were analyzed, and both the artistic style and technical constraints of the scenes were defined. The development process included the design of visual effects, creation of scene prototypes, lighting setup, material selection, and the implementation of custom shaders using Blender's node-based system (Shader Editor). Special attention was given to non-photorealistic, stylized materials that emulate artistic techniques such as cel-shading, contour lines, and painterly textures.

The shader development process focused on optimization — reducing the number of calculations, leveraging procedural textures, and minimizing GPU load. The created scenes were tested in both Eevee and Cycles rendering engines to evaluate visual quality and performance across different hardware configurations.

As a result, a set of optimized shaders was developed and several demo scenes were created, combining stylized visuals with technically efficient solutions.

ЗМІСТ

Вступ.....	6
1 Аналіз предметної області.....	7
1.2 Особливості рушію.....	9
1.3 Визначення та аналіз вимог до програмного забезпечення.....	11
1.4 Постановка завдань.....	12
2 Розробка базових 3-d сцен.....	14
2.2 Розробка indoor-lit сцени.....	17
2.3 Створення outdoor-lit сцени.....	23
3 Створення шейдерів.....	35
3.2 Створення toon шейдеру.....	35
3.3 Створення outline шейдеру.....	39
3.4 Створення dither шейдера.....	41
3.5 Розробка watercolor шейдеру.....	44
4 Налаштування сцен та оптимізація.....	46
4.1 Створення матеріалів.....	46
4.2 Імплементація шейдерів в indoor-lit сцену.....	56
4.3 Стилізація outdoor-lit сцени.....	61
4.4 Оптимізація та заміри продуктивності.....	67
Висновки.....	71
Перелік джерел посилання.....	72

ВСТУП

У сучасній комп'ютерній графіці значну роль відіграє не лише технічна досконалість зображення, а й його естетична виразність. З розвитком індустрії відеоігор, інтерактивних медіа та анімації зростає попит на нестандартні, художньо стилізовані сцени, що здатні викликати емоційний відгук у користувача та підкреслити унікальність візуального стилю проєкту. Одним із ключових інструментів, що дозволяє досягти такого ефекту, є шейдери — спеціальні програми, які керують візуалізацією об'єктів на рівні графічного процесора.

Використання шейдерів дозволяє реалізовувати різноманітні візуальні ефекти: від простого освітлення до складних процедурних матеріалів та постобробки. Однак створення шейдерів, які є одночасно візуально переконливими та продуктивними, вимагає глибокого розуміння як технічних аспектів рендерингу, так і художніх принципів стилізації. Особливо це актуально для сцен, що відходять від фотореалізму і прагнуть передати певну атмосферу чи художній настрій.

Метою даної дипломної роботи є розробка та оптимізація шейдерів для створення стилізованих і переконливих сцен у середовищі Blender. У ході дослідження були створені авторські матеріали, проведено оптимізацію обчислень для забезпечення продуктивності та проаналізовано способи передачі емоційного забарвлення через візуальні засоби.

Актуальність обраної теми зумовлена зростаючим інтересом до художньо виразної графіки та потребою в інструментах, що поєднують естетику з ефективністю. Результати цієї роботи можуть бути використані в ігровій розробці, цифровій анімації, візуалізації та інших сферах, де важлива емоційна складова візуального середовища.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Візуальна складова є ключовим елементом у комп'ютерній графіці, відеоіграх, віртуальній реальності, анімації та цифровому мистецтві. Значна частина естетичного враження, яке отримує користувач, формується завдяки освітленню, кольоровим переходам, матеріалам та спецефектам. Основним інструментом для керування цими аспектами виступають шейдери – невеликі програми, що виконуються на графічному процесорі (GPU) та відповідають за обчислення кольору пікселів, положення вершин, обробку текстур та інші графічні ефекти.

У традиційному підході велика увага приділяється фотореалізму, що потребує високих обчислювальних ресурсів та складних моделей освітлення. Однак зростає інтерес до стилізованої графіки, яка не імітує реальність, а створює унікальний художній стиль — наприклад, мальований ефект, комікс-стиль, low-poly естетику або експресивне освітлення. Такий підхід дозволяє розробникам візуально вирізняти свій проєкт, зменшувати апаратні вимоги та передавати специфічну атмосферу сцени.

У середовищі Blender, яке є потужним інструментом для 3D-моделювання, анімації та рендерингу, реалізовано широкі можливості для створення та налаштування шейдерів за допомогою візуального редактора вузлів (Shader Editor). Blender підтримує два основні рушії рендерингу – Cycles (фотореалістичний трасувальник променів) та Eevee (реальний час), що дозволяє створювати як високоякісні, так і швидкодіючі візуалізації. Стилізовані сцени зазвичай реалізуються в Eevee, оскільки він підтримує більшість необхідних ефектів і працює у режимі реального часу.

В межах даної предметної області особливо актуальними є такі аспекти:

- побудова нестандартних матеріалів – створення процедурних або комбінованих шейдерів, які імітують художні стилі;
- оптимізація продуктивності – зменшення навантаження на GPU, зниження

кількості обчислень у фрагментних та вершинних шейдерах;

-передача настрою та емоцій — за допомогою кольорової палітри, освітлення, контрастів та спотворень;

-універсальність рішень — придатність створених шейдерів для інтеграції в різні типи проєктів: від ігор до рекламних відео чи архітектурних візуалізацій;

Таким чином, предметна область об'єднує знання з комп'ютерної графіки, програмування, дизайну та візуального мистецтва, а робота зосереджена на розробці інструментів, що дозволяють ефективно поєднувати художню стилізацію з технологічною ефективністю.

У сфері створення 3D-графіки існує чимало професійних програмних рішень, кожне з яких має свої сильні сторони та сфери застосування. Серед основних альтернатив Blender можна виокремити такі середовища:

- Autodesk Maya;
- Autodesk 3ds Max;
- Cinema 4D;
- Houdini;
- ZBrush;
- Unity та Unreal Engine.

Попри наявність таких потужних інструментів, у межах цього проєкту було свідомо обрано Blender, і на це є кілька ключових причин:

По-перше, Blender — безкоштовне та відкрите ПЗ, що робить його ідеальним для індивідуальних розробників, інді-студій та навчальних проєктів. Це значно знижує бар'єр входу в галузь комп'ютерної графіки.

По-друге, його функціональність охоплює весь пайплайн 3D-виробництва — від моделювання, анімації та рендерингу до композитингу та відеомонтажу. Усе це дозволяє працювати в єдиному середовищі без необхідності в додатковому програмному забезпеченні.

Особливо важливо для тематики цієї роботи те, що Blender має гнучкий Shader Editor із підтримкою візуального програмування, що ідеально підходить для створення стилізованих матеріалів. Крім того, рушій Eevee, орієнтований на

рендеринг у реальному часі, забезпечує швидку візуалізацію без значного технічного навантаження — критично важливо для розробки інтерактивних та художньо стилізованих сцен. Нарешті, активна спільнота, постійні оновлення та численні ресурси для навчання роблять Blender не лише інструментом, але й екосистемою, що постійно розвивається.

1.2 Особливості рушію

Як ми вже розібрались, Blender — це безкоштовне, вільне та відкрито розповсюджуване програмне забезпечення, і факт його відкритості є ключовим в його привабливості для людей що тільки починають свій шлях в комп'ютерній графіці. Проєкт підтримується великою спільнотою ентузіастів та професіоналів, а також організацією Blender Foundation. Завдяки відкритому вихідному коду Blender постійно вдосконалюється та інтегрує новітні технології, зберігаючи при цьому повну доступність для всіх користувачів.

Основні переваги Blender:

- безкоштовність і відкритий код;
- універсальність ;
- активна спільнота та підтримка ;
- компактність;
- підтримка сучасних API.

Blender надає потужні візуальні редактори, які дозволяють створювати як прості, так і складні матеріали та візуальні ефекти без необхідності писати код вручну. Саме візуальний підхід до створення ефектів та його доступність та зрозумілість нетехнічним спеціалістам роблять системи шейдерів та геометрії в Blender настільки зручними і новаційними.

Shader Editor — це вузловий редактор, призначений для створення матеріалів у Blender. Він працює у контексті фізично коректного рендерингу

(PBR), але також дозволяє реалізувати стилізовані та експериментальні ефекти.

Основними особливостями є:

- побудова матеріалів через систему вузлів (Nodes);
- повна інтеграція з рендерерами
- підтримка нодів для текстур, освітлення, нормалей, емісії, прозорості, ефектів обводки, зміщення та багатьох інших.

Geometry Nodes — це ще один вузловий інструмент, що дозволяє створювати процедурні моделі, ефекти та навіть системи розташування об'єктів чи частинок:

- керування формою, розмірами, кількістю, положенням геометрії на сцені за допомогою параметрів і вузлів;
- можливість впливати на властивості об'єктів включно з матеріалами — динамічно та інтерактивно;
- використовується для процедурної генерації;
- сумісність з Shader Editor.

Цей підхід дуже зручний для повторюваних або складних структур. Наприклад, можна автоматично розмістити десятки дерев на полі, побудувати плитку, генерувати будівлі чи абстрактні форми. Завдяки процедурності можна легко змінювати результат — просто пересунувши слайдер чи змінивши значення параметра.

Geometry Nodes особливо корисні для побудови великих середовищ або об'єктів, які підлягають частій зміні. Крім того, вузлова структура дає змогу створювати динамічні сцени — наприклад, з урахуванням часу, позицій інших об'єктів чи зовнішніх параметрів. Ця система відкриває шлях до процедурного моделювання в реальному часі з можливістю інтеграції із симуляціями, матеріалами та анімацією[1].

Тож Geometry Nodes, дуже часто використовують для швидкого наповнення сцени, створення рослинності, будівель, дерев, та багатьох інших об'єктів які потребують бути унікальними але швидко змінюваними.

Щодо шейдерів, Shader Editor забезпечує вузловий підхід до створення матеріалів та шейдерів. Користувач може візуально налаштовувати зовнішній вигляд об'єктів, експериментуючи з текстурами, кольорами, прозорістю, ефектами світла та поверхні. Вузлова система дозволяє легко поєднувати різні ефекти, не потребуючи програмування, що суттєво спрощує процес стилізації сцени[2].

1.3 Визначення та аналіз вимог до програмного забезпечення

Визначимо вимоги до програмного забезпечення. Для ефективної реалізації поставленої мети — створення стилізованих та переконливих сцен із використанням шейдерів — важливо сформулювати чіткі функціональні та нефункціональні вимоги до програмного забезпечення, зокрема до продуктивності, інтерактивності та якості графічного відображення.

Основні вимоги до продуктивності:

- один кадр повинен рендеритися менш ніж за 1 секунду при роздільній здатності 1920x1080 пікселів;

- у режимі перегляду (viewport) сцена повинна відтворюватися зі швидкістю не менше 30 кадрів на секунду;

- програма має стабільно працювати на середньостатистичних ПК (GTX 1060, 8–16 ГБ ОЗП, процесор середнього рівня);

- шейдери повинні підтримувати гнучке налаштування параметрів;

- візуальні ефекти мають відповідати стилізованому художньому підходу;

- не допускається використання важких обчислень, що значно впливають на час рендеру або плавність роботи.

Ці вимоги сформовані з урахуванням специфіки задачі — розробити шейдери, які дозволяють створювати стилізовані та емоційно виразні сцени без значного технічного навантаження на систему.

Перш за все, швидкий рендеринг кадру (менше 1 секунди при Full HD)

дозволяє ефективно працювати з великою кількістю варіантів сцен і матеріалів. Це особливо важливо при створенні анімацій або інтерактивного контенту, де користувач очікує швидкого зворотного зв'язку після внесених змін.

Плавна робота в режимі перегляду (viewport) зі швидкістю від 30 FPS — критично важлива під час налаштування сцен, камер, світла та матеріалів. Якщо інтерфейс "гальмує", це ускладнює роботу й суттєво уповільнює весь процес створення. Саме тому оптимізація для viewport має таку саму вагу, як і оптимізація рендеру.

Забезпечення сумісності з комп'ютерами середнього рівня робить систему доступною для ширшого кола користувачів. Проєкт не орієнтований на вузькоспеціалізоване апаратне забезпечення, тому ефективна робота на умовно бюджетному ПК є необхідною умовою.

Гнучке налаштування шейдерів дозволяє адаптувати сцену під конкретні потреби: зменшити навантаження на систему, змінити візуальний стиль, увімкнути або вимкнути додаткові ефекти. Це розширює можливості шейдерів і робить їх універсальними в різних типах проєктів.

Нарешті, візуальна складова відіграє ключову роль. Стилізовані сцени відрізняються тим, що не прагнуть до фотореалізму, а передають настрій і художню ідею через спрощення, символізм і кольорові акценти. Шейдери мають підкреслювати ці риси за допомогою спеціальних ефектів, таких як контурне затінення чи нестандартне освітлення.

1.4 Постановка завдань

У ході виконання дипломної роботи необхідно вирішити низку послідовних завдань, що охоплюють усі етапи розробки стилізованої тривимірної сцени з оптимізованими шейдерами у середовищі Blender. Спершу слід провести ґрунтовний аналіз предметної області — вивчити сучасні методи стилізованого

рендерингу, техніки художньої стилізації, принципи побудови шейдерів, а також обмеження систем рендерингу в реальному часі. Доцільно також ознайомитись із наявними підходами у професійному середовищі, щоб сформувати більш цілісне уявлення про можливості реалізації проєкту. На основі цього аналізу потрібно сформувати технічне завдання, у якому чітко визначити вимоги до візуальної якості, продуктивності та бажаного художнього стилю сцени, а також попередньо описати функціональні та технічні обмеження.

Далі необхідно виконати проєктування сцени, включаючи побудову базового середовища, продуману композицію, логічне розташування об'єктів, налаштування освітлення та загального колористичного балансу. Одним із важливих завдань стане створення процедурної системи трави за допомогою Geometry Nodes — з урахуванням варіативності геометрії, можливості динамічного керування параметрами об'єктів та подальшої масштабованості сцени. Після цього слід зосередитися на створенні власних шейдерів у редакторі вузлів Shader Editor, реалізувавши стилізоване освітлення, ефекти cel-shading, контурів, процедурного фарбування, прозорості та кольорової варіативності з використанням шумових текстур і нормалізованих координат.

Окремим етапом стане оптимізація — необхідно зменшити кількість обчислень у вузлах, уникати надмірно складних структур, використовувати процедурні текстури замість імпортованих, забезпечити мінімальне навантаження на GPU, особливо у режимі реального часу та під час роботи в Eevee. Оптимізація також передбачає аналіз часу рендерингу і моніторинг споживання ресурсів сценою.

На фінальному етапі необхідно налаштувати камеру, підібрати найкращі ракурси, підготувати презентаційні кадри сцени, додати візуальні акценти й, за потреби, інтегрувати скайбокс, щоб покращити загальну цілісність композиції, уникнути порожнього фону та підкреслити атмосферу середовища. Усі ці завдання повинні бути реалізовані послідовно, з урахуванням як технічних вимог, так і художнього бачення фінального результату, що дозволить досягти гармонійного балансу між естетикою та ефективністю.

2 РОЗРОБКА БАЗОВИХ 3-D СЦЕН

Шейдери можуть бути застосовані в багатьох сценаріях та варіантах. Щоб гарно побачити застосування стилізованих шейдерів на достатньому спектрі сцен, ми розробимо 2 головні сцени:

- indoor-lit сцену (освітлену сцену приміщення);

- outdoor-lit сцену (освітлену сцену зовнішнього середовища).

Вибір саме цих двох типів сцен зумовлений бажанням охопити найпоширеніші ситуації, у яких використовуються стилізовані шейдери, та водночас продемонструвати їхню гнучкість і ефективність в умовах різного освітлення. Outdoor-сцена на природі дозволяє показати, як шейдери поведуться при денному освітленні, з великою кількістю розсіяного світла, атмосферою перспективою, ефектами неба, води або листя. Це середовище дає змогу протестувати стилізацію природних матеріалів: ґрунту, рослинності, каміння, неба — і водночас оцінити, наскільки зберігається художній стиль при зміні кута огляду, часу доби чи погодних умов.

Indoor-сцена, навпаки, реалізована з нічним або слабким штучним освітленням. Це дозволить зосередитись на нюансах взаємодії шейдерів зі світлом у локалізованих ділянках сцени — наприклад, як реагує матеріал на джерело світла поряд, чи правильно відображаються тіні, блики, емісійні ефекти. Такий підхід добре підкреслює важливість точного налаштування освітлення й поведінки матеріалу в умовах обмеженого світлового середовища, що є критично важливим для створення атмосферних сцен.

Разом ці дві сцени дають змогу комплексно перевірити шейдери в умовах, максимально наближених до реальних або художньо-стилізованих сценаріїв, та виявити потенційні слабкі місця в їхній реалізації.

Перед тим як перейти до складних композицій та стилізації, доцільно розпочати з найпростішої базової сцени, яка дозволяє швидко перевірити коректність роботи освітлення, камери та матеріалів. У середовищі Blender

типовою відправною точкою є сцена, що містить один об'єкт (куб), джерело світла та камеру

Ця сцена створюється автоматично при відкритті нового проєкту в Blender, і, попри свою простоту, вона є важливою відправною точкою для подальшої роботи. Надалі ця базова сцена була адаптована для створення двох окремих варіантів — внутрішньої кімнати з локальними джерелами світла та відкритого зовнішнього простору з природним освітленням(рис 2.1).

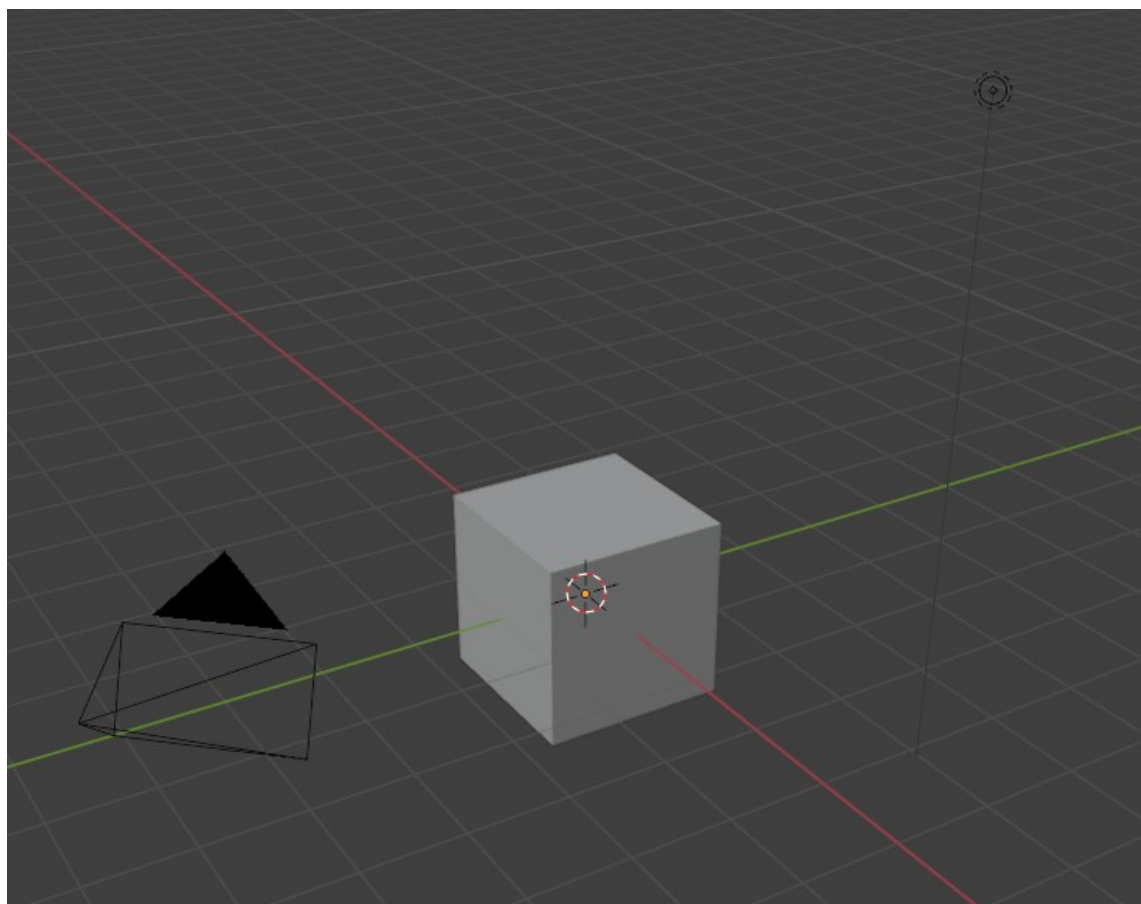


Рисунок 2.1- Базова 3-D сцена в блендері

Розробка моделей для 3D-сцен є важливим етапом створення переконливого віртуального середовища. Процес моделювання зазвичай починається з базових примітивів (куб, сфера, циліндр), які поступово деталізуються за допомогою редагування вершин, ребер і граней у режимі Edit Mode. У Blender існує широкий набір інструментів і технік, які дозволяють

створювати як реалістичні, так і стилізовані об'єкти, наближені до тих, що ми бачимо щодня.

Одна з основних методик — це полігональне моделювання, коли форма об'єкта створюється через зміну топології сітки. Для більш складних форм застосовується сабдивізіонне згладжування (Subdivision Surface), що дає змогу отримати плавні, округлі поверхні без великої кількості полігонів.

Коли йдеться про об'єкти з органічними або складними формами (наприклад, меблі, посуд, побутова техніка), часто використовуються модифікатори (наприклад, Mirror, Array, Boolean), які значно пришвидшують роботу та дозволяють легко редагувати симетричні або повторювані елементи.

Для побутових або інтер'єрних сцен особливо важливим є дотримання правильних пропорцій та масштабу. Blender дозволяє точно працювати з реальними розмірами, що важливо для створення правдоподібних середовищ. Крім того, візуальна переконливість досягається за допомогою UV-розгортки та нанесення текстур, які додають об'єктам додатковий рівень деталізації без ускладнення геометрії.

У нашому випадку процес моделювання починається з підбору фото-референсів. Це можуть бути фотографії реальних місць, об'єктів або архітектурних елементів, які мають стати основою для створення стилізованої, але впізнаваної сцени. Наявність якісного візуального референсу дозволяє краще розуміти пропорції, матеріали, структуру об'єктів, а також ухвалювати зважені рішення щодо стилізації, спрощення чи абстрагування форм.

Після збору референсів створюється базовий макет сцени — груба геометрія інтер'єру, яка визначає основну структуру простору: стіни, підлога, дверні прорізи, великі меблі. Цей етап дозволяє оцінити масштаби, перспективу, основне освітлення та камеру. Важливо з самого початку дотримуватись реальних розмірів об'єктів, тому в Blender часто використовуються точні числові значення та інструменти вирівнювання.

Після побудови основи сцени моделювання переходить на наступний рівень — створення дрібних, стилістично важливих об'єктів: стільці, світильники,

посуд, книги, декор. Тут активно застосовуються такі інструменти, як Bevel Modifier — для згладжування різких кутів, що робить моделі більш реалістичними і дозволяє краще працювати зі світлом. Бевелінг також надає об'єктам відчуття фізичної присутності, оскільки в реальному світі практично не існує ідеально гострих кутів.

Для певних об'єктів, таких як килими, тканини або рослини, використовується Hair System (система волосся), яка у Blender дозволяє створювати волокнисті або м'які поверхні. У сценах з інтер'єром така система може застосовуватись не тільки для волосся персонажів, але й для моделювання трави, хутра, щетини, ворсу та подібних матеріалів, що значно збагачують візуальний вигляд сцени[3].

