

Міністерство освіти і науки України  
Університет митної справи та фінансів

Факультет інноваційних технологій  
Кафедра комп'ютерних наук та інженерії програмного забезпечення

### Кваліфікаційна робота магістра

на тему: «Комп'ютерне 3D моделювання задач проєктування з  
використанням технології Unity»

Виконав: студента групи К22-1М  
Спеціальність 122 «Комп'ютерні науки»

Недря С.В.  
(прізвище та ініціали)

Керівник к.т.н., доц. Ульяновська Ю.В.  
(науковий ступінь, вчене звання, прізвище та ініціали)

Рецензент Університет митної справи та фінансів  
(місто роботи)

Доцент кафедри кібербезпеки та  
(посада)

інформаційних технологій  
(посада)

к.т.н., доц. Прокопович-Ткаченко Д.І.  
(науковий ступінь, вчене звання, прізвище та ініціали)

Дніпро – 2024

## АНОТАЦІЯ

*Недря С.В.* Комп'ютерне 3D моделювання задач проєктування з використанням технології Unity.

Дипломна робота на здобуття освітнього ступеня магістр за спеціальністю 122 «Комп'ютерні науки». – Університет митної справи та фінансів, Дніпро, 2024.

Об'єктом дослідження є процес 3D – моделювання ландшафту.

Предмет дослідження це методи та алгоритми 3D-моделювання та їх програмна реалізація

Метою роботи є реалізація 3D-моделювання реального світу в Unity.

Ця робота присвячена аналізу та застосуванню комп'ютерного 3D моделювання в моделюванні ландшафту за допомогою платформи Unity. В роботі розглядаються основні функції та можливості Unity, включаючи інтеграцію з віртуальною та доповненою реальністю, що відкриває нові перспективи для представлення проєктів та взаємодії з ними. Використовуючи ряд прикладів, дослідження демонструє, як Unity може слугувати не лише як інструмент для створення візуалізацій, але й як платформа для комплексного аналізу та реалізації архітектурних та ландшафтних проєктів. Результатом розробки є додаток, який відкриває широкі можливості для моделювання завдань архітектури. Шляхом створення реалістичної 3D-моделі поверхні та ефективного розміщення моделей об'єктів, програма надає зручний та ілюстративний інструмент для візуалізації вигляду системи постачання води.

Ключові слова: 3D-моделювання, Unity, DDA, MapBox, OpenStreetMap.

## ABSTRACT

*Nedria S.V.* Computer 3D modeling of design tasks using Unity technology.  
Diploma thesis (project) for the degree of Master's Degree in specialty 122  
"Computer Science." - University of Customs and Finance, Dnipro, 2024.

The object of study is the process of 3D modeling.

The subject of the study is the methods and algorithms of 3D modeling and their software implementation.

The purpose of the work is to implement 3D modeling of the real world in Unity.

This work is dedicated to the analysis and application of computer 3D modeling in landscape modeling using the Unity platform. The paper examines the main functions and capabilities of Unity, including integration with virtual and augmented reality, which opens new perspectives for project presentation and interaction. Using a series of examples, the research demonstrates how Unity can serve not only as a tool for creating visualizations but also as a platform for comprehensive analysis and implementation of architectural and landscape projects. The result of the development is an application that opens wide possibilities for modeling architectural tasks. By creating a realistic 3D model of the surface and efficient placement of object models, the program provides a convenient and illustrative tool for visualizing the appearance of the water supply system.

Key words: 3D modeling, Unity, DDA, MapBox, OpenStreetMap.

## ЗМІСТ

ВСТУП .....	5
РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ.....	8
1.1 Аналіз публікацій щодо 3D моделювання.....	8
1.2 Аналіз методів 3D моделювання .....	12
1.3 Висновки до першого розділу.....	20
РОЗДІЛ 2 СТВОРЕННЯ ВІРТУАЛЬНИХ СЕРЕДОВИЩ НА ОСНОВІ ВІДКРИТИХ ГЕОПРОСТОРОВИХ ДАНИХ ТА ІГРОВОГО РУШІЯ UNITY .....	22
2.1 3D-моделювання ландшафту .....	22
2.2 Методи обробки топографічних даних з Unity. ....	23
2.3 Покращення методу моделювання.....	32
2.4 Висновок розділу.....	35
РОЗДІЛ 3 РОЗРОБКА 3D МОДЕЛІ ПОБУДОВИ ВОДОПОСТАЧАННЯ.....	36
3.1 Актуальність розробки 3D моделювання завдання.....	36
3.2 Інструменти розробки.....	38
3.3 Розробка додатку.....	42
3.4 Тестування створеного продукту .....	57
3.5 Висновок розділу.....	62
ВИСНОВОК.....	63
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	65
ДОДАТОК А.....	67

## ВСТУП

Комп'ютерне 3D-моделювання відіграє ключову роль в сучасній цифровій епохі, дозволяючи нам створювати деталізовані моделі реального світу. Застосування цих технологій охоплює широкий спектр галузей – від архітектури та інженерії до розваг та освіти, змінюючи способи, якими ми навчаємось, працюємо та відпочиваємо. Однією з передових платформ для реалізації проектів у цій області є Unity – потужний інструмент для розробки відеоігор та інтерактивного контенту, який відкриває безмежні можливості для 3D моделювання.