У фінальному етапі відбувається розстановка об'єктів, тобто композиційне наповнення сцени з урахуванням основних принципів дизайну: баланс, акценти, глибина, ритм. Саме в цей момент сцена починає виглядати живою і наповненою сенсом — кожен предмет має своє місце, орієнтацію і функцію. Тільки після того як кожний етап виконано і сцена виглядає цілісно можемо ми починати задумуватись щодо освітлення та стилізації матеріалів.

2.2 Розробка indoor-lit сцени

Було вирішено розробляти стилізовану сцену приміщення з переважно темним освітленням, що симулює нічну пору доби. Для референсу була підібрана сцена з рисунка 2.2

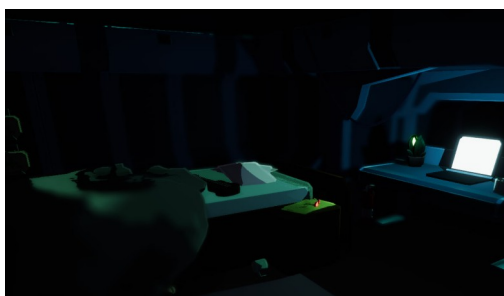


Рисунок 2.2— Референс сцени приміщення з обмеженим освітленням

За цим референсом побачимо основні об'єкти що нам треба було розробити:

- ліжко;
- подушки;
- ковдра;
- матрац;
- тумбочка;
- книги;
- світильник або лампа;
- ноутбук;
- горщик з рослиною;
- чашка;
- укулеле;
- стінові панелі;
- віконні ніші;
- стельові балки;
- підлога;
- пляшка або флакон;
- килим;
- декоративне освітлення (наприклад, LED або нічник);
- окуляри.

У процесі створення сцени всі об'єкти моделюються з урахуванням їх функціональності, розміру та візуального впливу на композицію. Основні великі елементи, такі як ліжко, стіл, тумбочка, а також архітектурні складові — стіни, підлога, стельові балки та віконні ніші — були змодельовані на основі простих примітивів (кубів), з подальшим редагуванням за допомогою інструментів *extrude*, *inset*, *loop cuts* тощо. Для надання об'єктам реалістичності були використані модифікатори *bevel* (для згладження кутів) та *mirror* (для зручності роботи з симетричними формами)(рис2.3)[4].

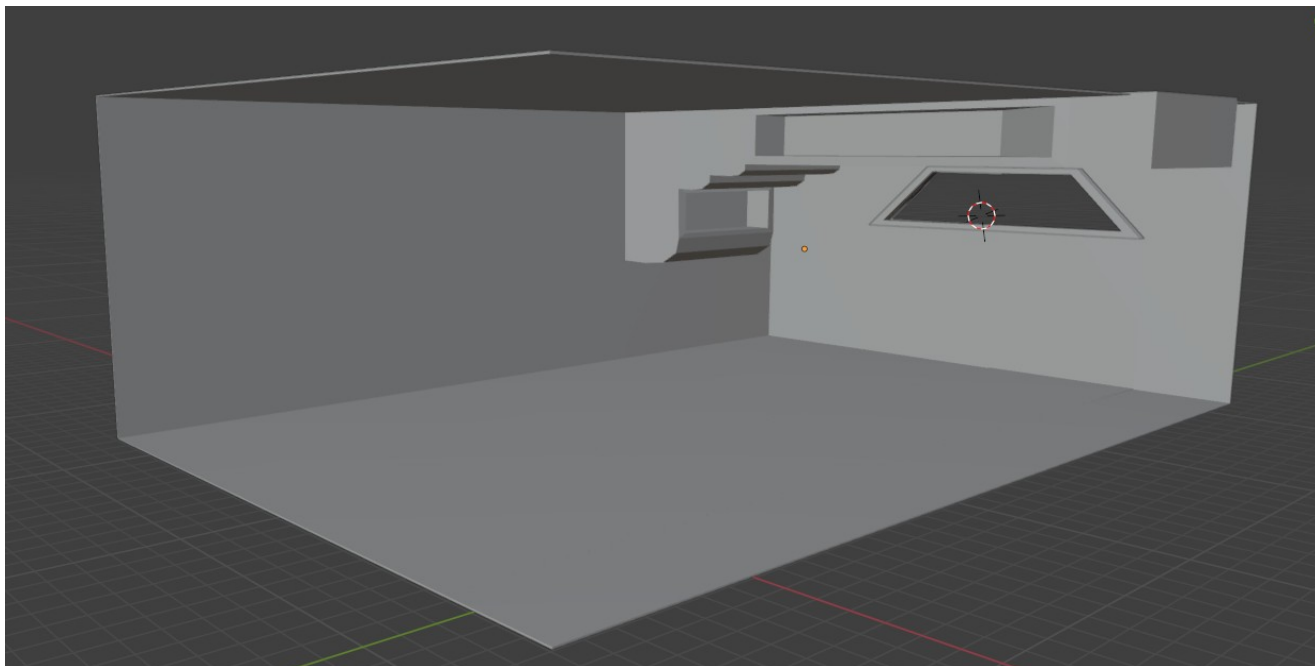


Рисунок 2.3 -Груба геометрія кімнати

Після цього заповнюємо сцену більш базовими меблями та геометрією, додаємо стіл, ліжко та тумбочку. Які змодельовали за тими ж самими техніками що розглядалися до цього(рис 2.4).

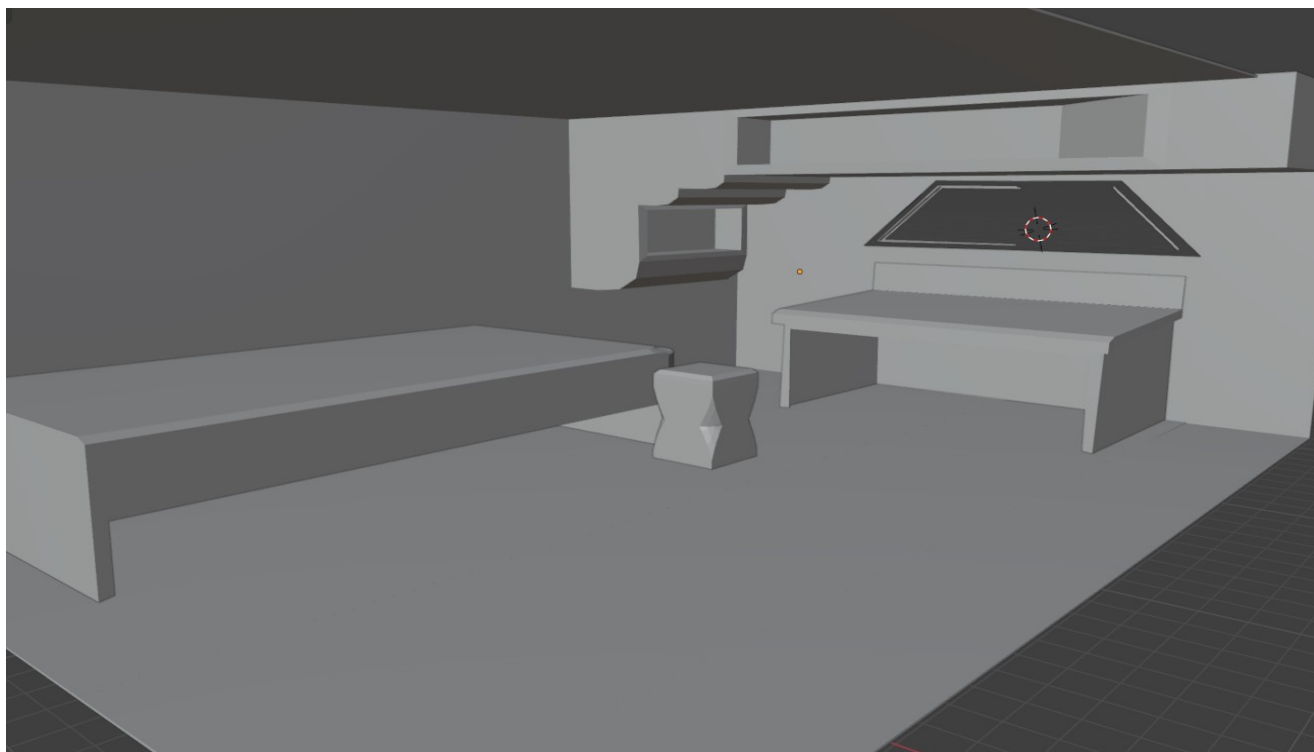


Рисунок 2.4 — Сцена після додання базової геомтерії

М'які об'єкти, як-от подушки, ковдра чи матрац, потребують природної деформації та вигину. Для цього було застосовано або sculpting, або симуляція тканин (cloth simulation), що дозволяє досягнути правдоподібних результатів при мінімальних витратах часу(рис2.5).

Cloth Simulation — це система фізичної симуляції тканини в Blender, яка дозволяє моделювати об'єкти з м'якою або гнучкою структурою. Вона працює на основі сітки (mesh), яка піддається гравітації, зіткненням з іншими об'єктами та іншим фізичним впливам. Завдяки цій системі можна досягти реалістичного драпірування, складок і руху тканини, що особливо корисно при створенні покривал, подушок або одягу(рис 2.6).

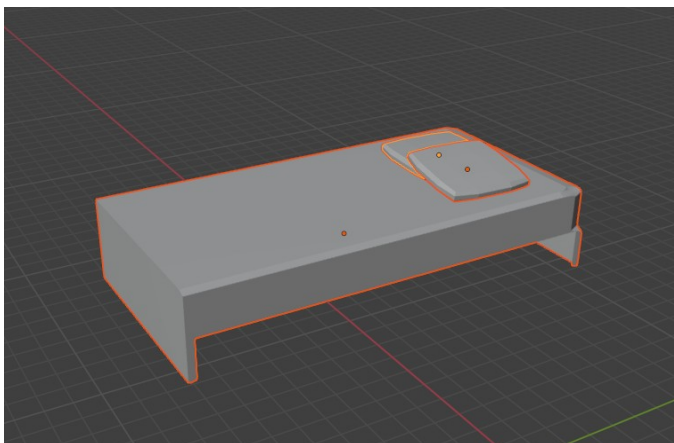


Рисунок 2.5- Ліжко до додавання ковдри за допомогою симуляції



Рисунок 2.6 — Ліжко після додавання ковдри

Декоративні деталі, зокрема чашка, горщик з рослиною, укулеле, окуляри, а також ноутбук — створюються вручну або з використанням кривих (curve objects), що дозволяє легко створювати вигнуті форми та органічні лінії(рис 2.8). У складних випадках застосовуватиметься subdivision surface для деталізації.

Для килима була використана окрема техніка моделювання за допомогою системи волосся, що дозволяє швидко створити хутрянну поверхню килима за відносно невелику вартість геометрії[5].

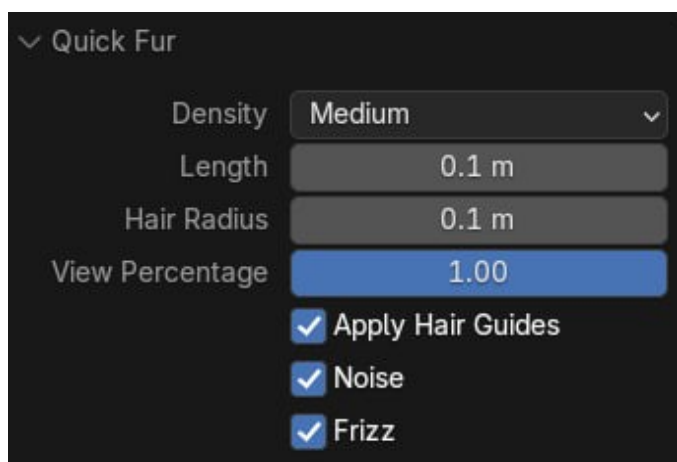


Рисунок 2.7 — Налаштування системи хутра

Було обрано середню густоту ворсу (Medium), що забезпечує баланс між щільністю волокон і продуктивністю сцени. Довжина ворсу встановлена на 0.1 метра — це відповідає високому та пухнастому килиму, який має виражену текстуру. Також було збільшено радіус кожної волосини до 0.1 метра, що дозволяє візуально сприймати ворс як товстий, важкий і м'який матеріал. Додатково було увімкнено напрямні волосся (Apply Hair Guides), шум (Noise) та ефект завивки (Frizz), які створюють природний безлад, характерний для справжнього ворсу. Завдяки такому підходу ми отримали деталізовану, а головне не важку за геометрією сцену придатну для стилізації(рис 2.7).

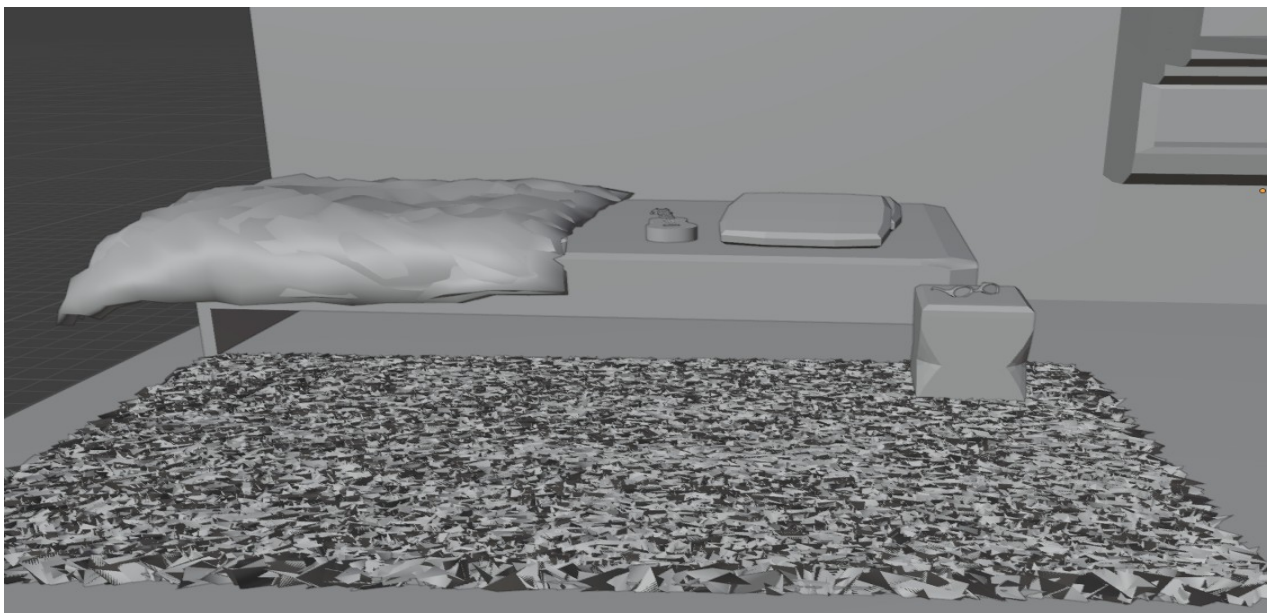


Рисунок 2.8 — Сцена після додавання додаткових елементів

Після створення базової структури приміщення та основних великих об'єктів, таких як ліжко, стіни, підлога та стельові балки, ми перейшли до деталізації сцени. Саме на цьому етапі відбувається наповнення простору дрібнішими, але не менш важливими елементами, які додають сцені характеру, реалістичності та завершеності. Серед них — ноутбук, який вказує на функціональність простору як місця для роботи або відпочинку; настільна лампа, що виконує не лише практичну функцію освітлення, але й створює додаткові світлові акценти та вносить до сцени елемент природи та кольорову рівновагу (рис. 2.9).

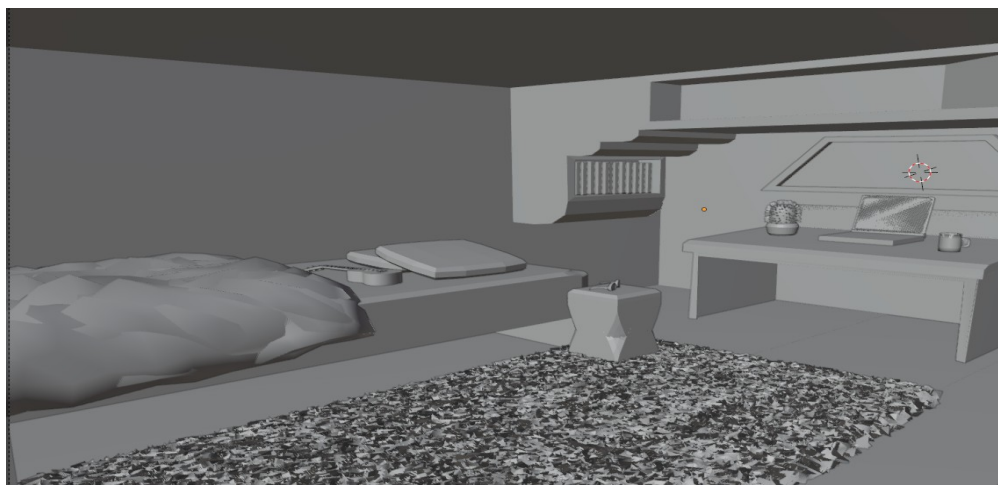


Рисунок 2.9 — Закінчена неосвітлена сцена

Уся сцена дотримується метричної системи одиниць, що забезпечить правильне масштабування моделей, а також дозволить фізично точніше обраховувати освітлення, взаємодію об'єктів та шейдерні ефекти. Такий підхід сприяє не тільки візуальній узгодженості, але й оптимізації робочого процесу в Blender.

2.3 Створення outdoor-lit сцени

Після завершення роботи над деталізованою indoor сценою ми можемо перейти до розробки outdoor-сцени, яка дозволить продемонструвати інші можливості Blender у стилізації великих відкритих просторів. Blender — це надзвичайно гнучкий інструмент, який надає широкі можливості не лише для моделювання та рендерингу інтер'єрів, але й для створення розлогих зовнішніх середовищ. Завдяки підтримці складних шейдерів, систем частинок, геометричних нодів та реалістичного (або стилізованого) освітлення, Blender дозволяє досягати глибоко стилізованих результатів навіть у роботі з такими складними сценами, як природні ландшафти, міські середовища чи фантастичні простори.

У цій частині ми використаємо вже розроблені нами інструменти — шейдери, знання про освітлення та стилізацію — для створення outdoor-сцени, яка розкриватиме художній потенціал Blender у роботі з простором, глибиною, атмосферними ефектами та природним освітленням.

Щоб задати чіткий художній напрямок для нашої outdoor-сцени, ми обрали референс — зображення мальовничого відкритого ландшафту, виконаного у стилі, що нагадує японську анімацію. Цей стиль відомий своєю чистотою форм, глибоким колористичним балансом та яскравим зображенням атмосферних ефектів. Саме він слугує натхненням для подальшого стилізованого моделювання у Blender.

На зображенні, яке ми обрали як референс, зображено простору зелену долину з різноманітними відтінками трави, що створюють враження глибини та об'єму. У центрі композиції височіє металева телевежа — високий штучний об'єкт, який створює контраст із м'якою природою навколо. По обидва боки вежі розташовані дерева з густим листям, промальованим у стилі мінімалістичних масивів кольору, з відсутністю дрібної текстури, але з виразною грою світла і тіні. На задньому плані — вражаюче детальні хмари, які створюють відчуття простору, глибини та гарної денного освітлення. Колір неба поступово переходить від більш насиченого синього вгорі до світлішого тону ближче до горизонту, додаючи сцені реалістичної повітряної перспективи.

Цей референс став основою для побудови нашої outdoor-сцени у Blender. Ми намагатимемося відтворити не лише геометричні форми, а й загальну атмосферу: тепле денне освітлення, чисте небо, насичене зелене середовище, та приємну, трохи ностальгічну композицію, де технології зустрічаються з природою (рис 2.10). Така стилістика дозволить застосувати розроблені нами шейдери — toon, watercolor, dither та outline — у новому контексті, підкреслюючи якінструментальні можливості Blender, так і гнучкість нашого підходу до візуальної стилізації.

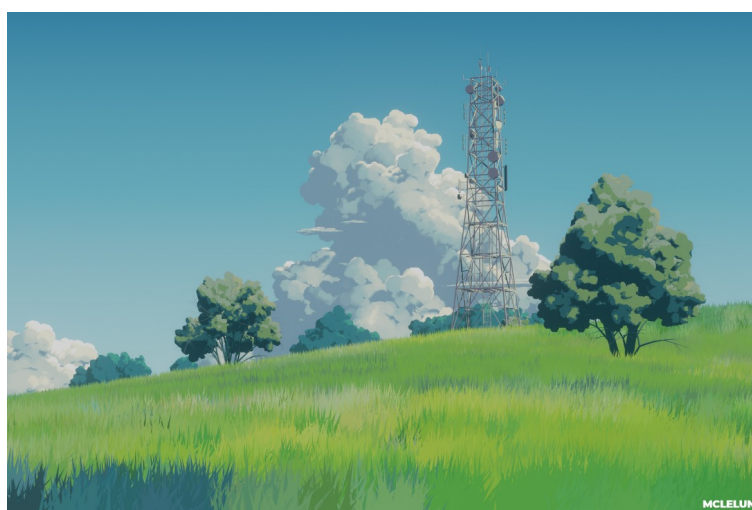


Рисунок 2.10 — Референс для створення outdoor-lit сцени

Як завжди, починаємо нашу роботу зі стандартної сцени в Blender, яка містить базові об'єкти — куб, камеру та джерело світла. Ця сцена є відправною точкою для більшості проєктів і надає нам базову структуру, яку можна поступово ускладнювати. У нашому випадку ми не змінювали радикально початкову конфігурацію, адже вона вже містить основні елементи, необхідні для старту роботи. Натомість ми вдосконалили її поступово, у рамках загальної стилізації outdoor-сцени.

Першим кроком стало додавання площини (plane) під наявний у сцені куб (рис 2.11). Це досить проста, але надзвичайно важлива дія — площина стала основою для створення стилізованого трав'яного поля, яке займає значну частину сцени. Ми переходимо у режим редагування або ж працюємо з об'єктом у Object Mode, і значно збільшуємо площину по осях X і Y — таким чином ми отримаємо достатню площу, яка була заповнена деталями, що імітують траву, а також дозволила б нам розмістити на ній інші елементи: дерева, вежу, або інші об'єкти довкілля.

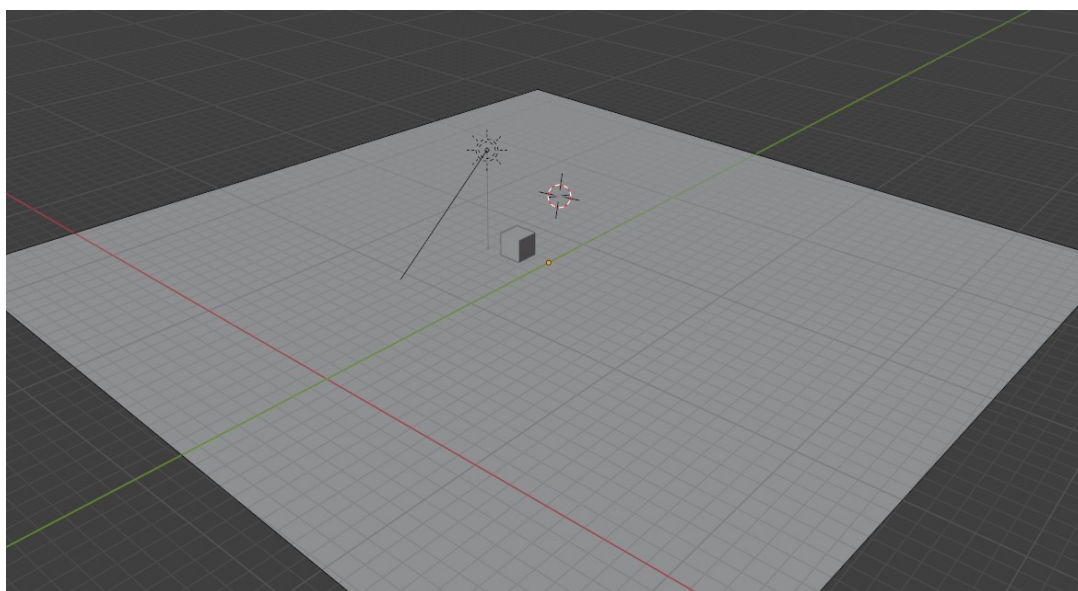


Рисунок 2.11 — Сцена з доданою площиною на даному етапі

Тепер ми можемо перейти до наступного етапу створення outdoor-сцени — формування трав'яного покриття. Візуальна стилізація довкілля значною мірою

залежить від того, наскільки органічно та природно виглядають елементи середовища, а трава — це одна з найважливіших складових такої сцени.

Для створення трави ми скористаємось знайомим нам підходом, який ми вже застосовували при генерації килима — це використання системи волосся (Hair System). Однак, на відміну від попереднього прикладу з килимом, де ми працювали з налаштуваннями, орієнтованими на коротке і щільне хутро, цього разу ми змінимо підхід. Замість того, щоб створювати хутро, ми сконцентруємося на використанні більш натуральної системи волосся, яка дозволяє генерувати окремі довгі елементи, що природно згинаються, розміщуються з певною випадковістю, та здатні імітувати реалістичну чи стилізовану траву.

Ключовою частиною нашого процесу було використання Geometry Nodes — це потужний інструмент в Blender, який надає змогу з великою гнучкістю створювати, редагувати та модифікувати геометрію за допомогою візуального програмування. Geometry Nodes відкривають величезні можливості для процедурного моделювання, що дозволяє досягати високого рівня деталізації без потреби вручну розміщувати тисячі елементів[6]. Ми зосередимо увагу саме на процедурному підході, який дозволить нам легко масштабувати сцену, змінювати щільність, довжину, напрямок та інші параметри трави без необхідності перебудовувати всю сцену з нуля.

Варто наголосити, що у цьому проєкті ми не поклалися на готові пресети або стандартні інструменти, а натомість вибудовували систему самостійно, поступово додаючи необхідні вузли (nodes) та працюючи з ними всередині Geometry Nodes Editor (рис2.12). Цей підхід не лише розширює наше розуміння процедурної графіки, а й дозволяє досягати точного візуального контролю над кожною складовою сцени.

Також важливою частиною нашого процесу був nondestructive workflow, який нам надає блендер. Nondestructive workflow (недеструктивний робочий процес) у Blender — це підхід до створення 3D-сцен і об'єктів, при якому жодні дії не змінюють остаточно вихідні дані, а всі операції залишаються зворотними та редагованими.

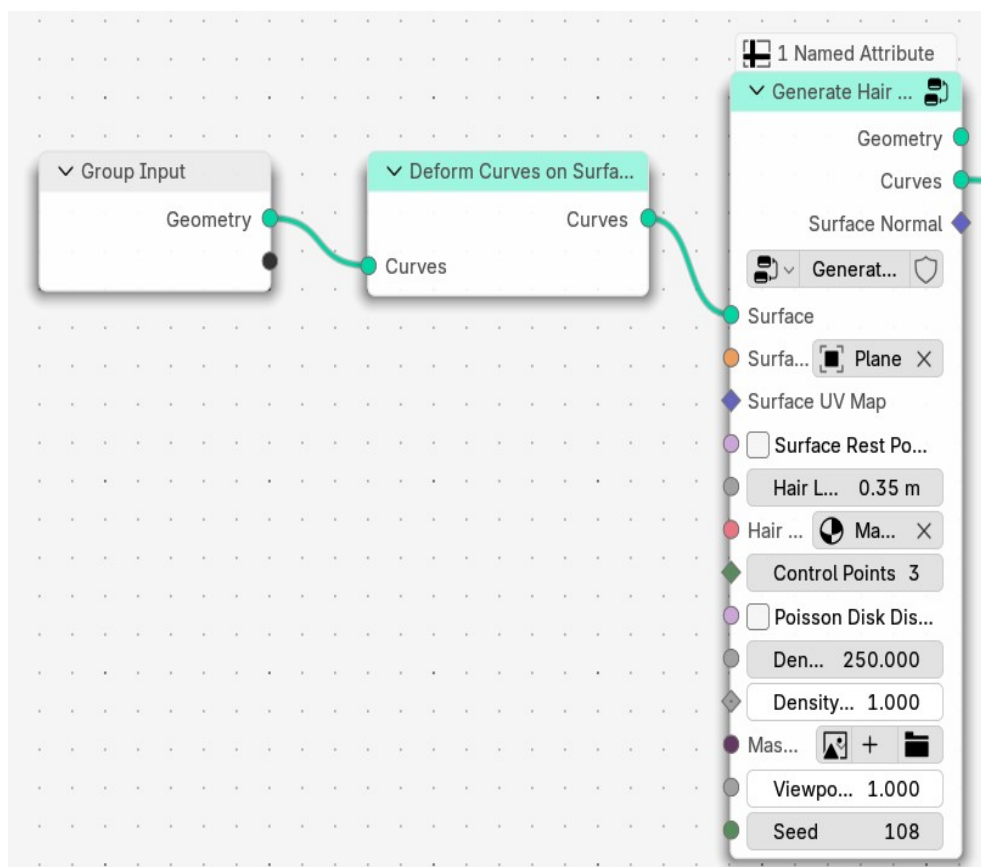


Рисунок 2.12 — Перші ноди необхідні для створення системи трави

Щоб створити волосся за допомогою Geometry Nodes, ми починаємо з вузла Group Input, який приймає вхідну геометрію нашої сцени. Далі ми підключаємо до нього вузол Deform Curves on Surface. Цей вузол має важливе значення, адже він бере криві та деформує їх, прив'язуючи до певної поверхні, яку ми вказуємо. У випадку з волоссям, це дозволяє нашим майбутнім волоскам правильно лягати на об'єкт, ніби вони ростуть з його поверхні (рис. 2.13).

Потім ми додаємо вузол Generate Hair Curves. Саме він генерує реальні криві, які й виглядають як волосся. До нього підключається геометрія, яка слугує основою для росту волосся, і в цьому вузлі ми можемо налаштувати безліч параметрів, щоб досягти бажаного вигляду: від довжини волосся та його густоти, до кількості контрольних точок на кожній волосинці, що впливає на її гнучкість, та навіть використання карт густоти для більш детального контролю над розподілом волосся. Він також дозволяє вибрати поверхню, на якій генерується волосся, як от "Plane" (Площина) у вашому прикладі, і використовувати UV-карту

цієї поверхні для точнішого контролю. Це дає нам повний контроль над тим, як волосся взаємодіє з поверхнею і як воно виглядає загалом. В результаті отримаємо маленькі волосинки які не мають товщини.

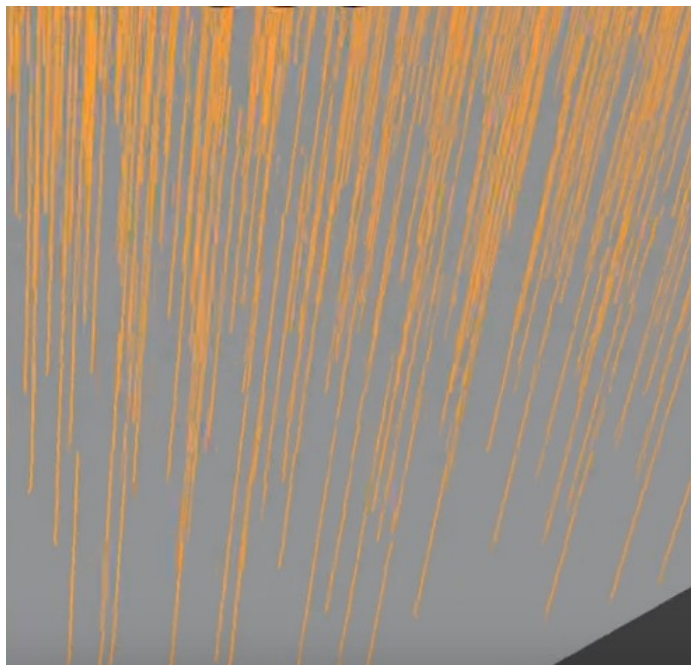


Рисунок 2.13- Площина з доданою до неї системою волосся

В результаті отримали маленькі волосинки які не мають товщини. Тож випрвимо це додаванням ще декількох вузлів(рис 2.14).

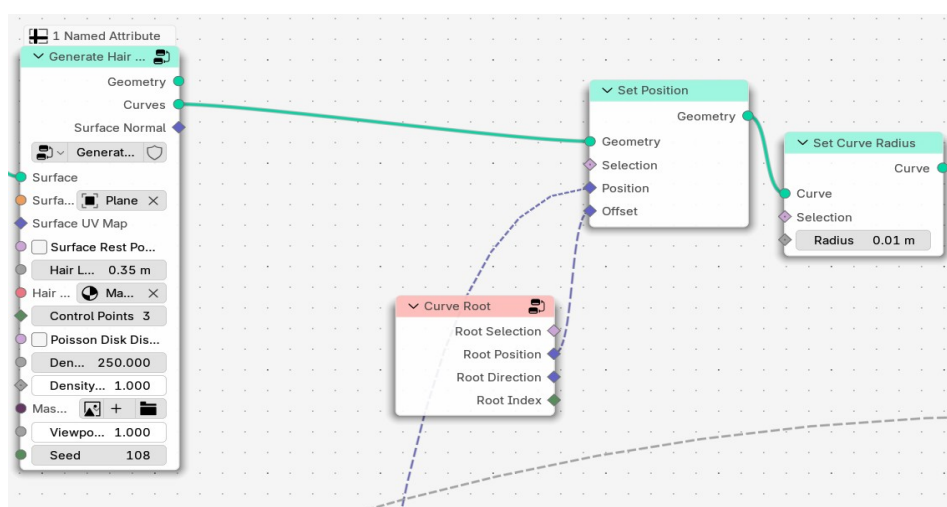


Рисунок 2.14 — Надання траві товщини за допомогою вузлів

Спочатку ми додаємо вузол Set Position (Задати Позицію). Цей вузол дозволяє нам зміщувати точки геометрії, і в даному випадку він підключений так, що його "Geometry" (Геометрія) отримує дані від вузла "Generate Hair Curves". До входу "Offset" (Зміщення) підключений вузол Curve Root (Корень Кривої). Це означає, що ми зміщували позицію точок кожної кривої, але тільки відносно її кореня (рис 2.15). Це може бути використано для того, щоб, створити певний напрямок росту або деформацію біля основи волосся.

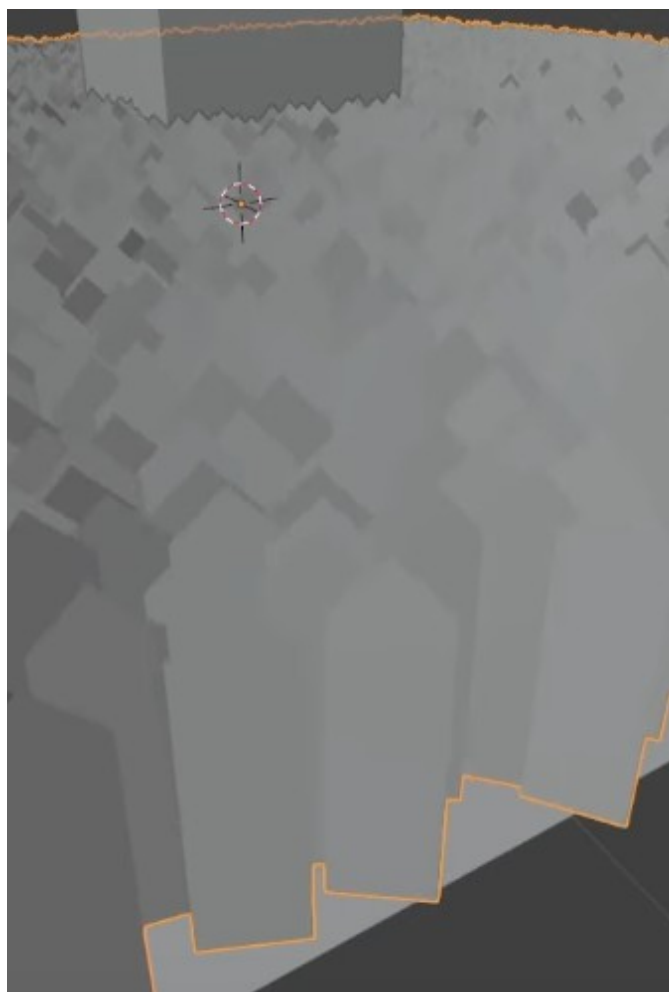


Рисунок 2.15— Демонстрація доданої товщини в системі трави

На даному етапі ми вже отримали базову генерацію трави за допомогою системи волосся в Geometry Nodes, однак візуальний результат все ще виглядає неприродно. Усі травинки мають однакову довжину і ростуть строго вертикально, що не відповідає природному вигляду. Щоб це виправити, ми додамо декілька

нодів у Geometry Nodes, які допоможуть створити випадкові викривлення. Це дозволить кожній травинці трохи відхилятися від вертикалі, імітуючи вплив вітру та природну нерівномірність (рис 2.16).

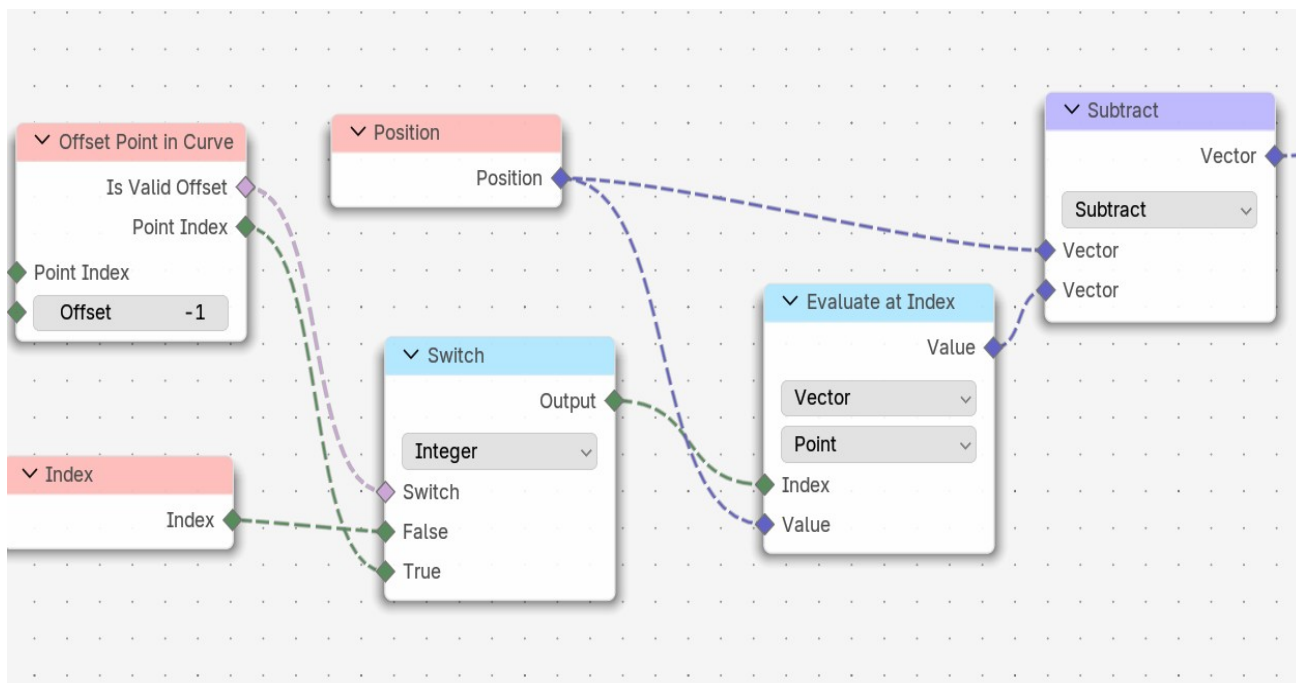


Рисунок 2.16 — Вузли що підготовлюють геометрію трави до руху

Це перший крок в для отримання цікавої різноманітної стилізованої трави. Ці вузли виконують розрахунок, який визначає напрямок і величину зміщення кожної точки трави відносно точки, що знаходиться безпосередньо над нею, або відносно кореня трави. Це створює "вектор руху" для кожної точки, який потім можна використовувати для симуляції коливання трави.

Спочатку вузол **Offset Point in Curve** (Зміщення Точки в Кривій). Він дозволяє нам отримати інформацію про позицію точки, що знаходиться на певній відстані вздовж кривої. У вашому випадку "Offset" встановлений на -1 отримує позицію попередньої точки на кривій від поточної. Це важливо, оскільки для моделювання руху нам потрібно знати, як одна точка розташована відносно іншої вздовж волосини трави.

Далі йде вузол **Position** (Позиція), який просто надає нам поточну позицію кожної точки на кривій трави.

Потім вузол Switch (Перемикач), використовується для вибору, Він підключається до вузла Index (Індекс), який, дає індекс поточної точки на кривій. Це дозволяє перемикачу вирішувати, яке значення використовувати залежно від індексу точки (наприклад, чи це корінь, середина чи кінчик трави).

Після цього ми бачимо вузол Evaluate at Index (Оцінити за Індексом), який, бере значення (наприклад, позицію) з певного індексу.

Нарешті, вузол Subtract (Віднімання) використовується для обчислення різниці між двома векторами. Він віднімає позицію однієї точки від позиції іншої. Якщо, наприклад, від позиції поточної точки віднімається позиція попередньої точки на кривій, це дає вектор, який вказує від попередньої точки до поточної. Цей вектор представляє собою напрямок сегмента кривої. Саме цей вектор і був використаний для створення руху трави, оскільки він показує, в якому напрямку і наскільки сильно кожна частина трави має коливатися, реагуючи на зовнішні впливи, такі як вітер.

Таким чином, ці вузли працюють разом, щоб обчислити напрямок і "гнучкість" кожного сегмента трави, що є основою для подальшої симуляції її руху.

Наступним шагом в анімуванні нашої трави стало під'єднання до неї нодів що забезпечать хаотичне коливання трави. На рисунку можна побачити приблизну структуру таких вузлів(рис 2.17).

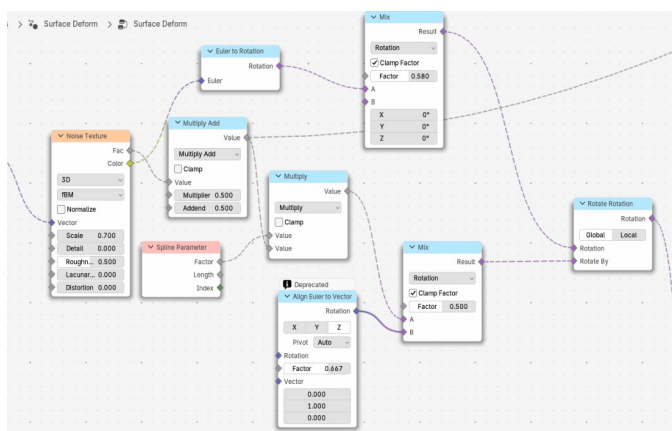


Рисунок 2.17 — Створення хаотичного руху за допомогою шуму

Першим у нашому ланцюгу вузлів є Noise Texture (Шумова Текстура). Цей вузол генерує випадкові значення, створюючи природні, органічні візерунки — саме те, що нам потрібно для симуляції хаотичного впливу вітру. У нашому випадку використовується 3D FBM (Fractal Brownian Motion) шум, який дає м'яке, багаторівневе варіювання. Це дозволяє зробити вигин кожної травинки унікальним[7].

Щоб адаптувати ці значення до подальшого використання, ми застосовуємо Multiply Add (Множити-Додати). Він дозволяє змінити амплітуду та змістити вихід шуму. Наприклад, множник на рівні 0.5 і доданок 0.5 допомагають нормалізувати шум у межах від 0 до 1. Таким чином ми отримуємо контрольоване, рівномірне базове значення, яке легко використовувати далі.

Наступним кроком іде Spline Parameter (Параметр Сплайна), що дає інформацію про положення кожної точки вздовж кривої — від кореня до кінчика. Це відкриває можливість зробити так, щоб коливання були сильнішими ближче до верхньої частини травинки, і майже відсутні в основі. Ці значення проходять через Multiply (Множення), який просто перемножує фактор позиції з модифікованим шумом. У результаті шум впливає лише частково, переважно на верхні частини трави, створюючи природну динаміку згинання. Далі ми використовуємо Align Euler to Vector (Вирівнювання Ейлера за Вектором), щоб перетворити цей напрямок вигину на обертання. Це дозволяє геометрії правильно реагувати на «вітер», орієнтуючись у випадкових напрямках, згенерованих шумом. І на завершення — Mix (Змішування). Цей вузол поєднує отримане обертання з базовим положенням травинки. Коефіцієнт змішування визначає, наскільки сильно ми дозволимо шуму впливати на поведінку кожної травинки, надаючи нам повний контроль над фінальним виглядом сцени.

Завдяки цим вузлам ми досягаємо реалістичного, динамічного руху трави, який імітує легке коливання на вітрі та додає сцені живості.

Тепер ми маємо всю логіку для того щоб реалізувати м'яке коливання трави та геометрію яка ця логіка має рухати тож об'єднаємо їх в вузол vector rotation і отримаємо коливання трави(рис 2.18).

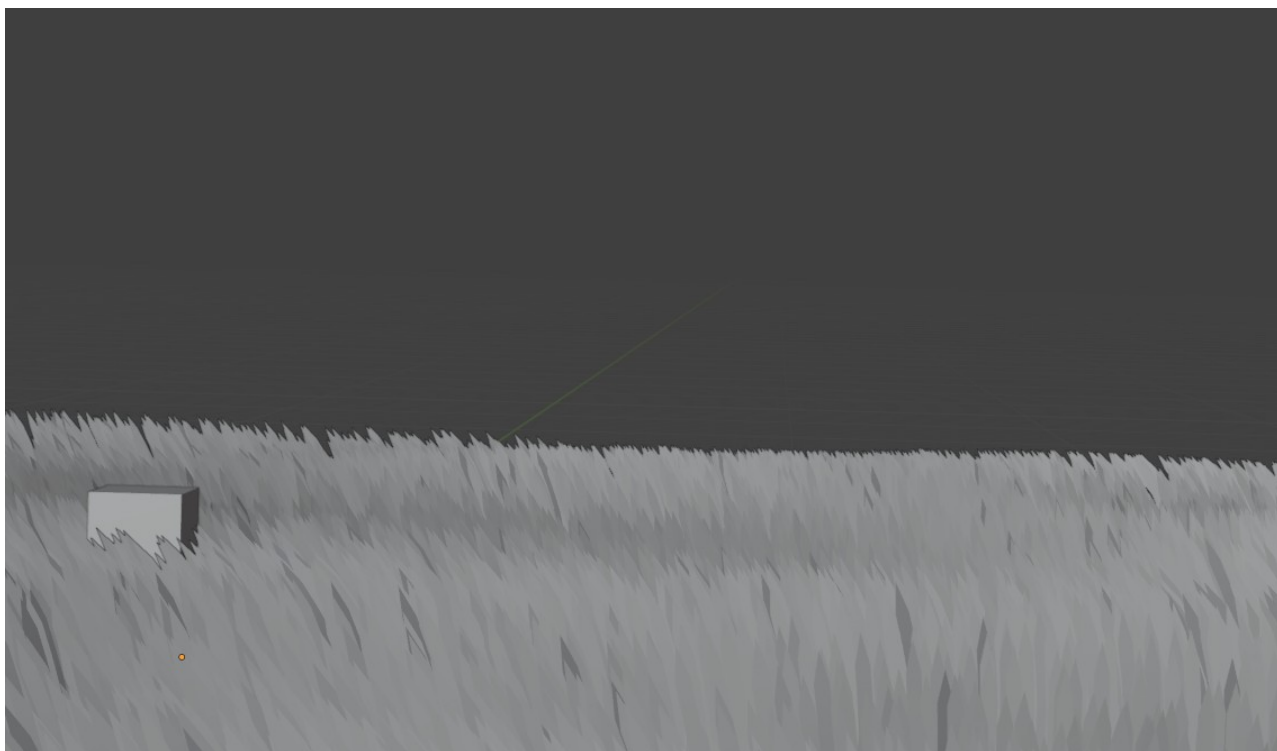


Рисунок 2.19 — Сцена після налаштування логіки та геометрії трави

Щоб візуально продемонструвати ефект руху порівняємо два різних кадра анімованої трави перш ніж перейдемо до більш детального освітлення та стилізації сцени(рис 2.20).