Прогрес в сфері інформаційних технологій та комп'ютерного обладнання призвів до глибоких змін у суспільстві та окремих його аспектах. Різні галузі приймають новітні методи для мінімізації часу, необхідного для виконання завдань. Комп'ютерна графіка відіграє ключову роль на кожному етапі архітектурного проектування, дозволяючи розробляти проекти від концепції до віртуальної візуалізації. Використання 3D графіки та анімації надає архітекторам засоби для ідеального представлення їхніх ідей, дозволяючи ефективно та гнучко показувати проекти і презентації. Основним елементом реалізації успішного та вражаючого проекту є застосування передового програмного забезпечення [1].

*Актуальність* теми комп'ютерного 3D моделювання з використанням технології Unity полягає в можливості для дизайну, планування та візуалізації. Дозволяє архітекторам, містобудівникам та ландшафтним дизайнерам створювати детальні та реалістичні 3D-моделі, які можуть бути використані для демонстрації проектів клієнтам, проведення віртуальних турів та оцінки впливу новобудов на навколишнє середовище до початку будівельних робіт. Unity також сприяє співпраці між фахівцями різних дисциплін, оскільки проект може бути легко поділений та переглянутий в реальному часі, дозволяючи командам ефективно працювати над складними проектами.

Окрім того, інтеграція з іншими технологіями, такими як віртуальна та доповнена реальність, розширює можливості використання моделей, створених у Unity, для інтерактивного досвіду, освіти та маркетингу. Враховуючи ці аспекти, тема комп'ютерного 3D моделювання ландшафту та архітектури за допомогою Unity.

*Новизна дослідження* полягає в інтеграції передових цифрових технологій з традиційними методами проектування. Unity, первісно розроблена як інструмент для створення відеоігор. Було запропоновано використати її як потужну платформу для архітектурної візуалізації та відображення ландшафту поверхні Землі в реальному часі, пропонуючи можливості, які раніше були недоступні або вимагали значних витрат часу та ресурсів.

*Метою дослідження є* реалізація 3D-моделювання реального світу технологіями Unity.

Для досягнення поставленої мети в кваліфікаційній роботі ставились та вирішувались наступні *завдання дослідження*:

1. Провести аналіз методів 3D-моделювання.
2. Визначити, підхід до 3D-моделювання ландшафту реального світу в Unity.
3. Розглянути можливі модифікації алгоритму, реалізувати на практиці та визначити, який метод є найбільш ефективним для різних типів рішень.
4. Розробити програмне забезпечення, засноване на модифікованому способі моделювання ландшафту.

*Методи дослідження* в роботі використовувалися методи 3D-моделювання, ігровий рушій Unity, мова програмування C#.

*Об'єкт дослідження*: Процес 3D – моделювання.

*Предмет дослідження*: Методи та алгоритми 3D-моделювання та їх програмна реалізація.

*Практичне значення отриманих результатів:* Автоматизація процесу 3D-моделювання в реальному часі, прискорення та спрощення 3D-моделювання.

Структура роботи:

Розділ 1 Дослідження предметної області та постановка задачі дослідження. У даному розділі буде виконано пошук та аналіз методів побудови 3D-моделей.

Розділ 2 Створення віртуальних середовищ на основі відкритих геопросторових даних та ігрового рушія Unity. В даному розділі буде проаналізовано та обрано кращий метод та 3D-моделювання ландшафту.

Розділ 3 Розробка 3d моделі побудови водопостачання. В даному розділі буде спроектовано та розроблено програму для вирішення важливих задачі проектування водогону.

*Робота складається* зі вступу, 3-х розділів, висновків, списку використаної літератури з 18 джерел, 1 додатку. Обсяг роботи 71 сторінок, 52 рисунків.

## РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ.

### 1.1 Аналіз публікацій щодо 3D моделювання

Швидкий прогрес в сфері інформаційних технологій та комп'ютерного обладнання призвів до глибоких змін у суспільстві та окремих його аспектах. Різні галузі приймають новітні методи для мінімізації часу, необхідного для виконання завдань. Комп'ютерна графіка відіграє ключову роль на кожному етапі архітектурного проектування, дозволяючи розробляти проекти від концепції до віртуальної візуалізації. Використання 3D графіки та анімації надає архітекторам засоби для ідеального представлення їхніх ідей, дозволяючи ефективно та гнучко показувати проекти і презентації. Основним елементом реалізації успішного та вражаючого проекту є застосування передового програмного забезпечення [1].

Сучасні технології в області комп'ютерної графіки відкривають перед майбутніми архітекторами безпрецедентні можливості для розвитку їхнього творчого потенціалу. Завдяки появі високопотужних комп'ютерів стало можливим втілювати в життя складні математичні алгоритми для інтеграції та візуалізації графічних об'єктів, що дозволяє архітекторам створювати деталізовані віртуальні проекти їхніх майбутніх творінь. Комп'ютерне моделювання перетворює процес архітектурного проектування, створюючи віртуальне середовище, де можливо динамічно формувати об'єкти в контексті їх природного або штучного середовища. Це досягається за допомогою комп'ютерних програм, які дозволяють враховувати навіть найдрібніші деталі рельєфу території.

Тривимірна графіка, часто відома як 3D-моделювання, є видом комп'ютерної графіки, що включає використання різноманітних методів та інструментів для створення об'ємних об'єктів у тривимірному просторі.



Відмінність 3D-зображень від двовимірних полягає в тому, що вони дозволяють створювати геометричні проєкції тривимірних моделей на площину за допомогою спеціальних програмних засобів. Створені моделі можуть як відображати реальні об'єкти, так і бути повністю абстрактними.

Використання 3D-графіки дозволяє імітувати процес фотографування або відеозйомки для об'єктів, створених в комп'ютерній пам'яті. Процес створення 3D-зображень включає кілька етапів: підготовку, конструювання геометричної моделі сцени, налаштування освітлення та камер, вибір та налаштування матеріалів для об'єктів, а також безпосередньо візуалізацію сцени. Таким чином, створюється віртуальний тривимірний світ.

Тривимірне моделювання надає можливість створювати високодеталізовані моделі, які максимально наближені до реальних об'єктів, значно підвищуючи наочність та зрозумілість проєктів. Це особливо важливо, коли мова йде про проєктування складних тривимірних об'єктів, які важко представити на двовимірній площині.

Застосування 3D-моделювання охоплює широкий спектр сфер діяльності, включаючи:

**Реклама та маркетинг:** Для візуалізації майбутніх продуктів, розробки дизайну упаковки, стендів для виставок та іншої зовнішньої реклами.

**Міське планування:** Моделювання архітектури та ландшафтів міст, дозволяючи архітекторам та інвесторам краще уявити проєктовані простори.

**Промисловість:** Проєктування деталей та об'єктів для створення прототипів і прес-форм, економія матеріалів та зниження витрат.

**Комп'ютерні ігри:** Створення тривимірних моделей героїв, ландшафтів, анімацій та концепт-арту, що використовуються у розробці ігрового світу та інтерактивного геймплею.

**Кінематограф:** Використання спеціалізованих 3D-камер для зйомок та створення віртуальних ландшафтів та об'єктів, що забезпечує нові можливості для фільммейкінгу.

Архітектура та дизайн інтер'єрів: Незамінне застосування при проектуванні будівель та внутрішнього оформлення, дозволяючи створювати деталізовані прототипи і візуалізації майбутніх споруд.

Анімація: Створення рухомих 3D-персонажів та анімованих відеороликів, що відкриває широкі можливості для аніматорів і режисерів.

В ході геологічних розвідок визначають структуру глибоких земних шарів. В області медицини технології комп'ютерної візуалізації відіграють ключову роль у діагностиці внутрішніх органів, використовуючи двовимірну графіку для аналізу, наприклад, енцефалограм та міограм, а тривимірну візуалізацію застосовують у томографії для отримання більш об'ємної картини. Результати таких досліджень можна відразу бачити на комп'ютерному моніторі.

Значний інтерес до тривимірної графіки та 3D-моделювання підтверджується зростанням обсягів пошукових запитів з цих тем за останні п'ять років у всьому світі, включаючи Україну. Завдяки стрімкому розвитку комп'ютерних технологій та вдосконаленню програмного забезпечення для 3D-графіки, цей напрямок продовжує активно розвиватися, привертаючи увагу спеціалістів з різних сфер, включаючи інженерів, дизайнерів, архітекторів, розробників ігор та науковців у таких областях, як медицина та астрономія. Тривимірне моделювання відіграє ключову роль у сучасному дизайні та виробництві, дозволяючи створювати більш реалістичні та детальні проекти.