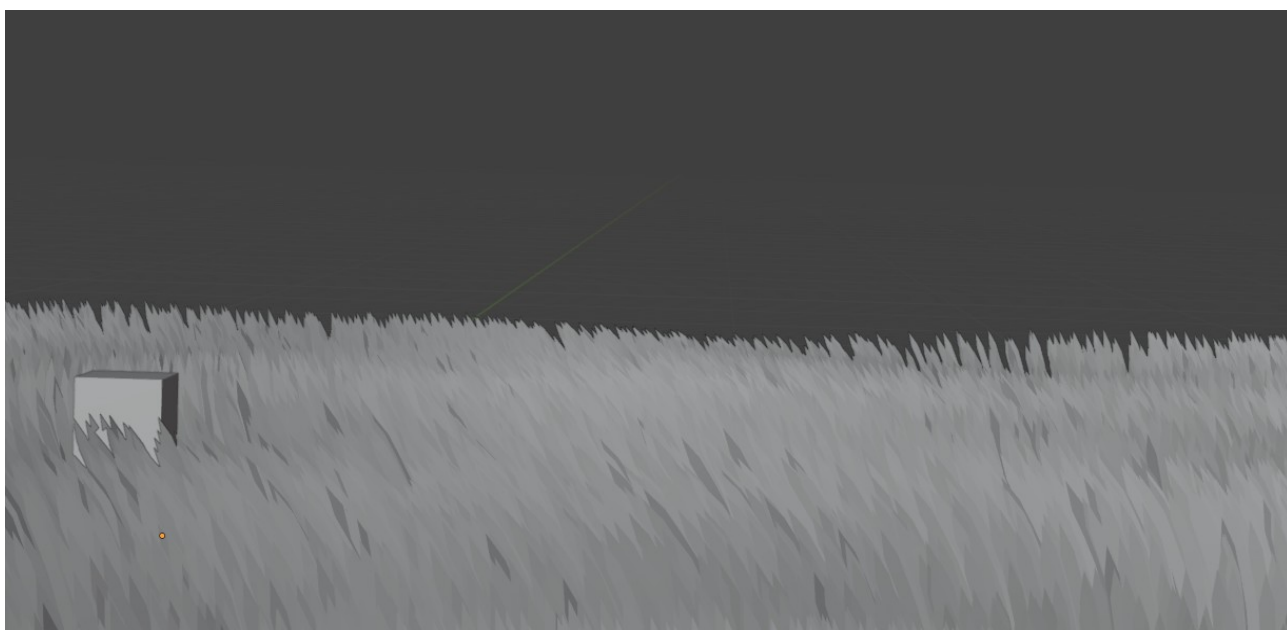


Рисунок 2.20 — Другий кадр для демонстрації руху трави

Як ми можемо спостерігати, зі зміною часу та активною роботою шумової текстури, деталі кожної окремої травинки зазнають динамічних змін. Травинки не просто коливаються в одну й ту ж сторону — кожна з них має унікальний характер руху: змінюється положення, кут нахилу, а також варіюється візуальний вигляд завдяки зміні значень по відтінках сірого. Це створює ефект органічного, майже хаотичного вітру, який не повторюється і виглядає природно.

На цьому етапі вся сцена ще досить нейтральна в кольорах, адже ми працюємо з базовими матеріалами. Всі зміни в кольорі поки що обмежуються відтінками сірого, які ми використовуємо як основу для тестування впливу шуму. Саме ці значення в майбутньому стануть базою для створення реального забарвлення трави — з градієнтами зеленого або стилізованими кольорами, в залежності від обраної художньої концепції.

Щойно ми додамо повноцінне кольорове рішення до сцени, динаміка коливання та зміни освітлення проявляться ще виразніше, і ми зможемо оцінити всю глибину нашої системи.

3 СТВОРЕННЯ ШЕЙДЕРІВ

3.2 Створення toon шейдеру

Toon-шейдер — стилізований шейдер, що імітує вигляд двовимірної графіки з чіткими кольоровими площинами без плавних градієнтів. Цей тип шейдера широко використовується в анімації, коміксах та мультфільмах, адже дозволяє передати образи яскраво, просто та виразно[8].

Toon-шейдери працюють шляхом розділення освітлення на чіткі рівні — замість поступового переходу від світла до тіні, поверхня об'єкта отримує конкретні зони освітлення залежно від кута між нормаллю поверхні та напрямком джерела світла. Саме це й забезпечує той характерний «площинний» вигляд.

У Blender процес створення базового Toon-шейдера є відносно простим і водночас ефективним. Цей тип шейдера не потребує складних обчислень або великої кількості вузлів, що дозволяє швидко досягти бажаного стилізованого вигляду. Основна ідея полягає в тому, щоб перетворити звичайне фізично-коректне освітлення на пласке, графічне представлення з чіткими межами між світлом і тінню.

Щоб реалізувати це, достатньо використати лише два основні вузли в Shader Editor'і: Shader to RGB і Color Ramp. Перший вузол конвертує інформацію про шейдер (освітлення, тіні) у кольорову інформацію, яку потім можна змінювати як звичайну текстуру. Другий вузол — Color Ramp — використовується для створення градації між світлом і тінню[9]. Щоб отримати чітке розділення зон освітлення, режим градієнта встановлюється на Constant. Це означає, що замість плавного переходу ми отримуємо чіткі блоки кольору, що і створює ефект, характерний для коміксів або мультфільмів(рис 3.1).

Завдяки такій мінімалістичній побудові Toon-шейдера, розробник отримує повний контроль над виглядом затінення — можна вручну визначити кількість тонів, їхні кольори та пороги між ними. Це відкриває широкі можливості для художньої експресії: сцена може мати як контрастний вигляд з двома-трьома рівнями тіней, так і більш м'яке розфарбування з кількома ступенями градації. Крім того, ці параметри легко анімувати або змінювати залежно від умов сцени, наприклад, часу доби чи положення джерела світла. Це робить Toon-шейдер не тільки засобом стилізації, а й потужним інструментом для візуального сторітелінгу у тривимірному середовищі.

Таким чином, усього кілька простих кроків дозволяють повністю змінити візуальне сприйняття об'єкта, зробивши його частиною стилізованої сцени. Нижче наведено приклад того яким чином це виглядає у редакторі шейдерів.



Рисунок 3.1 — Приклад створення toon shader-у в редакторі шейдерів

Для наглядної демонстрації роботи шейдерів використовують мавпячу голову. Нижче представлено порівняння звичайного матеріалу зі стандартними налаштуваннями та нашого шейдера з toon ефектом(рис 3.2, 3.3).

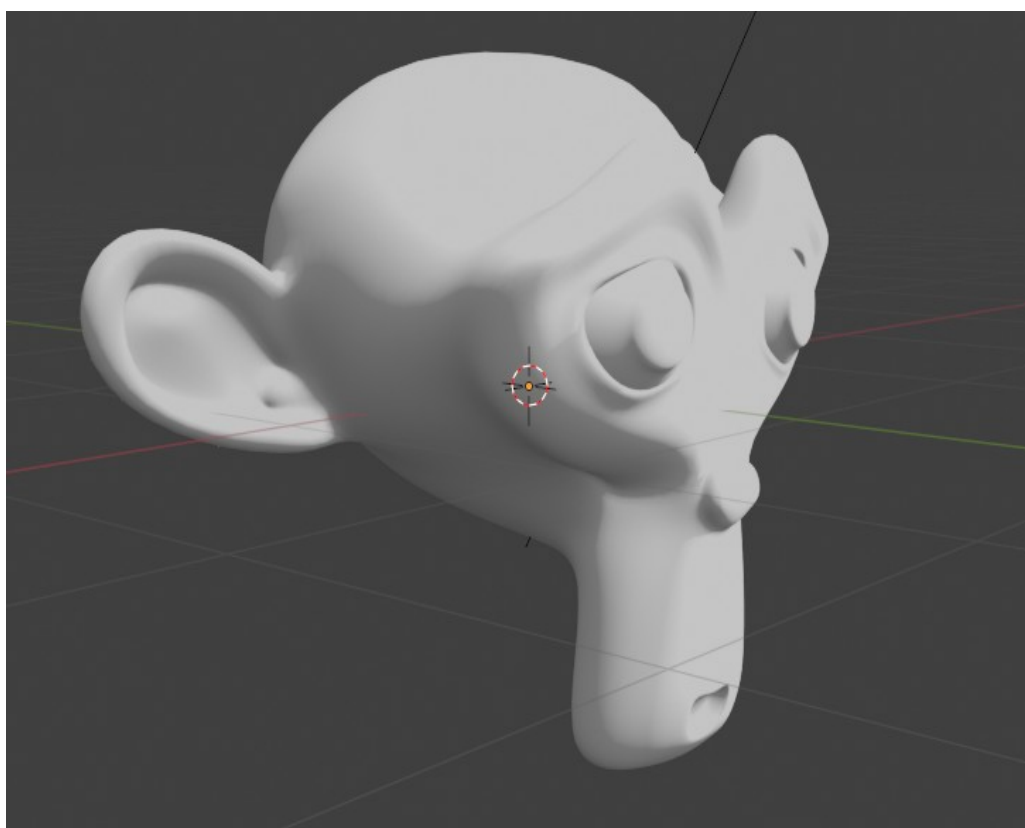


Рисунок 3.2 — Стандартний матеріал

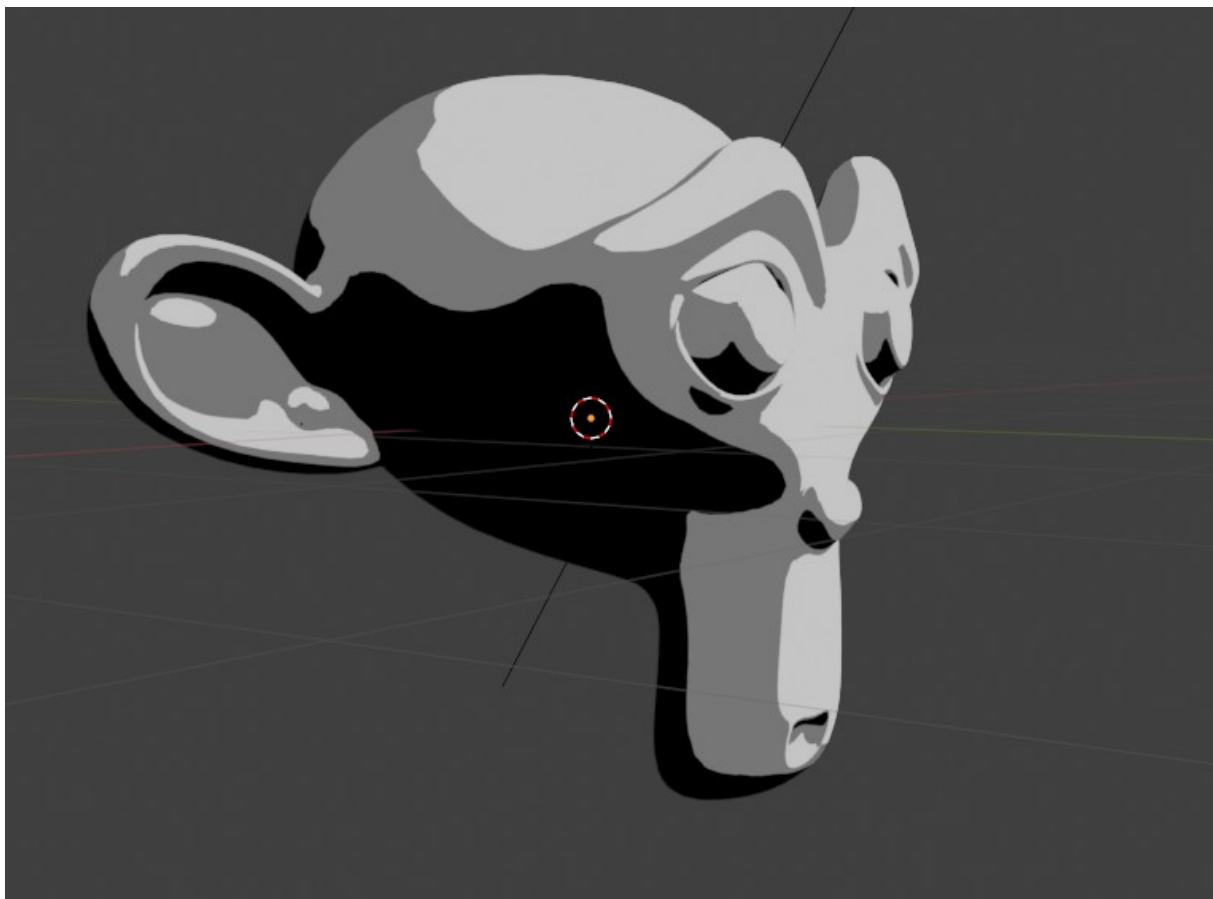


Рисунок 3.3- Мавпяча голова з застосуванням toon шейдеру

На двох рисунках дуже наглядно видно відмінності між двома підходами до затінення об'єктів. У випадку зі стандартним матеріалом, що використовує фізично-коректне освітлення, світло м'яко переходить у тінь, утворюючи градієнт — такий ефект наближений до реального сприйняття об'єктів у природному середовищі. Це дозволяє досягати високого ступеня фотореалістичності, але не завжди підходить для художньо-стилізованих сцен.

Натомість у Toon Shader тіні мають чітко визначені межі, які не змиваються і не розмиваються, а утворюють різкий контраст між світлом і темрявою. Це дозволяє візуально спростити форму об'єкта, підкреслити його силует та зробити сцену більш виразною. Такий підхід активно використовується в анімації, відеоіграх та ілюстраціях, де важлива не фізична достовірність, а художня інтерпретація. Таким чином, Toon Shader відкриває широкі можливості для стилізації сцени та надає їй унікального графічного характеру.

3.3 Створення outline шейдеру

Створення outline (контурного) шейдера є важливим етапом стилізації сцени, особливо якщо ми прагнемо досягти візуального ефекту, подібного до ручної анімації або графічного роману. Контурний шейдер дозволяє обводити силуети об'єктів тонкими лініями, тим самим покращуючи візуальне сприйняття форми, глибини та меж між елементами сцени. Це не лише додає художнього стилю, але й значно покращує читабельність складних сцен, особливо в умовах зниженої деталізації або специфічного освітлення.

У парі з Toon Shader, який відповідає за різкі переходи між світлом і тінню, Outline Shader виконує функцію завершального штриха — підкреслює обриси та структуру моделі.

Щоб створити outline shader, нам так само не знадобиться багато нодів — він створюється відносно просто, але при цьому дозволяє досягти дуже ефектностилізованого результату. На зображенні вище зображено приклад такого шейдера, що використовує базовий матеріал у поєднанні з контуром, який формується на основі кута огляду (Fresnel ефекту) та логічного порівняння. На рисунку можна побачити приклад створення такого шейдеру в редакторі шейдерів(рис 3.4).

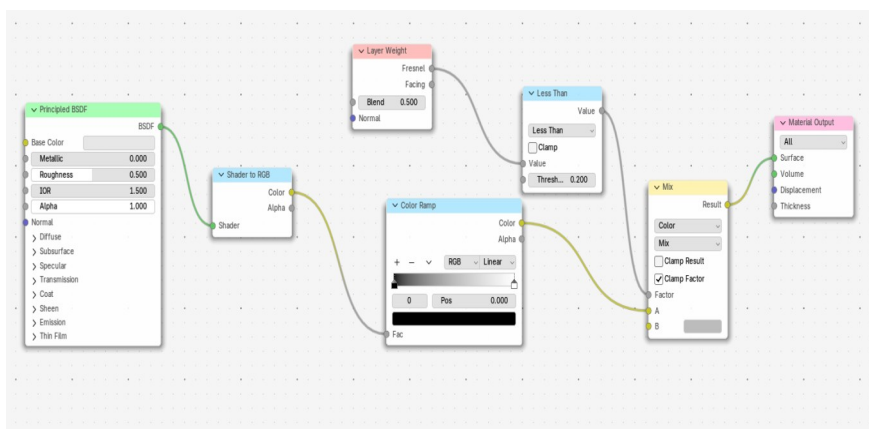


Рисунок 3.4 — Приклад створення outline шейдеру в редакторі шейдерів

Починаємо з вузла Principled BSDF, який формує базовий матеріал об'єкта. Він надсилається у вузол Shader to RGB, що переводить шейдер у колір, який ми можемо обробляти як звичайне зображення. Далі він подається у Color Ramp, де ми можемо задати палітру кольорів — у даному випадку вона використовується для основного забарвлення поверхні.

Паралельно формується сигнал для контурів: Layer Weight генерує значення Fresnel, яке залежить від кута огляду. Це значення подається в вузол Less Than, що дозволяє виділити ті області, які розташовані під певним кутом до камери (тобто — краї об'єктів). Отриманий результат змішується з основним матеріалом за допомогою вузла Mix, де другий колір (B) встановлений чорним — саме він і формує обвідку (outline) об'єкта. Таким чином, при певному куті огляду сцена автоматично додає до країв чорний контур, що і створює ефект outline. На рисунку 24 можна побачити як виглядатиме мавпяча голова з стандартним матеріалом та outline шейдером (рис 3.5).

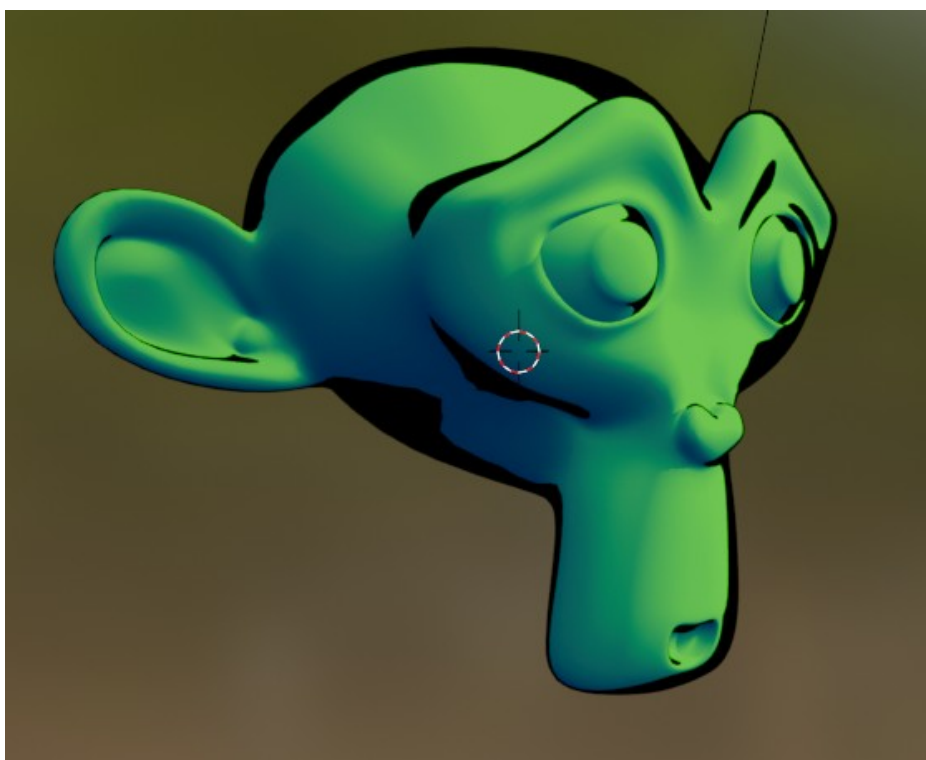


Рисунок 3.5 — Мавпяча голова зі стандартним матеріалом та outline шейдером

Тепер поєднаємо наш outline шейдер з toon шейдером і подивимось на результати(рис 3.6)



Рисунок 3.6 — Outline та toon шейдер застосовані до мавпячої голови

Було змінено кольорову палітру toon шейдера для більшої наглядності обводки outline шейдера. Це дозволило створити чіткіший контраст між основними кольорами об'єкта та його контурами, завдяки чому силуети стали значно виразнішими.

У поєднанні ці два шейдери працюють синергічно: toon створює спрощену, ілюстративну подачу кольору, а outline підкреслює форму та відокремлює об'єкти один від одного, роблячи композицію більш читабельною. Завдяки зміні палітри toon шейдера можна досягти кращого візуального балансу, де контури не зливаються з фоном або тінями, а чітко виділяють форми в сцені.

3.4 Створення dither шейдера

Щоб створити шейдер дизерингу, що імітує візерунок з точок або штрихів, нам так само не знадобиться багато нодів — він створюється відносно просто, але

при цьому дозволяє досягти дуже ефектного стилізованого результату, що нагадує друковану графіку або комікси[10].

Починається все з вузла Principled BSDF, який формує базовий матеріал об'єкта. Він надсилається у вузол Shader to RGB, що переводить шейдер у колір, який ми можемо обробляти як звичайне зображення. Далі він подається у перший Color Ramp, де ми можемо задати початковий діапазон яскравості, який був використаний для визначення щільності дизерингу.

Паралельно формується шаблонний сигнал для дизерингу: вузол Texture Coordinate (в режимі Generated) надає базові координати поверхні. Ці координати обробляються вузлами Mapping та Multiply для масштабування та контролю розміру бажаних "точок" дизерингу. Ключовим для створення повторюваного візерунка є вузол Modulo, який генерує "сітку" повторюваних значень, а Distance розраховує відстань від центру кожної такої "комірки", створюючи таким чином плями або кола, що повторюються по всій поверхні.

Отриманий шаблонний сигнал (Distance) додається до яскравості базового матеріалу (результату першого Color Ramp) за допомогою вузла Add. Це дозволяє "підняти" або "опустити" значення яскравості в залежності від шаблону. Результат цієї суми потім обробляється вузлом Multiply для додаткового контролю інтенсивності.

Нарешті, модифікований сигнал подається у другий Color Ramp. Цей вузол виступає як поріг: він перетворює плавні градієнтні значення на бінарний результат – чорні або білі "точки". Чим світліша ділянка поверхні (після додавання шаблону), тим вище ймовірність, що вона перетнувши поріг стане білою. І навпаки, чим темніша ділянка, тим менше білих "точок" відображено.

На рисунку 3.7 зображено приклад створення dither шейдеру в редакторі шейдерів.

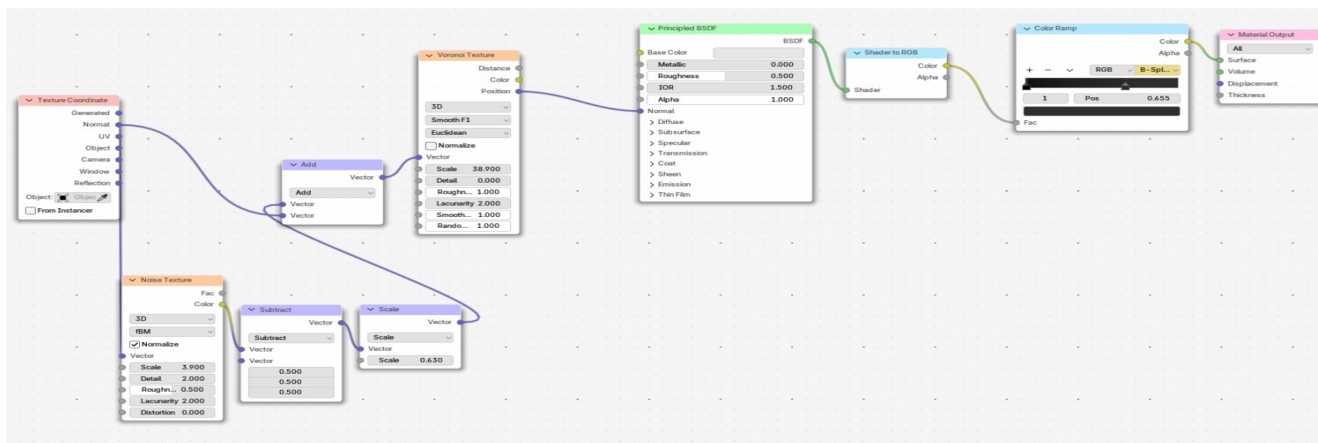


Рисунок 3.7 — Dither шейдер розроблений в редакторі шейдерів

Таким чином, у яскравих областях об'єкта ми бачимо високу щільність білих точок, тоді як у тіньових зонах їх кількість значно зменшується, створюючи візуальний ефект дизерингу або штрихування. Це чудово поєднується зі стилізованою графікою, дозволяючи досягти унікального візуального стилю, що імітує растрову графіку, газетний друк або комікси, додаючи особливої художньої виразності сцені(рис 3.8).

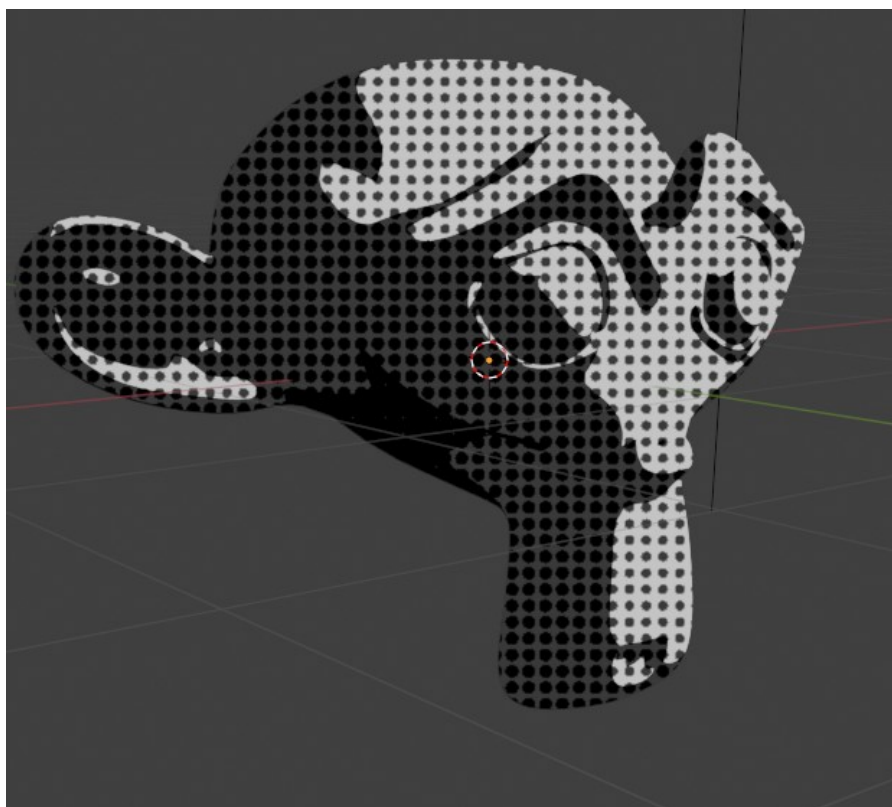


Рисунок 3.8 — Dither шейдер застосований до мавпячої голови

3.5 Розробка watercolor шейдеру

Watercolor шейдер — це ще один художній підхід до стилізації сцени, який імітує вигляд традиційного акварельного живопису. Такий ефект додає візуальній подачі м'якості, текстурованості та органічності, що може значно змінити сприйняття сцени, зробивши її більш емоційною, теплою або казковою[11].

Головною метою watercolor шейдера є створення відчуття ручної роботи — з нерівномірними заливками кольору, м'якими переходами та легким зернистим шумом, який нагадує текстуру паперу або акварельних плям. Цей ефект добре поєднується з іншими стилізованими шейдерами, як-от toon та outline, дозволяючи додати більш виразну художню інтерпретацію сцени.

На рисунку 3.9 можна побачити приклад створення watercolor шейдера в редакторі шейдерів.

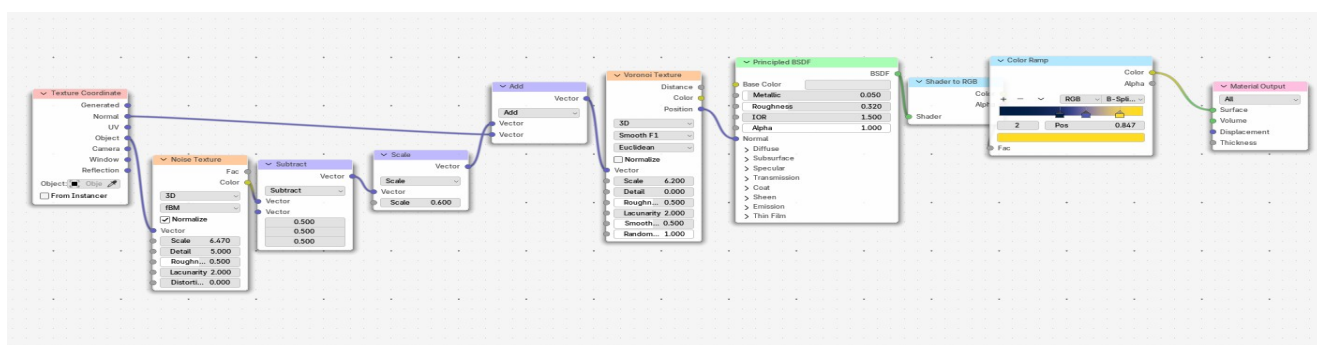


Рисунок 3.9 — Приклад watercolor шейдера в системі нодів

Починаємо все з вузла Principled BSDF, який формує базовий матеріал об'єкта. Він надсилається у вузол Shader to RGB, що переводить шейдер у колір, який ми можемо обробляти як звичайне зображення. Цей колір потім подається у Color Ramp, де ми можемо задати палітру кольорів або тональні переходи, які повинні виглядати як розведені пігменти акварелі. У даному випадку, палітра змінюється від синього до жовтого, що може створювати цікаві колірні переходи.

Вузол Texture Coordinate (в режимі Generated) надає базові координати поверхні. Ці координати подаються у Noise Texture (текстура шуму). Цей вузол

генерує випадкові, але плавні варіації, які є основою для імітації нерівностей та зернистості паперу або розводів фарби. Параметри Scale та Detail дозволяють контролювати розмір та складність цього шуму. Результат Noise Texture (його Color вихід) потім використовується як вхід для вузла Subtract, який віднімає константне значення від кожної компоненти кольору. Використовується для центрування значень навколо нуля, щоб подальші операції мали більший діапазон впливу. Далі результат Subtract подається у вузол Scale, що множить вектор на константу (0.600). Це додаткове масштабування, що впливає на інтенсивність або розмір ефекту шуму. Нарешті, цей модифікований сигнал (що пройшов через Noise Texture, Subtract, Scale) додається до вихідних координат об'єкта (Texture Coordinate) за допомогою вузла Add. Це створює "викривлення" або "спотворення" оригінальних координат, що є ключовим для імітації нерегулярних форм акварельних розводів. Отримані спотворені координати подаються у вузол Voronoi Texture. Текстура Вороного відома своєю здатністю створювати "клітинні" або "краплинні" візерунки, які можуть бути використані для імітації характерних плям та розводів акварельної фарби, де пігмент накопичується нерівномірно. Параметри Scale та Smoothness контролюють розмір та "м'якість" цих "крапель"(рис 3.10).

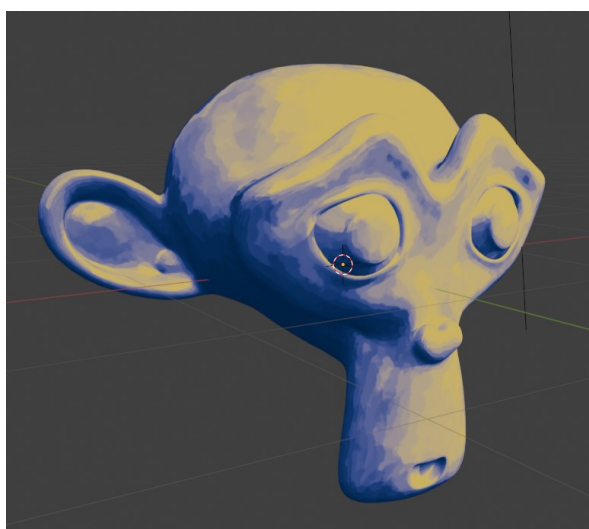


Рисунок 3.10 — Результат watercolor шейдера застосований на мавпячій голові

4 НАЛАШТУВАННЯ СЦЕН ТА ОПТИМІЗАЦІЯ

4.1 Створення матеріалів

Далі можна переходити до наступного ключового етапу розробки сцени — додавання матеріалів, налаштування освітлення та розміщення камери. На цьому етапі важливо не лише технічно оформити сцену, а й забезпечити її цілісність та атмосферу.

Матеріали відіграють вирішальну роль у створенні візуальної достовірності — саме вони визначають, як об'єкти взаємодіють зі світлом, відбивають його чи пропускають. Для кожного елемента сцени необхідно підібрати або створити відповідний шейдер. Наприклад, матрац, подушки й ковдра потребують м'яких, розсіяних матеріалів із невеликою кількістю блиску, а для ноутбука та окулярів потрібні поверхні з дзеркальними відблисками.

Освітлення у сцені не лише формує тіні й видимість, а й створює настрій. У даному випадку важливу роль відіграє контраст між м'яким навколишнім освітленням і точковими джерелами світла, такими як лампа на столі чи декоративне підсвічування. Необхідно ретельно підібрати інтенсивність та колір джерел світла, щоб підкреслити ключові об'єкти в композиції й надати сцені виразності. Особливу увагу було приділено використанню емісійних матеріалів для екрана ноутбука чи світлодіодних акцентів — вони додають сцені глибини й створюють цікаві зони контрасту.

Розміщення камери — ще один критично важливий аспект. Обрана композиція повинна не лише передавати зміст сцени, а й вести погляд глядача, виділяючи головні деталі й емоційні центри. Камера була розташована так, щоб охопити ліжку, тумбочку, ноутбук та інші ключові елементи, створюючи цілісну картину кімнати. При цьому важливо враховувати перспективу, фокусну відстань і можливість використання глибини різкості (Depth of Field) для акцентування уваги. Перш за все треба необхідно налаштувати матеріал світу(рис 4.1).

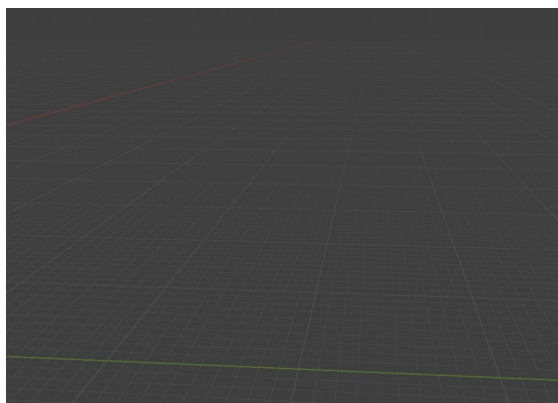


Рисунок 4.1 — Стандартний матеріал світу

Стандартний колір світу (world color) у Blender за замовчуванням має світло-сірий або майже білий відтінок, який випромінює розсіяне освітлення по всій сцені. Це може бути корисно на етапі базового моделювання для кращої видимості об'єктів, однак у фінальному рендері такий колір часто спотворює атмосферу сцени. У нашому випадку сцена має передавати відчуття затишку в нічний час, тому надмірна яскравість середовища створює враження штучної освітленості, яка руйнує задуману естетику.

Щоб виправити це, ми змінюємо базовий колір середовища на темно-синій або глибокий синьо-фіолетовий відтінок. Такий вибір не лише зменшує небажане глобальне освітлення, але й створює загальний фон, який нагадує вечірній або нічний настрій(рис 4.2).

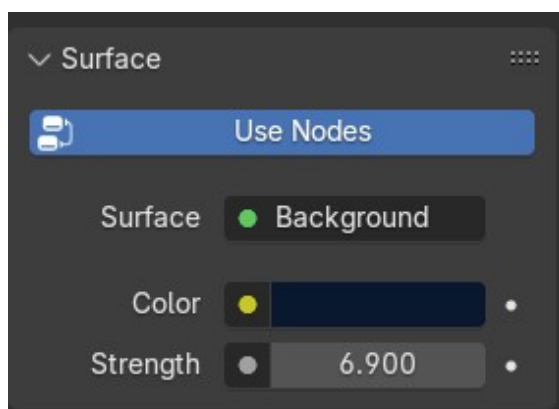


Рисунок 4.2 — Змінені налаштування світу

Ми зробили силу освітлення більшою щоб допомогти з непрямим освітленням сцени(бліки, тіні і тд). Також це допомогло забезпечити видимість об'єктів в умовах поганого освітлення.

Після завершення моделювання та розстановки освітлення настає один із найважливіших етапів — створення матеріалів для об'єктів. Саме матеріали визначають те, як поверхня виглядатиме у фінальному рендері: її колір, ступінь глянцею, прозорість, наявність віддзеркалення, м'якість або жорсткість світла, що від неї відбивається.

Насамперед усім об'єктам задається базовий колір (Base Color), який відповідає їхній фактурі та ролі у сцені — для дерев'яних поверхонь використовуються теплі відтінки коричневого, для текстилю — м'які та приглушені тони, а для техніки — прохолодні відтінки сірого або чорного. Далі налаштовуються такі властивості, як roughness (шорсткість) і metallic (металевість). Наприклад, для м'яких об'єктів, як-от ковдра чи подушки, параметр roughness встановлюється на високому рівні — близько 0.8–1.0, щоб надати поверхні матовий вигляд без відблисків. Для глянцевих об'єктів, таких як ноутбук або окуляри, roughness зменшується до значень у діапазоні 0.2–0.4, що забезпечує чітке віддзеркалення світла(рис 4.3).

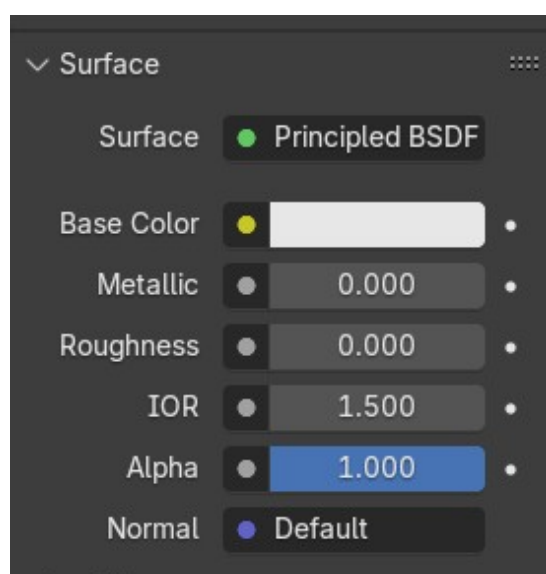


Рисунок 4.3 — Приклад налаштування металевості і шорсткості матеріалу

Особливу увагу приділи прозорим та скляним елементам, наприклад, декоративним пляшкам або флаконам, які знаходяться на тумбочці. Основні параметри, що налаштовуються — це IOR (Index of Refraction) та roughness. Значення IOR визначає, як світло заломлюється в матеріалі. Для звичайного скла воно встановлюється приблизно на рівні 1.45–1.52, що є фізично коректним значенням. Roughness для скла зазвичай залишається низьким — 0.0–0.1, якщо ми хочемо бачити чіткі прозорі форми. Але для матового або трохи запотілого скла можна підняти його до 0.2–0.3, щоб надати розсіяного ефекту(рис 4.4).

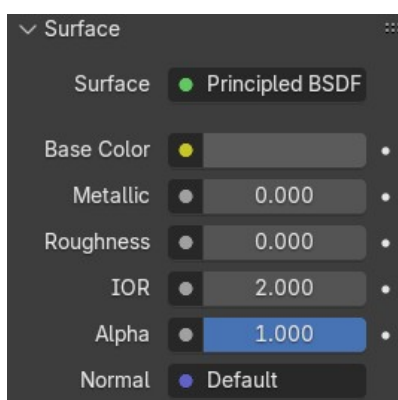


Рисунок 4.4 — Налаштування вікна у сцені

Після того як усі об'єкти в сцені отримали відповідні матеріали та були налаштовані їхні фізичні властивості — зокрема колір, шорсткість, прозорість, металевість та інші параметри — ми переходимо до етапу попереднього перегляду сцени з активним освітленням. На цьому етапі важливо оцінити загальне візуальне сприйняття сцени за умов стандартного освітлення — без додаткових джерел світла або ефектів. Такий рендер допомагає визначити, чи всі матеріали поведуться правильно, чи відповідають очікуваному вигляду, чи не виникає небажаних відблисків або помилок у прозорих елементах. Це своєрідна точка відліку, яка дозволяє скоригувати параметри перед переходом до фінального освітлення сцени(рис 4.5).

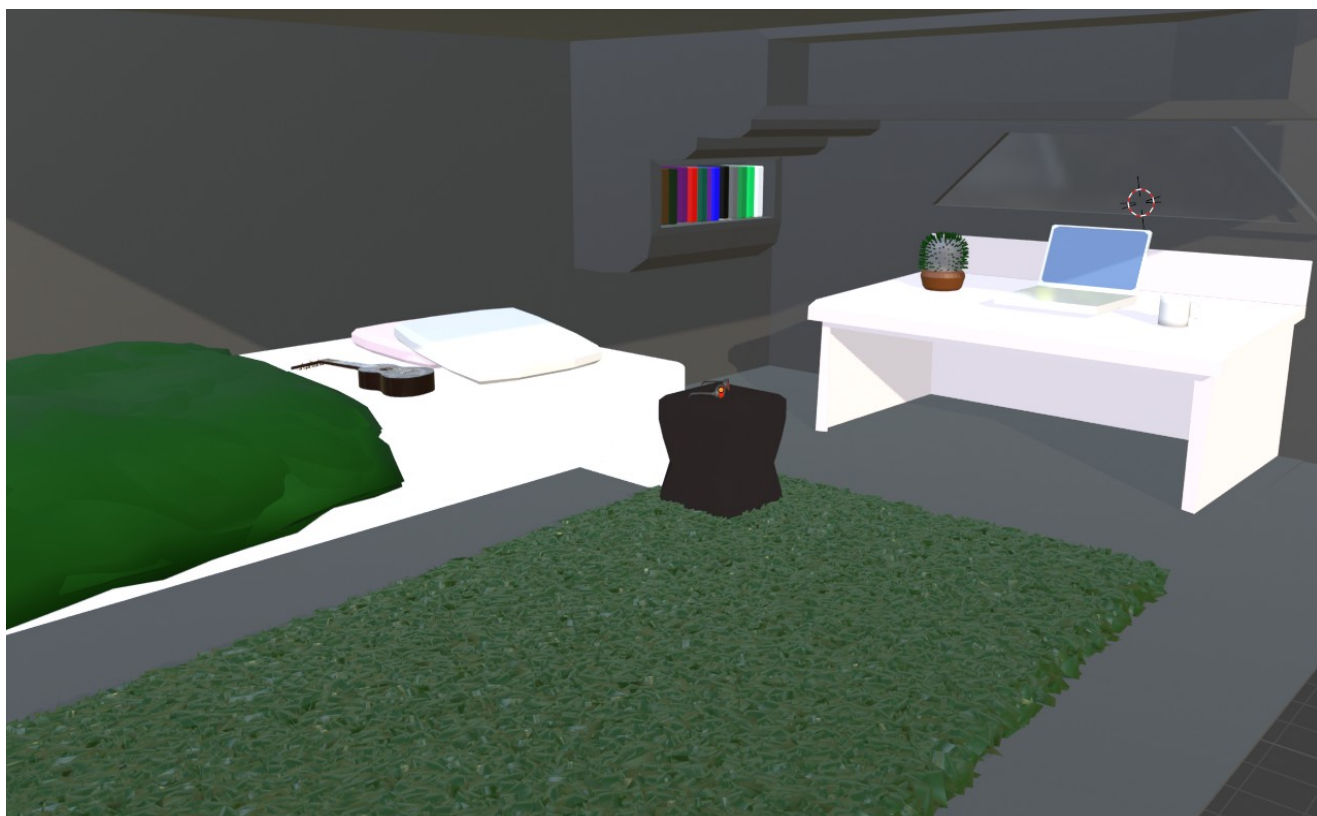


Рисунок 4.5 — Сцена з стандартними налаштуваннями освітлення

З цього результату можемо побачити, що з освітленням у сцені є явні проблеми: об'єкти виглядають плоско, засвічено та дуже примітивно. Освітлення не створює глибини, немає правильного контрасту між тінями і світлом, а загальна атмосфера не відповідає бажаному нічному настрою. Такий результат є типовим для сцени з увімкненим стандартним світлом світу, яке за замовчуванням має світло-сірий або білуватий відтінок. Щоб покращити ситуацію та наблизити сцену до бажаного нічного вигляду, ми змінили налаштування освітлення світу, використавши темніший відтінок синього кольору. Це дозволило приглушити загальну яскравість сцени, створити більш виразні тіні та надати всьому зображенню характерної холодної нічної атмосфери. Завдяки цьому матеріали краще розкриваються у світлі, об'єкти виглядають більш об'ємно, а вся сцена — глибшою та атмосферною. На рисунку 4.6 продемонстровано результат сцени з оновленими налаштуваннями нічного освітлення.

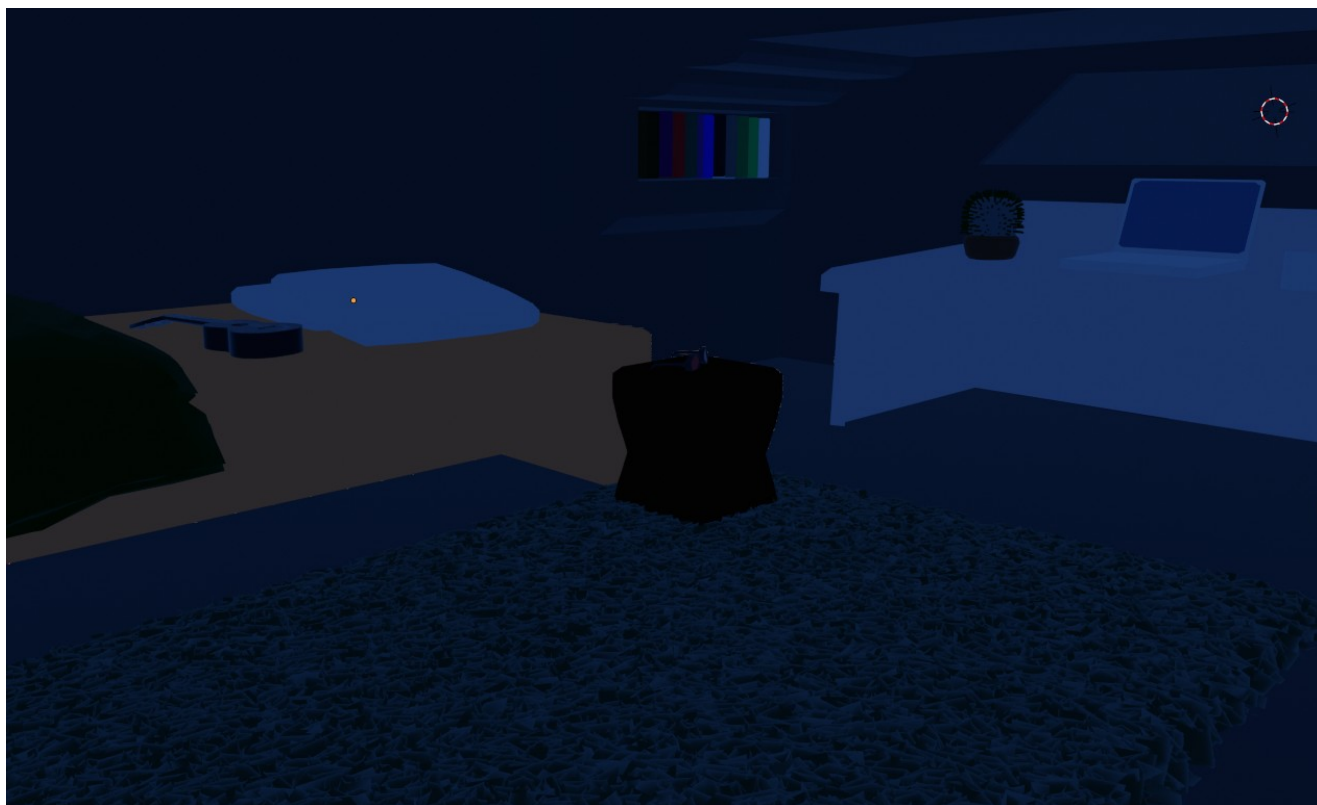


Рисунок 4.6 - Результат сцени з оновленими налаштуваннями нічного освітлення

Це вже виглядає набагато ближче до бажаного нічного настрою, однак на цьому етапі сцена має очевидну проблему — повну відсутність активного джерела світла. Без нього об'єкти в кадрі залишаються затемненими або погано видимими, що ускладнює сприйняття сцени та порушує візуальний баланс. Для вирішення цієї проблеми необхідно додати нові джерела світла, які не лише освітлюватимуть простір, але й підкреслюватимуть важливі об'єкти та формуватимуть композиційний центр.

У Blender це можливо зробити двома основними способами. Перший — це пряме додавання джерел світла до сцени, які почали працювати як незалежні об'єкти, що освітлюють інші елементи. Другий — це використання властивості Emission у матеріалах: при цьому самі об'єкти стають джерелами світла, випромінюючи його на оточення. Цей підхід особливо добре працює для створення декоративного або локального освітлення — як-от світильники, монітори, LED-підсвітка тощо.

Серед стандартних типів джерел світла в Blender можна виділити наступні:

- point;
- sun;
- spot;
- area;
- hemi.

Ці інструменти дозволяють досягти різноманітних ефектів залежно від типу сцени, її масштабу та стилістики. У нашій сцені було використано як звичайні джерела світла, так і емісійні матеріали для локального підсвічування окремих об'єктів[12].

Після додавання емісійних матеріалів до деяких ключових об'єктів сцени — таких як світильник-рослина, екран ноутбука та окуляри — ми помічаємо суттєве покращення в освітленні. Ці об'єкти починають самостійно випромінювати світло, створюючи м'які світлові плями та підкреслюючи форму навколишніх предметів. Щоб оцінити ці зміни, можемо перейти до перегляду оновленого рендеру сцени(рис4.7).

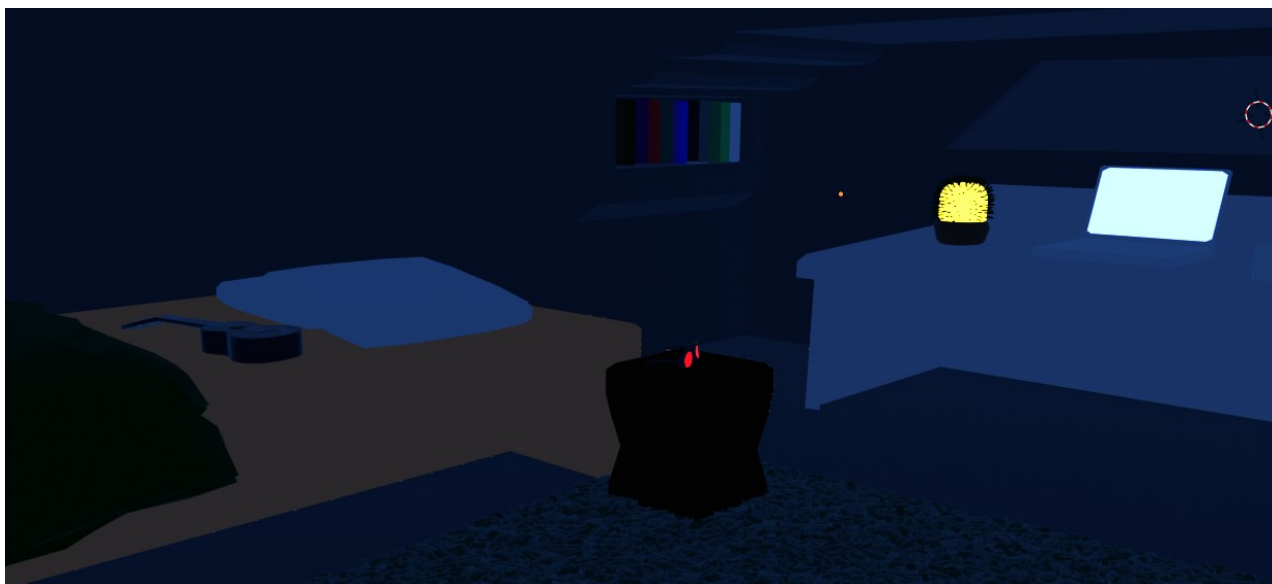


Рисунок 4.7— оновлена сцена з емісіями

Також ми додаємо джерело світла типу Area Light, розташоване біля ноутбука, щоб краще висвітлити ліжко та змодельовати ефект непрямого освітлення, яке могло б виникати від екрану. Це допомагає створити м'яке заповнююче світло, яке посилює глибину сцени та робить її більш природною для сприйняття (рис 4.8).

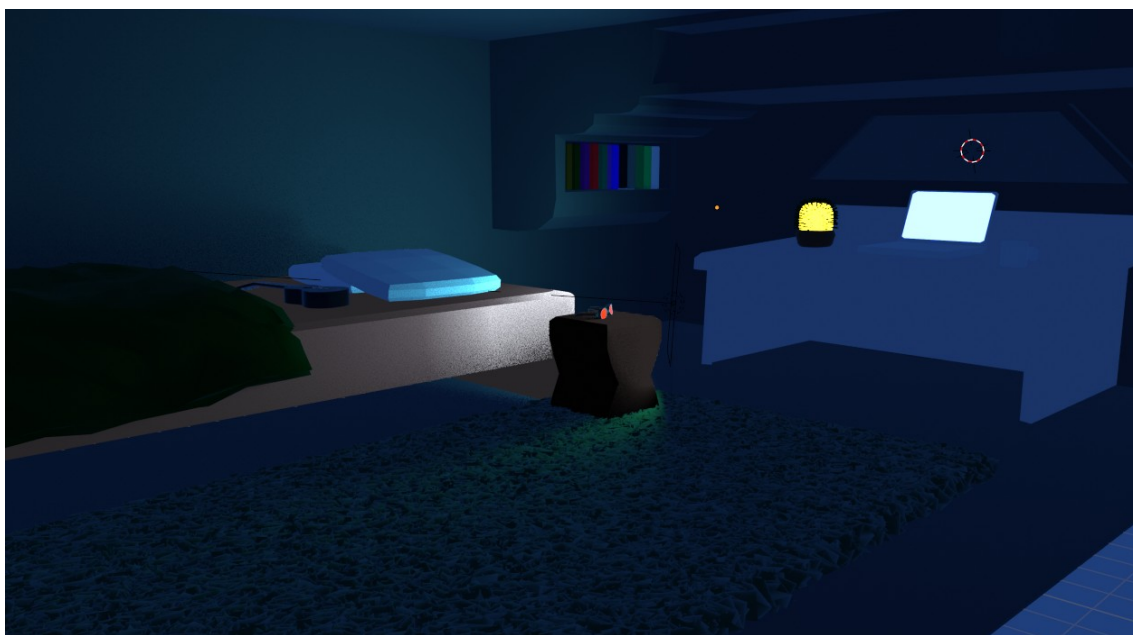


Рисунок 4.8 — сцена з доданим area light

Також для вирішення проблеми пласкості об'єктів ми додаємо до сцени непряме освітлення. Зокрема, об'єкти з емісійними матеріалами, такі як ноутбук чи світильник, починають виступати як джерела світла, і їх вплив на навколишнє середовище також має бути прорахований. Щоб забезпечити це, ми використовуємо спеціальний інструмент у Blender — Light Probe Volume.

Light Probe Volume — це система зондових елементів, які автоматично зчитують та зберігають інформацію про непряме освітлення у тривимірному просторі сцени. Вони дозволяють програмі симулювати, як світло, що випускається емісійними об'єктами або відбивається від інших поверхонь, поширюється в оточенні. На рисунку 4.9 можна побачити результати використання light probe volume.



Рисунок 4.9— Результати додавання light probe volume

Також, методом покращення освітлення в нашій сцені може стати рейтрейсінг. Рейтрейсінг (ray tracing) — це сучасна техніка освітлення сцен у 3D-графіці, яка імітує фізичні властивості світла шляхом трасування променів від камери через пікселі зображення в сцену. Ці промені перевіряють, з якими поверхнями вони стикаються, і як світло взаємодіє з цими поверхнями: відбивається, заломлюється або поглинається.

Рейтрейсінг є вкрай ресурсомістким методом. Для кожного пікселя зображення може бути необхідно обчислювати десятки або навіть сотні променів, враховуючи всі їхні відбивання, заломлення та взаємодії з матеріалами. У реальному часі це призводить до значного навантаження на графічний процесор, що суттєво знижує продуктивність, особливо у складних сценах з великою кількістю світлових джерел або прозорих матеріалів.

Незважаючи на це, результати, які дає рейтрейсінг, переважають інші методи, якщо головним завданням є досягнення фотореалізму. Це робить його особливо цінним у візуалізації, рендерингу фільмів, архітектурних проєктах та сценах, де якість зображення має вирішальне значення (рис 4.10). У сучасних рушіях, таких як Blender (Cycles) або Unreal Engine, рейтрейсінг став доступним у

реальному часі завдяки апаратному прискоренню (наприклад, через RTX), хоча й залишається вибором для потужніших систем або фінального рендеру.

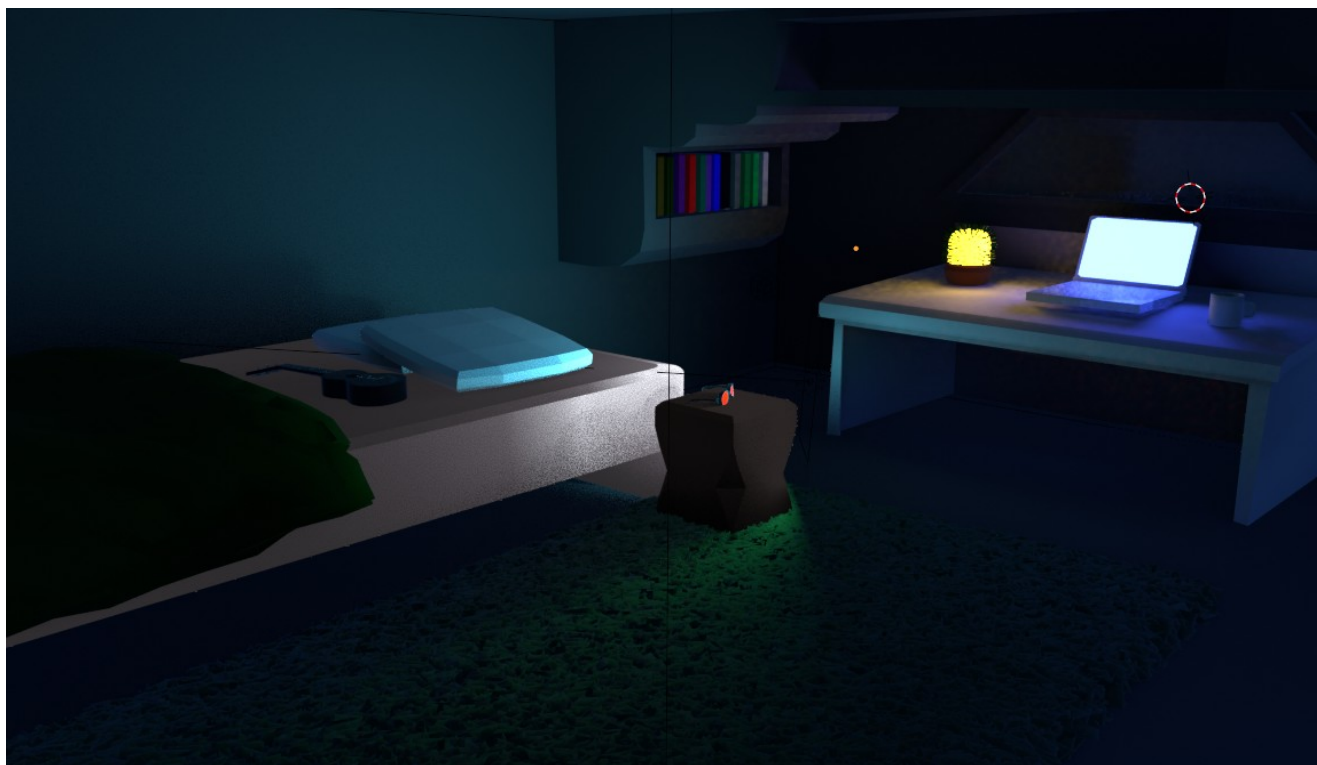


Рисунок 4.10 — Сцена з використанням рейтрейсінгу

Також варто відмітити що велика робота була проведена для того щоб налаштувати камеру сцени. Тип об'єктива встановлений на Perspective, що забезпечує реалістичну передачу простору з ефектом глибини. Фокусна відстань обрана як 29 мм, що дозволяє отримати широкий кут огляду, підходящий для зйомки інтер'єрів, де важливо охопити багато деталей у кадрі. Shift X = -0.050 та Y = 0.060 зміщують камеру відносно центру кадру, щоб краще скомпонувати зображення та підкреслити потрібні об'єкти у сцені.

Крім того, увімкнено глибину різкості (Depth of Field), що додає сцені реалістичності, розмиваючи об'єкти, які знаходяться поза зоною фокусування. Камера сфокусована на конкретний об'єкт, а фокусна відстань виставлена на 32.55 м, що дозволяє досягти чіткої глибини поля з поступовим розмиванням заднього та переднього плану. Це підсилює ефект присутності глядача в сцені та акцентує увагу на ключових елементах.

Тепер, коли ми досягли задовільного рівня освітлення сцени, можемо перейти до етапу стилізації за допомогою шейдерів. Це дозволить не лише покращити візуальне сприйняття, а й надати сцені унікального художнього стилю. У нашому проєкті ми використали кілька спеціально розроблених шейдерів:

- toon;
- outline;
- dither;
- colorpaint.

Toon-шейдер надає об'єктам графічного вигляду з обмеженою кількістю відтінків тіней, що нагадує ілюстрацію або мультфільм. Dither-шейдер додає сцені ефект "зернистості" чи точкового переходу між кольорами, що пом'якшує границі тіней. Outline-шейдер створює контури навколо об'єктів, допомагаючи краще їх виділяти на фоні[13].

4.2 Імплементация шейдерів в indoor-lit сцену

Тепер, коли ми маємо в нашому розпорядженні набір розроблених шейдерів, можемо перейти до етапу стилізації сцени. Серед реалізованих нами шейдерів є такі:

- toon shader;
- dither shader;
- putline shader;
- watercolor shader.

Однак при стилізації сцени важливо розуміти, що ми не можемо просто взяти й замінити всі матеріали на стилізовані версії — це призвело б до втрати глибини, різноманіття й реалістичності візуального сприйняття. Стилiзація у 3D не повинна зводитися до одноманітного покриття сцени одним шейдером — навпаки, ефект досягається через баланс між стилізованими та фізично-

коректними матеріалами. Ми обираємо використовувати стилізовані шейдери переважно для акцентних деталей, об'єктів, що взаємодіють зі світлом, або тих, які привертають увагу глядача (наприклад, декор, освітлені зони, предмети з важливим візуальним навантаженням).

Завдяки цьому підходу ми зберігаємо візуальну читабельність сцени, її глибину та об'єм, одночасно додаючи стилістичну виразність. Шейдери стають не універсальним інструментом, а засобом художнього виділення, який допомагає керувати увагою користувача та формує емоційний настрій сцени. Тож почнемо з застосування toon шейдеру на нашу indoor сцену. На рисунку 4.11 можна побачити сцену з усіма стандартними матеріалами а на 4.12 з нашими стилізованими шейдерами вже застосованими до необхідних об'єктів.

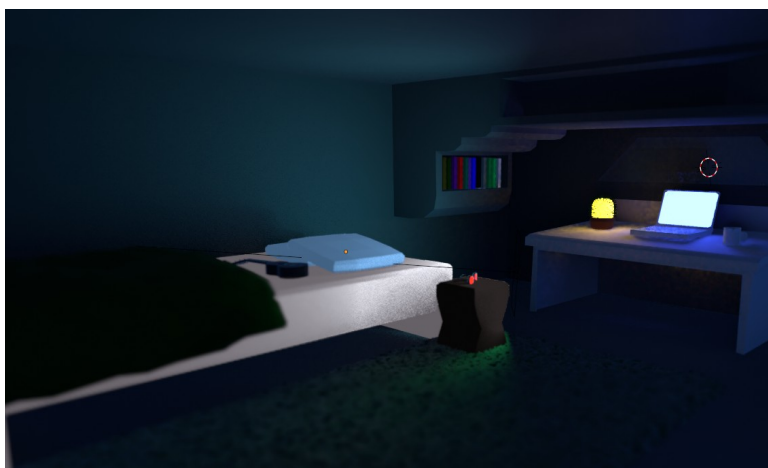


Рисунок 4.11 — Сцена зі звичайними матеріалами

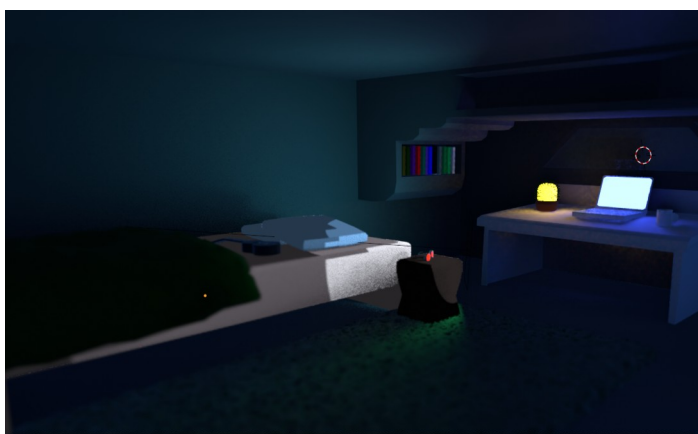


Рисунок 4.12— Сцена з застосуванням toon шейдеру на ключові об'єкти

На рисунку добре видно характерні особливості Тооп шейдеру, які ми вже розглядали раніше. Різкий перехід між освітленими та затіненими ділянками, що створює ефект стилізованого затінення, схожого на зображення у мультфільмах чи коміксах. На відміну від звичайних матеріалів, які плавно передають світло й тінь, Тооп шейдер ділить освітлення на чітко визначені зони з фіксованими тонами — світлими, середніми та темними.

Цей підхід дає кілька переваг: по-перше, візуальна простота дозволяє глядачеві швидше зчитувати форму об'єктів, не відволікаючись на зайві деталі. По-друге, художній контроль над стилем дозволяє задати настрій сцени і створити унікальну візуальну ідентичність. По-третє, виразність контрастів підвищує увагу до важливих елементів сцени. Таким чином, Тооп шейдер не лише стилізує зображення, а й допомагає краще керувати композицією і сприйняттям сцени.

На рисунку 4.13 можна побачити сцену з dither шейдером.

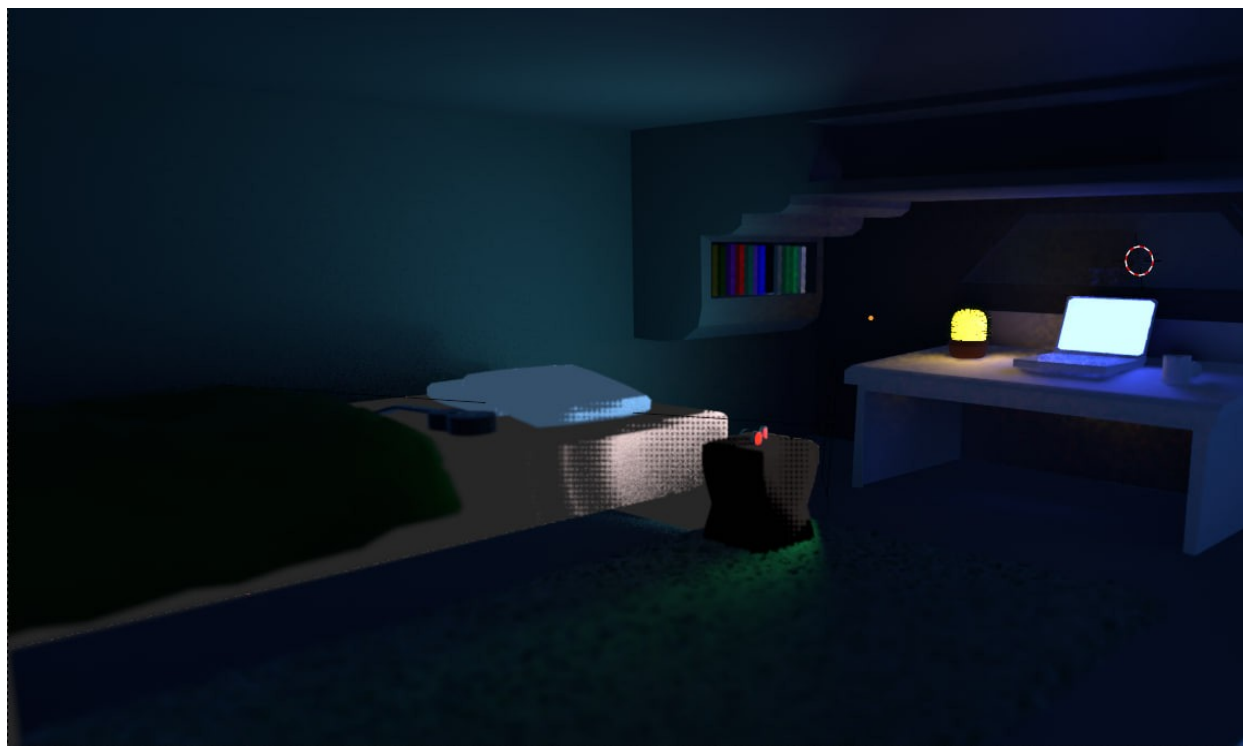


Рисунок 4.13 — Сцена з dither шейдером

Dither шейдер замінює плавні переходи на графічний ефект, у якому різні рівні освітленості імітуються за допомогою регулярного або псевдовипадкового візерунка з пікселів. Світліші й темніші зони виглядають як «зерно» або сітка з крапок, що створює відчуття стилізованої, вручну промальованої сцени. Це дозволяє добитися унікального художнього вигляду, який неможливо отримати за допомогою звичайних матеріалів.

Тепер ми замінімо матеріали ключових об'єктів у сцені на watercolor шейдер, щоб досягти м'якої, живописної стилізації. Такий шейдер дозволяє надати сцені більш художнього, емоційного вигляду, наблизити її до стилю ручного малюнка й підсилити атмосферу затишку та казковості. Завдяки цим особливостям поверхні об'єктів виглядають менш реалістично, проте значно виразніше й стилізовано. На рисунку 4.14 видно сцену з watercolor шейдером.

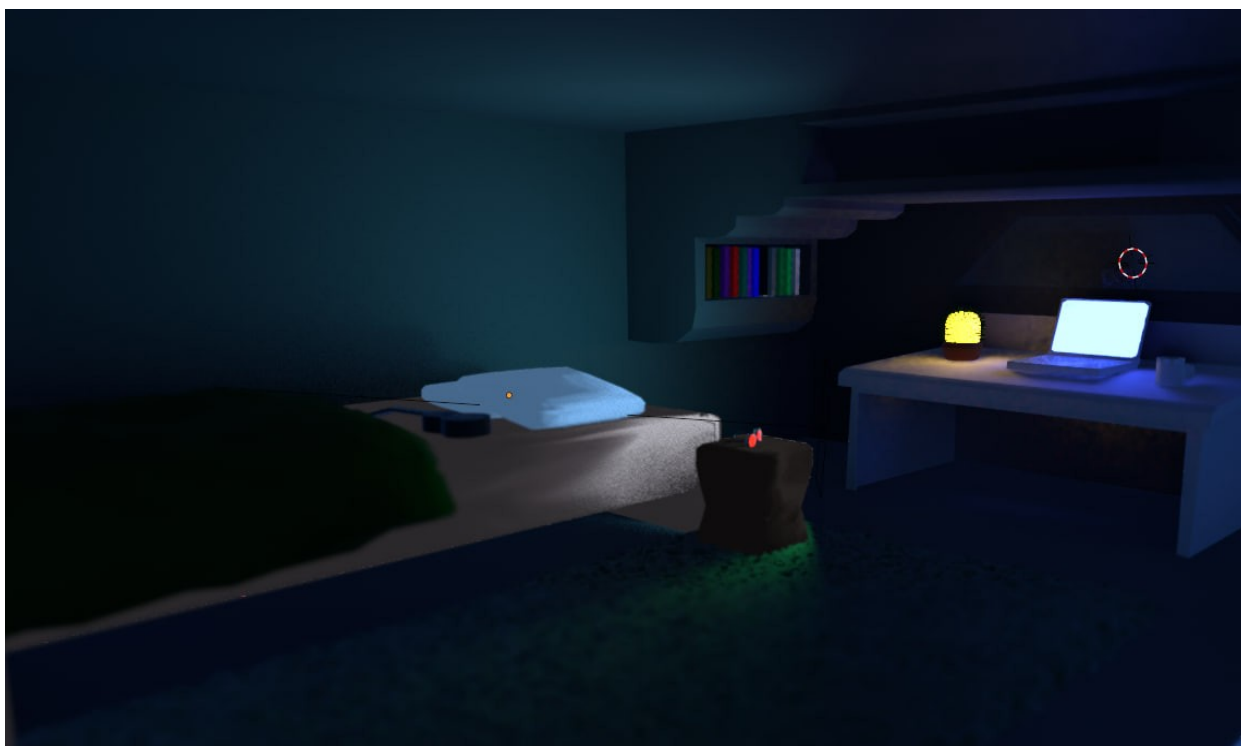


Рисунок 4.14 — Сцена з watercolor шейдером

У підсумку можна сказати, що була пророблена комплексна й багатоступенева робота зі створення детально опрацьованої сцени інтер'єру (indoor-локації), яка поєднує в собі як технічну точність, так і художню

виразність. Основною метою стало створення сцени, яка виглядала б не лише правдоподібно з технічної точки зору, але й була б стилістично цікавою, виразною та емоційно насиченою. Робота над сценою почалася з побудови базових об'єктів: від великих структур — підлоги, стін, меблів — до дрібних декоративних елементів, таких як чашка, рослина, окуляри, укулеле та інші. Для моделювання складних форм, наприклад, вигнутих контурів або м'яких органічних ліній, використовувалися криві (curve objects), що значно спростило процес створення плавних, природних форм. У деяких випадках додатково застосовувався модифікатор subdivision surface для покращення деталізації, особливо там, де це було важливо для наближення об'єкта до реалістичного вигляду. Одним із важливих етапів стало опрацювання килима — складного об'єкта з точки зору геометрії. Для його створення використовувалася система волосся (hair system), що дозволила імітувати товсті ворсини при мінімальному навантаженні на продуктивність. Було обрано відповідні параметри довжини, густоти, товщини та варіацій нахилу ворсинок, щоб досягти максимально реалістичного й виразного вигляду поверхні. Наступним етапом стало налаштування освітлення сцени. Стандартне освітлення в Blender, яке йде за замовчуванням, було надто яскравим, через що сцена виглядала занадто пласкою й засвіченою. Ми вирішили змінити параметри освітлення світу, зокрема змінити колір оточення на темно-синій, що дозволило відтворити нічну атмосферу. Після цього в сцену було додано джерела світла: емісійні матеріали (emission shaders) для таких об'єктів, як екран ноутбука, лампа-рослина, окуляри, а також стандартні джерела світла, такі як area light. Щоб симулювати непряме освітлення й зробити сцену глибшою, були застосовані інструменти типу light probe volume, а також активовано рейтрейсінг (ray tracing) для покращення тіней, відбиттів та загального сприйняття простору. Попри те, що рейтрейсінг значно підвищує навантаження на систему, він дозволяє досягти бажаного рівня освітленості локації. Після фінального налаштування освітлення сцена вже виглядала достатньо переконливо, проте для досягнення стилізованого, художнього ефекту була

розпочата робота з розробки власних шейдерів. Усього було створено декілька ключових шейдерів, кожен з яких має унікальні візуальні властивості та мету:

- toon shader ;
- outline shader;
- dither shader;
- watercolor shader.

Кожен із цих шейдерів був протестований у сцені й налаштований відповідно до її естетичних вимог. Toon та outline шейдери взаємодіють разом, дозволяючи створити контрастну стилізацію з чіткими межами, а watercolor shader був використаний для ключових об'єктів, щоб створити відчуття тепла, затишку та ручної роботи. Dither шейдер дозволив ввести додаткову текстурну виразність без значних змін геометрії. Стилізація сцени відбувалася поступово: було проаналізовано кожен об'єкт на предмет важливості для загального враження та прийнято рішення, чи варто застосовувати до нього стилізований матеріал, чи залишити стандартний, реалістичний. Такий підхід забезпечив візуальний баланс — сцена не виглядає перевантаженою стилізацією, натомість зберігає гармонію між художністю та технічною достовірністю. Зрештою, ми отримали сцену, яка виглядає не лише як достовірна інтер'єрна візуалізація, а й як повноцінне художнє полотно. Завдяки гнучкості Blender, можливостям Shader Editor та детальному налаштуванню матеріалів і освітлення, нам вдалося досягти результату, що поєднує глибину, стиль і виразність. Створені шейдери стали не просто технічними елементами, а повноцінними інструментами художнього впливу.

4.3 Стилізація outdoor-lit сцени

Після успішного створення outdoor сцени та побудови системи трави з динамічним коливанням, ми можемо перейти до наступного ключового етапу — стилізації. Цей розділ присвячений художньому опрацюванню нашої сцени, а

саме — створенню виразного візуального стилю для трави за допомогою написаних раніше шейдерів.

Візуальна стилізація є не просто декоративним доповненням, а важливою частиною побудови нарративу та атмосфери сцени. Тепер ми застосуємо їх до нашої outdoor сцени, протестуємо, як вони взаємодіють з геометрією трави, та порівняємо результати.

Метою цього розділу є не просто "перекрасити" сцену, а створити художньо узгоджене середовище, де стилізація допомагає підкреслити ключові елементи, покращує читабельність композиції та додає унікальності візуальному стилю. У кожному з прикладів ми покажемо, як зміна лише шейдера змінює настрої та враження від сцени, і продемонструємо потенціал Blender як інструменту для гнучкої та багаторівневої стилізації 3D-середовища(4.15).

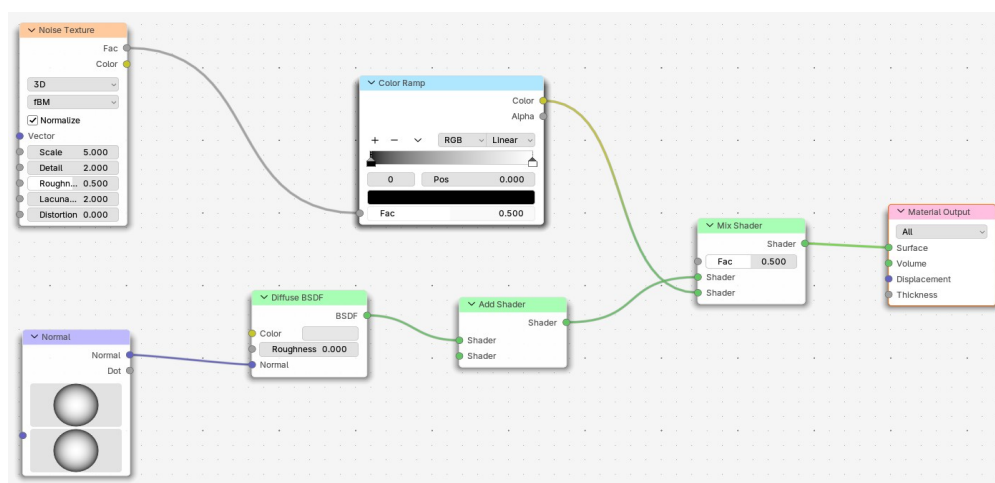


Рисунок 4.15— Основна логіка для налаштування toon шейдеру для нашої системи трави

Основна ідея полягає в тому, щоб замінити плавне освітлення на різкі межі між світлом і тінню, чим часто користуються у мультфільмах або коміксах. Все починається з вузла Normal, який визначає напрямок кожної точки поверхні. Далі ми використовуємо скалярний добуток (Dot Product), щоб порівняти ці напрямки з напрямком світла — це дає значення освітленості. Потім застосовується Diffuse BSDF, який розсіює світло по поверхні, але результат не виводиться напряму.

Замість цього ми використовуємо Shader to RGB, щоб перетворити його на колірні дані. Далі Color Ramp допомагає зробити освітлення стилізованим — завдяки різким переходам між відтінками. На фінальному етапі ми можемо змішати цей результат з іншим кольором або ефектом через Mix, і передати готовий вигляд на Material Output. У підсумку — отримуємо чистий, контрольований toon-ефект[14].

Після цього ми змішуємо вихід з Color ramp з через вузол shader mix та налаштовуємо систему за якою при значеннях шумової текстури на траві проявитись більше або менше конкретних відтінків кольору[15](рис4.16).

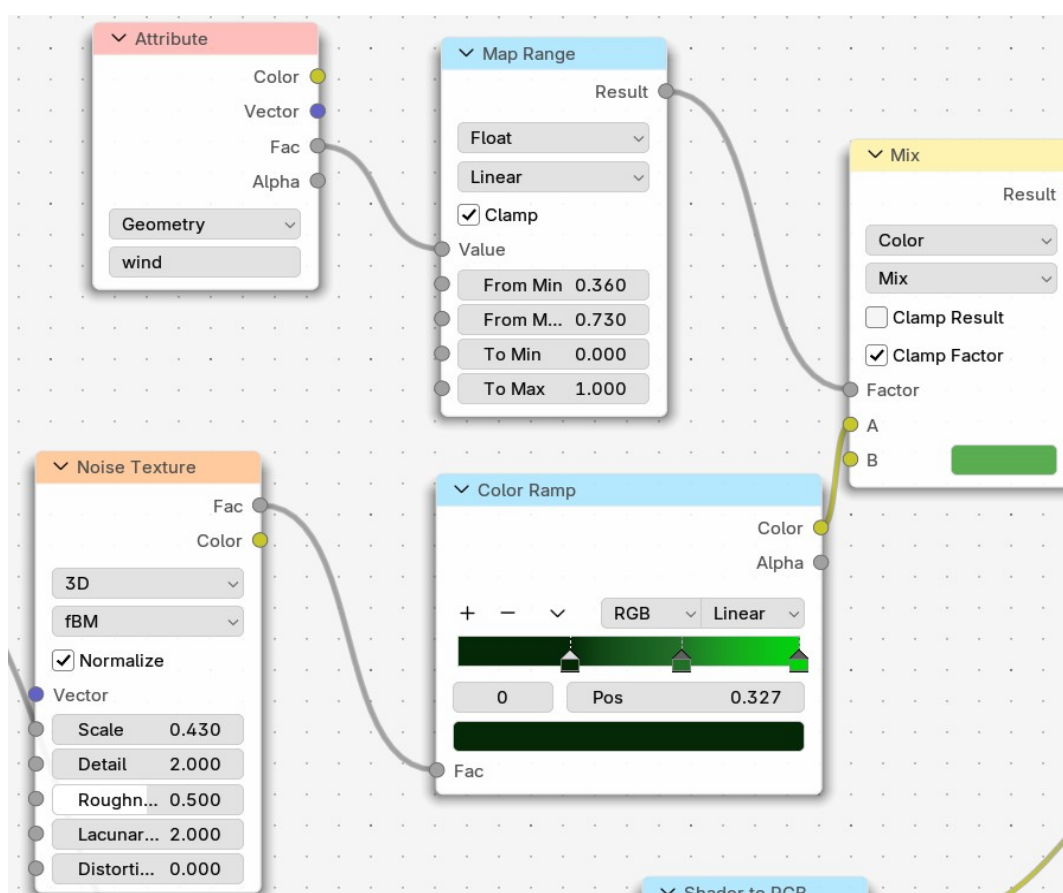


Рисунок 4.16 — Більш витончена система налаштування кольору.

Те, що ми отримали в результаті всіх попередніх вузлів — це колір із варіативною яскравістю для кожної окремої травинки. Однак, щоб ще більше підсилити враження глибини та природності, ми змішуємо цей результат із ще однією системою, яка додає прозорість. Цей ефект прозорості дозволяє краще

передати візуальну легкість трави — коли травинки частково накладаються одна на одну, створюючи відчуття об'єму без потреби у великій кількості геометрії[16].

Додавання прозорості також покращує ілюзію м'якого, плавного переходу між окремими травинками, ніби вони зливаються між собою в єдиний покрив. Завдяки цьому сцена сприймається більш суцільною, а не як набір ізольованих об'єктів. Це особливо важливо для стилізованих сцен, де ми не намагаємось досягти фотореалізму, а хочемо створити емоційне, художнє враження. Застосування прозорості на цьому етапі дає змогу зберегти легкість зображення й уникнути візуального "перевантаження", водночас залишаючи текстури живими та виразними(рис 4.17).

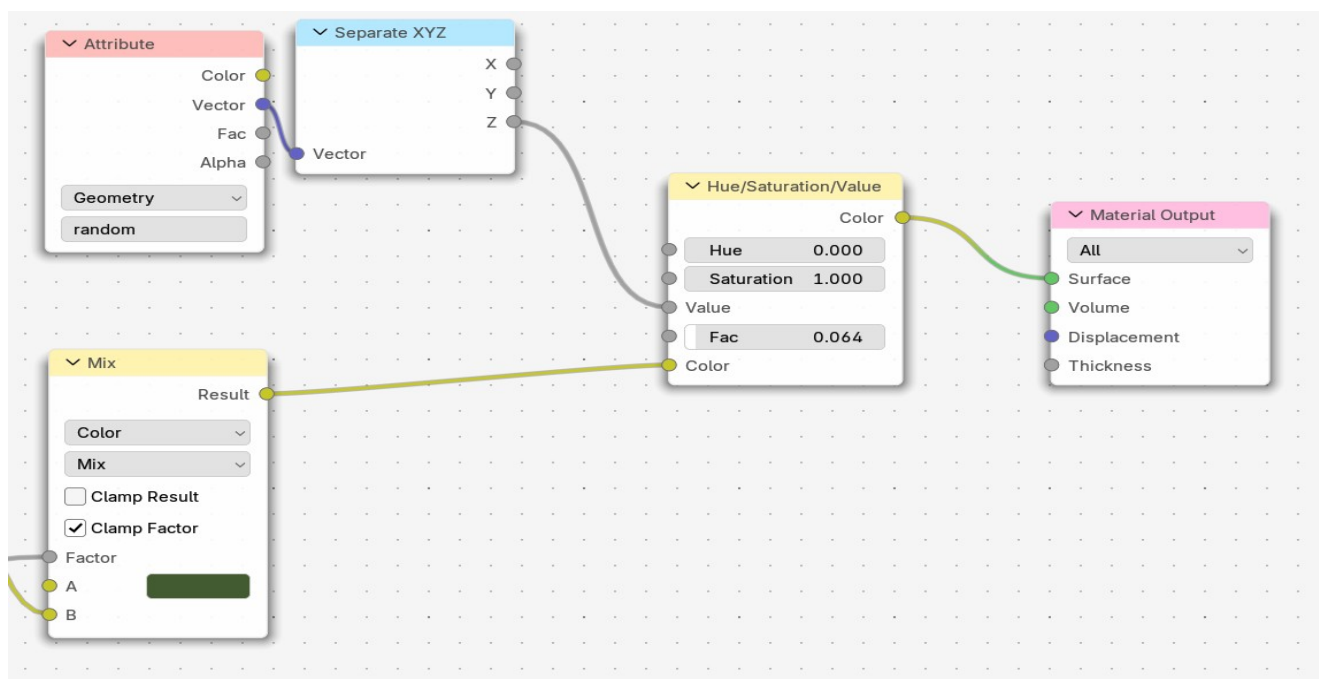


Рисунок 4.17 — Приклад системи що забезпечує ефект вітру в трави

Ми реалізували коливання травинок за допомогою шуму та часової анімації, завдяки чому отримали м'який рух, що імітує вплив вітру. Кожна травинка отримала свою унікальну позицію, нахил і навіть колір, що зробило покрив різноманітним і природним. Окрім геометрії та руху, також додали колірну варіацію, керовану випадковим фактором,(рис 4.18).

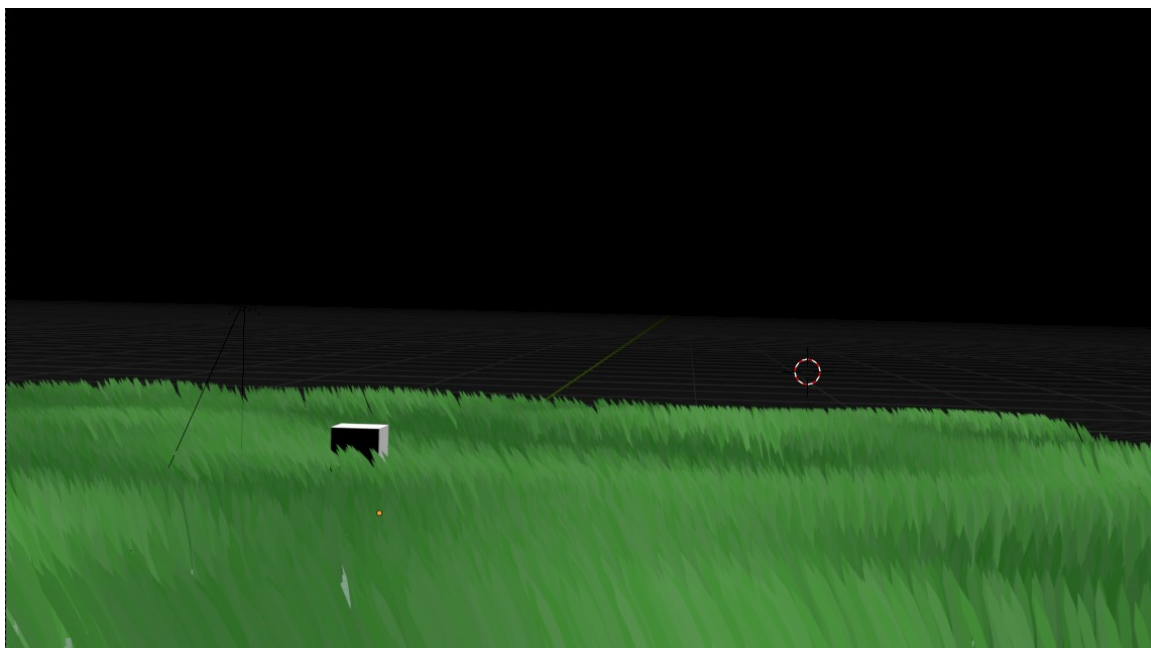


Рисунок 4.18 — Стилїзована outdoor-lit сцена

Так само перевіряємо роботу шумової текстури порівнюючи два кадри між собою(4.19).

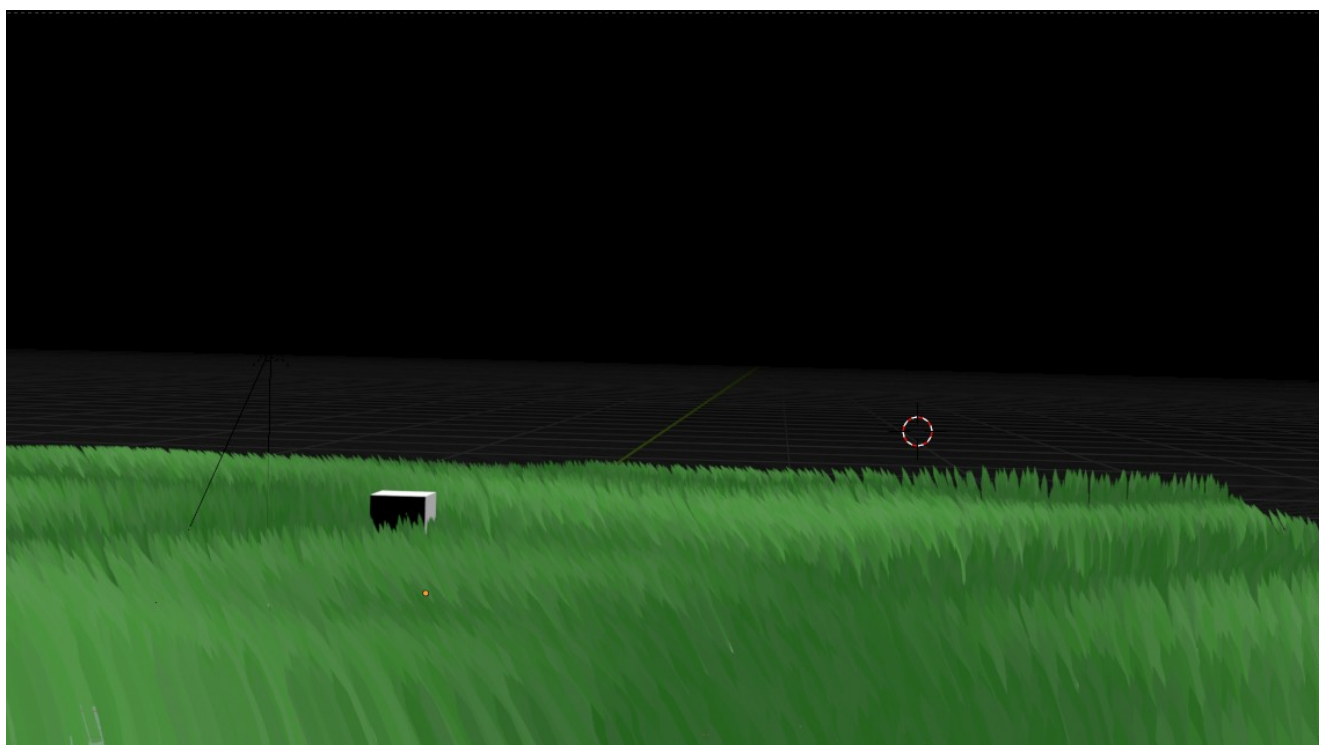


Рисунок 4.19 — Другий кадр стилїзованої сцени

Шумова текстура у нашій системі відіграє ключову роль у створенні відчуття живої, органічної трави. Вона генерує псевдовипадкові значення, які ми використовуємо для зміни параметрів окремих травинок: зокрема, їхнього кута нахилу, положення та навіть відтінку кольору. Завдяки цьому кожна травинка отримує трохи іншу поведінку, що додає сцені природної варіативності. Ці зміни відбуваються постійно у часі, створюючи ефект легкого, м'якого коливання трави під впливом вітру та світла. Це робить нашу сцену не тільки стилізованою, а й динамічною та більш живою.

Однак, незважаючи на це, при перегляді результату стало помітно, що задній план виглядає занадто порожньо та монотонно. Відсутність глибини або деталей у фоні послаблює загальне враження від сцени, навіть якщо передній план виглядає динамічно та насичено[17].

Щоб виправити цю ситуацію та надати фону більше атмосфери, ми вирішили додати скайбокс — текстуроване оточення, що імітує небо або іншу далеку панораму. Це допомогло не лише наповнити простір за межами сцени, а й створити більш реалістичне освітлення та настрій, який гармонійно підтримує нашу стилізовану траву(4.20).



Рисунок 4.20 — Сцена зі стилізованим скайбоксом

У підсумку, розділ про стилізацію трави демонструє, як за допомогою шейдерів можна значно змінити візуальне сприйняття сцени. Ми застосували різноманітні техніки для того, щоб надати траві виразного стилю — від toon-ефекту з різкими тінями до варіативного забарвлення кожної травинки. Завдяки використанню шумових текстур, ми досягли природної різноманітності у формі та кольорі, а також створили ефект коливання, що додає динаміки. Усі ці елементи не тільки покращують естетику сцени, а й дозволяють зберегти її легкість і читабельність, зберігаючи водночас художню цілісність та задану стилізацію.

4.4 Оптимізація та заміри продуктивності

Почнемо з повторення вимог до нашого програмного забезпечення. Для ефективного досягнення мети — створення стилізованих та переконливих сцен із використанням шейдерів — необхідно визначити чіткі функціональні та нефункціональні вимоги до програмного забезпечення. Особливу увагу слід приділити таким аспектам, як продуктивність, інтерактивність та якість графічного відображення. У контексті цього, основні вимоги до продуктивності формулюються наступним чином:

- один кадр повинен рендеритися менш ніж за 1 секунду при роздільній здатності 1920x1080 пікселів;

- у режимі перегляду (viewport) сцена повинна відтворюватися зі швидкістю не менше 30 кадрів на секунду;

- програма має стабільно працювати на середньостатистичних ПК (GTX 1060, 8–16 ГБ ОЗП, процесор середнього рівня);

- шейдери повинні підтримувати гнучке налаштування параметрів;

- візуальні ефекти мають відповідати стилізованому художньому підходу;

- не допускається використання важких обчислень, що значно впливають на час рендеру або плавність роботи.

Ці вимоги враховують специфіку завдання — розробку ефективних і гнучких шейдерів, здатних забезпечити емоційну виразність сцен без надмірного навантаження на систему. Тож перейдемо до порівняння outdoor-lit та indoor-lit сцен за показниками продуктивності. Для об'єктивної оцінки було використано кілька рівнів якості рендерингу, які задаються параметром render samples у Blender. Render samples — це кількість зразків (семплів), що використовуються для обчислення кольору кожного пікселя під час рендерингу. Чим більше значення семплів, тим точніше обчислюється світло, тіні та глобальне освітлення, а отже, краща якість фінального зображення. Однак зі збільшенням кількості семплів зростає й час рендерингу та споживання ресурсів системи, особливо в сценах зі складним освітленням або прозорими/об'ємними матеріалами[18].

У таблиці нижче наведено порівняння двох типів сцен за трьома ключовими показниками продуктивності:

- алокція пам'яті (обсяг оперативної пам'яті, яку виділено для рендерингу сцени),
- час рендеру одного кадру (у секундах);
- загальне навантаження на систему (оцінка відносної складності сцени).

Порівняння було проведено для чотирьох типових значень семплів: 24, 32, 64 та 128, що дозволить оцінити масштаб впливу рівня якості на продуктивність обох типів сцен. Важливо підмітити, що в indoor-lit сцені використовуються досить дорогі техніки освітлення, тож порівнювати вплив шейдерів на час рендеру треба спочатку без них, а після окремо ними(рис 4.21).

Таблиця 4.1 – Дані щодо продуктивності indoor-lit сцени з вимкненим рейтрейсінгом.

Семпли	Алокація пам'яті	Час рендеру
24	723,67(796,65)мб	1,12с
32	723,06(796,77)мб	1,36с
64	723,27(796,99)мб	2,28с
128	723,49(797,20)мб	4,11с

Нижче представлена візуалізація таблиці 4.1 що та пояснення до неї

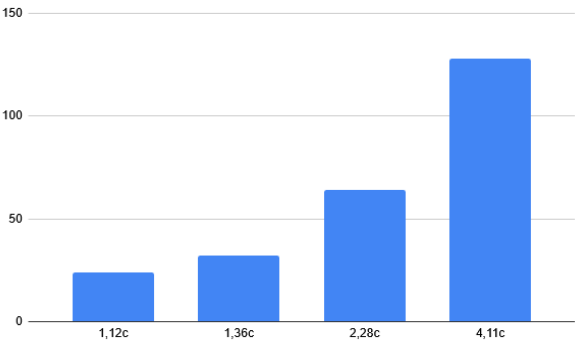


Рисунок 4.21 – залежність часу рендеру від кількості семплів

Як можна помітити, збільшення кількості семплів практично не впливає на алокацію пам’яті — її обсяг залишається майже незмінним. Натомість час рендеру зростає суттєво: від швидкого 1 секунди на кадр до 4 секунд при 128 семплах. Це вказує на те, що головне навантаження при підвищенні якості зображення припадає саме на обчислення, а не на використання пам’яті.

Таблиця 4.2 — Дані щодо продуктивності indoor-lit сцени з ввімкненим рейтрейсінгом.

Семпли	Алокація пам’яті	Час рендеру
24	725,05(799,09)мб	4,77с
32	725,28(799,28)мб	6,05с
64	725,71(799,72)мб	11,53с
128	726,16(800,16)мб	22,46с

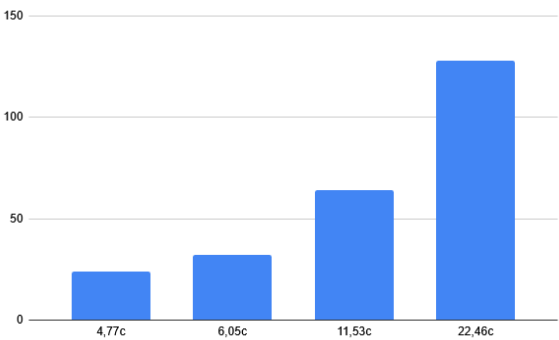


Рисунок 4.22 — графік часу рендерінгу в indoor-lit сцені з рейтрейсінгом

З ввімкненим рейтрейсінгом можемо побачити що навіть найменша кількість семплів вже знаходиться поза досягом середніх систем з 4 секундами на кадр, а при підвищенні кількості семплів взагалі переходить в розділ десятків секунд на кадр(рис 4.22). Також через те що outdoor-lit сцена є досить оптимізованою ми змінимо рамки тестування на 64, 128, 256, 512[19].

Таблиця 4.3 — Дані щодо продуктивності outdoor-lit сцени

Семпли	Алокація пам'яті	Час рендеру
64	132,22(194,40)мб	1,33с
128	132,34(194,51)мб	2,46с
256	132,43(194,61)мб	4,52с
512	132,53(194,70)мб	8,73с

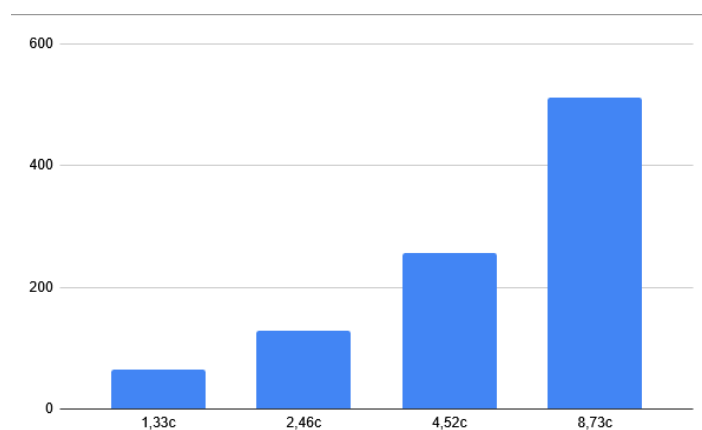


Рисунок 4.23 — графік часу рендеру в outdoor-lit сцені

Як можемо побачити, навіть при збільшенні кількості семплів наша outdoor-lit сцена демонструє стабільну та ефективну роботу під навантаженням(рис 4.23). Завдяки стилізованому художньому підходу, деталізація і точність освітлення не потребують надмірно високої кількості семплів. У таких випадках використання більше ніж 100 семплів вважається зайвим, адже покращення якості зображення при цьому є майже непомітним, тоді як час рендерингу суттєво зростає. Таким чином, для outdoor-lit сцен оптимальним є баланс між достатньою якістю і продуктивністю без надмірного навантаження на систему[20].

ВИСНОВКИ

У рамках дипломного проєкту було здійснено всебічне дослідження та практичну реалізацію концепції створення стилізованих та візуально переконливих 3D-сцен за допомогою розробки та оптимізації шейдерів. Центральною частиною роботи стала розробка двох комплексних демонстраційних сцен – деталізованої внутрішньої (indoor) та масштабної зовнішньої (outdoor) – кожна з яких потребувала індивідуального підходу до візуального стилю та технічної реалізації. Процес створення цих сцен базувався на глибокому зануренні у функціонал Blender, зокрема, на активному використанні його потужних систем Geometry Nodes та розширеного редактора шейдерів (Shader Editor). Завдяки Geometry Nodes, ми змогли реалізувати складні процедурні об'єкти та динамічні ефекти, що значно прискорило процес моделювання та дозволило досягти унікальної стилізації. Водночас, розробка численних кастомних шейдерів стала ключовим елементом у формуванні бажаної естетики, дозволивши гнучко контролювати властивості матеріалів та освітлення відтворюючи різноманітні візуальні стилі. Особлива увага була приділена аспектам продуктивності. Після етапу розробки, було проведено серію детальних замірів продуктивності обох сцен. На основі отриманих даних було проведено цілеспрямовану оптимізацію, яка включала як модифікації геометрії так і оптимізацію шейдерів. Ці заходи дозволили значно покращити швидкодію сцен, забезпечуючи плавну роботу в реальному часі без втрати візуальної якості. Таким чином, даний дипломний проєкт не тільки продемонстрував практичні навички в розробці складного 3D-контенту але й підкреслює глибоке розуміння принципів комп'ютерної графіки, зокрема, в частині шейдерної розробки, процедурного моделювання та оптимізації продуктивності. Отриманні результати є свідченням нашої здатності вирішувати комплексні завдання у сфері 3D-візуалізації та застосовувати передові інструменти для досягнення високої якості та ефективності.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

- 1.Lens, S. Procedural 3D Modeling Using Geometry Nodes in Blender. Blender Market (2022). URL: <https://blendermarket.com/products/procedural-3d-modeling-using-geometry-nodes>
- 2.Green, C. Optimizing Game Models: A Comparison of Geometry Nodes and Traditional Modelling Techniques in Blender. Theseus.fi (2021). URL: https://www.theseus.fi/bitstream/handle/10024/799858/Green_Christy.pdf?sequence=2
- 3.Kogut, W. The analysis of Blender open-source software cloth simulation capabilities. Journal of Computer Sciences Institute (2023). URL: https://www.researchgate.net/publication/369673349_The_analysis_of_Blender_open-source_software_cloth_simulation_capabilities
- 4.Bohak, C. 3D Modeling using Blender. LGM FRI (2022). URL: <https://lgm.fri.uni-lj.si/wp-content/uploads/2022/06/Blender-2022-eng.pdf>
- 5.Hautakangas, A. Fur Simulation. Theseus.fi (2024). URL: https://www.theseus.fi/bitstream/handle/10024/863065/Hautakangas_Anu.pdf?sequence=2
- 6.Green, C. Optimizing Game Models: A Comparison of Geometry Nodes and Traditional Modelling Techniques in Blender. Theseus.fi (2021). URL: https://www.theseus.fi/bitstream/handle/10024/799858/Green_Christy.pdf?sequence=2
- 7.Lagae, A. A Survey of Procedural Noise Functions. Computer Graphics Forum (2010). URL: <https://www.cs.umd.edu/~zwicker/publications/SurveyProceduralNoise-CGF10.pdf>
- 8.Hudon, M. 2DToonShade: A stroke based toon shading system. ScienceDirect (2019). URL: <https://www.sciencedirect.com/science/article/pii/S2590148619300032>
- 9.Gonzalez Vivo, P., Lowe, J. The Book of Shaders (2015). URL: <https://thebookofshaders.com/>

10. Pang, K. Pixel Perfect Dithering Shader in Blender. Medium (2022). URL: <https://medium.com/@pangkhangee/pixel-perfect-dithering-shader-in-blender-fd6898e8ba23>
11. Müri, K. Setting Up Watercolor Renders in Blender Eevee. 80.lv (2023). URL: <https://80.lv/articles/setting-up-watercolor-renders-in-blender-eevee>
12. Kent, B. R. 3D Scientific Visualization with Blender®. IOP Publishing (2015). URL: <https://iopscience.iop.org/book/mono/978-1-6270-5612-0>
13. Blender Foundation. Shader Editor – Blender Manual (2024). URL: https://docs.blender.org/manual/en/latest/editors/shader_editor.html
14. Creating Fluffy Stylized Grass in Blender Eevee. 80.lv (2021). URL: <https://80.lv/articles/creating-fluffy-stylized-grass-in-blender-eevee>
15. Haines, E., Akenine-Möller, T. Real-Time Rendering, 3rd Edition (2011). URL: https://www.realtimerendering.com/Real-Time_Rendering_4th-TOC_Preface_Intro_Bib_Index.pdf
16. Vince, J. Vector Analysis for Computer Graphics (2011) 246c.
17. Chu, W. The Method to Add Sky Background in 3D Virtual Engine. IOP Conference Series (2020). URL: <https://iopscience.iop.org/article/10.1088/1742-6596/1673/1/012010/pdf>
18. Barkovska, O. Research on Model Rendering Performance in Blender 3D. ResearchGate (2023). URL: https://www.researchgate.net/publication/375260393_Research_on_Model_Rendering_Performance_in_Blender_3D_Using_Massively_Parallel_Systems.
19. Lefebvre, S., Hornus, S. An Introduction to Optimization Techniques in Computer Graphics. Inria (2014). URL: <https://inria.hal.science/hal-00976357>
20. Optimization in Computer Graphics and Interactive Techniques. ACM (2024). URL: <https://dl.acm.org/doi/10.1145/3664475.3664562>

Національний університет “Запорізька політехніка”

Факультет комп’ютерних наук і технологій

Кафедра комп’ютерних систем та мереж

Розробка та оптимізація шейдерів для створення стилізованих та переконливих сцен

Виконавець:
ГЕТМАН Сергій Валерійович
студент групи КНТ-511

Керівник:
КИРИЧЕК Галина Григорівна
к. т. н., доцент кафедри
комп’ютерних систем та мереж

2025р.

МЕТА РОБОТИ

Мета

Розробка та оптимізація шейдерів для створення стилізованих та переконливих сцен у Blender, з метою покращення візуального сприйняття без значних втрат продуктивності.

Предмет дослідження

Шейдери та техніки не-фотореалістичного рендеру. Комп’ютерна графіка.

МЕТА РОБОТИ

- У сучасному цифровому середовищі зростає попит на візуально привабливий та оптимізований графічний контент для ігор, Стилїзована графіка дозволяє не лише зменшити навантаження на систему, а й створити унікальну художню атмосферу, що вирізняє проєкт серед інших
- Використання процедурних методів моделювання та власних шейдерів відкриває широкі можливості для оптимізації сцен у реальному часі без втрати естетичної якості
- Актуальність теми зумовлена потребою поєднання технічної ефективності та творчого підходу при створенні сучасного 3D-контенту

3

АНАЛІЗ АЛЬТЕРНАТИВ

- Серед відомих альтернатив Blender-у як програмі для створення 3-D сцен відомі:
 - cinema 4D;
 - houdini;
 - zbrush.

Також, варто відмітити Unity та Unreal Engine, але вони будучи ігровими рушіями, скоріш доповнюють існуючий функціонал.

4

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Основні вимоги до продуктивності:

- один кадр повинен рендеритися менш ніж за 1 секунду при роздільній здатності 1920x1080 пікселів;
- у режимі перегляду (viewport) сцена повинна відтворюватися зі швидкістю не менше 30 кадрів на секунду;
- програма має стабільно працювати на середньостатистичних ПК (GTX 1060, 8–16 ГБ ОЗП, процесор середнього рівня);
- шейдери повинні підтримувати гнучке налаштування параметрів;
- візуальні ефекти мають відповідати стилізованому художньому підходу;
- не допускається використання важких обчислень, що значно впливають на час рендеру або плавність роботи.

6

ПОСТАНОВКА ЗАВДАНЬ

Під час розробки було прийнято рішення розробити 2 сцени, що охопили б значущу частину проблем з якими стикаються розробники під час створення 3-D сцен. Такими сценами стали сцена інтер'єру(indoor-lit scene) та сцена просто неба(outdoor-lit scene).

7

ТЕХНІКИ РОЗРОБКИ



8

INDOOR СЦЕНА



Після процесу розробки що більш з залученням всіх перелічених технік було отриману ось таку неосвітлену сцену інтер'єру

9

ОСВІТЛЕННЯ

Blender надає широкий вибір інструментів для створення освітлення у 3-D сценах. Під час розробки indoor сцени ми використали:

- світлові джерела (Point, Sun, Area, Spot);
- volume освітлення;
- трасування променів(Ray tracing).

Ray tracing (трасування променів) — це технологія, яка дозволяє робити освітлення у 3D-сценах дуже реалістичним. Вона «простежує» шлях світла у віртуальному просторі — як він відбивається, проходить крізь об'єкти та створює тіні й відбиття.

10

INDOOR СЦЕНА

В результаті після освітлення сцени було отримано такий результат



11

ШЕЙДЕРИ

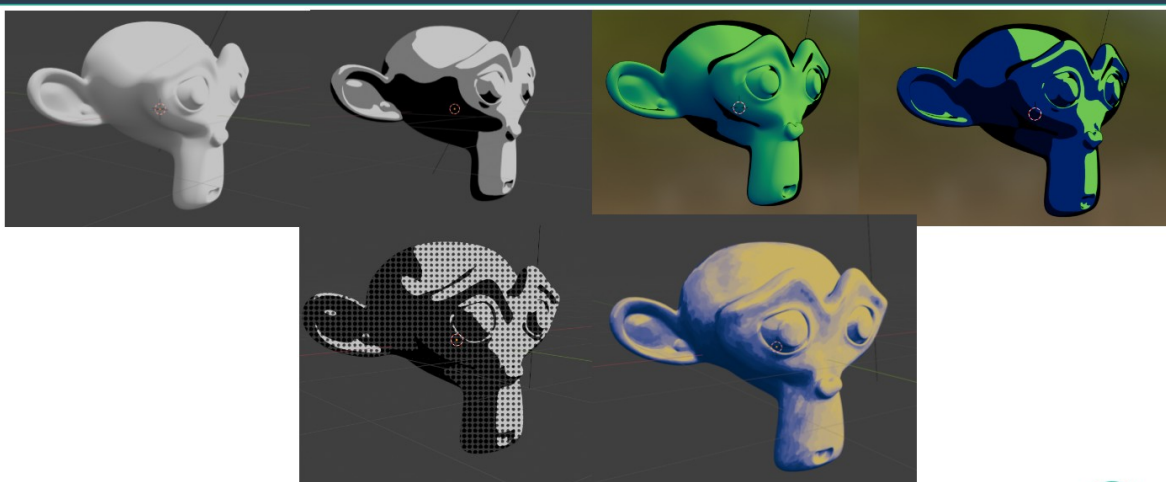
Наступним шагом в розробці стало створення шейдерів для нашої сцени.

Було розроблено:

- toon-shader;
- outline-shader;
- dither(halftone)-shader;
- watercolor-shader.

12

ДЕМОНСТРАЦІЯ ШЕЙДЕРІВ



13

ВИСНОВКИ

- Досліджено теоретичні основи функціонування шейдерів, їх класифікацію та роль у формуванні візуальних ефектів у 3D-графіці.
- Детально вивчено математичний апарат, що лежить в основі 3D-графіки та роботи шейдерів, включаючи векторний аналіз, матричні перетворення та інші необхідні математичні концепції.
- Проаналізовано сучасні підходи до розробки та оптимізації шейдерів, а також програмні засоби та інструменти, що використовуються в даній галузі.
- Розроблено та реалізовано графічний прототип, що включає систему камери, яка дозволяє вільно переміщуватися та змінювати орієнтацію у тривимірному просторі.
- Імплементовано систему шейдерів для створення різноманітних графічних ефектів, таких як освітлення, відображення об'єктів та стилізована візуалізація.
- Реалізовано функціонал для додавання фонових зображень та об'єктів до сцени, що дозволяє створювати комплексні та завершені візуальні композиції.
- Підтверджено можливість створення переконливих та стилізованих сцен за допомогою програмно реалізованих шейдерів, що відповідає основним цілям дипломного проєкту.
- Отримані результати демонструють ефективність використаних підходів в частині візуалізації та художньої виразності створюваних сцен.