Автори досліджень підкреслюють значення використання цих технологій для підвищення якості та ефективності архітектурного проектування. Зокрема, існує широкий спектр програмного забезпечення, призначеного для моделювання природних і штучних середовищ, включаючи геоінформаційні системи, 3D-редактори, інструменти для ландшафтного дизайну та інші спеціалізовані рішення. Кожна програма пропонує унікальний набір інструментів та функцій, що дозволяє досягати різноманітних цілей проектування.

Дослідження в області спеціалізованого програмного забезпечення для 3D-моделювання ландшафту вказують на важливість побудови тривимірних моделей, які можуть відтворювати реальні об'єкти з високим рівнем деталізації. Проте, таке моделювання вимагає значних обчислювальних ресурсів. Водночас, деякі програми надають можливість користувачам без спеціальної підготовки створювати простіші 3D-моделі, роблячи процес більш доступним.

Генератори ландшафтів використовують складні алгоритми для створення реалістичних моделей місцевості, базуючись на географічних даних та картах висот. Вони можуть використовувати як реальні дані з геоінформаційних систем, так і генерувати рельєф згідно з параметрами, заданими користувачем. Це дозволяє детально відтворити природний ландшафт або створити унікальні штучні середовища. Додатково, функціональність цих програм може бути розширена за допомогою плагінів, що надає користувачам ще більше інструментів для реалізації їх творчих задумів.

У сфері автоматизованого проектування, моделі територій створюються шляхом адаптації до конкретних вимог проекту, заснованих на топографічних даних. Це дозволяє ефективно та якісно розробляти проекти, забезпечуючи високу точність відтворення місцевості [10].

Застосування процесів Building Information Modeling (BIM) сприяє значному підвищенню продуктивності в будівельній галузі, дозволяючи вирішити багато актуальних проблем, зокрема, пов'язаних з обміном інформацією між учасниками проекту. Інформаційне моделювання будівель значно спрощує роботу з проектами, надаючи переваги над традиційними методами проектування. Воно дозволяє віртуально розробити та скоординувати всі компоненти та системи майбутньої споруди, забезпечуючи перевірку їх життєздатності, функціональності та експлуатаційних характеристик до початку будівельних робіт.

Завдяки цим технологіям, архітектори та проектанти мають у своєму розпорядженні потужні інструменти для реалізації найсміливіших ідей, забезпечуючи при цьому високу якість та ефективність проектних робіт.

## 1.2 Аналіз методів 3D моделювання

Формування об'єктів у 3D-моделюванні можливе за допомогою різноманітних методик і інструментів. Одним із способів є екструдування, яке полягає у витягуванні нових форм із вже існуючих точок, ліній або поверхонь. Цей процес схожий на витягування форми з масиву матеріалу. Інший метод, скульптинг, нагадує роботу з глиною, де моделіст формує об'єкти з урахуванням їхньої анатомії, такі як людське тіло. Також існує методика створення об'єктів за допомогою кривих, що ідеально підходить для моделювання складних лінійних структур, наприклад, мотузок або бинтів.

Матеріал, який призначається для об'єкта, визначається набором характеристик, що впливають на те, як об'єкт взаємодіє зі світлом, та як він виглядає у віртуальному просторі. На цьому етапі стає очевидним, що для кожної специфічної задачі може знадобитися окремий інструмент або навіть спеціалізоване програмне забезпечення.

Для глибшого розуміння цього процесу важливо ознайомитися з програмним забезпеченням, необхідним для створення тривимірних моделей та їх подальшої анімації. Відразу варто відмовитися від програм, які спеціалізуються лише на фізиці або генерації об'єктів через сканування фотографій, оскільки наша мета — моделювання об'єктів та призначення їм матеріалів. Для цих цілей можна використовувати такі програми як Zbrush, Blender, 3Ds MAX або MAYA для моделювання, а для накладення матеріалів підійдуть Substance Painter або Quixel Mixer, які дозволяють додати об'єктам реалістичні текстури та матеріальні характеристики.

ZBrush є програмним забезпеченням для роботи з тривимірною графікою, розробленим компанією Pixologic. Його унікальна властивість полягає в тому, що воно дозволяє користувачам імітувати процес ручного ліплення, подібно до роботи з глиною, що робить його ідеальним для створення деталізованих скульптурних об'єктів. Завдяки вбудованому рушію рендерингу в реальному часі, програма забезпечує миттєве відображення змін, спрощуючи процес моделювання та дозволяючи художникам бачити кінцевий результат без затримок. Це особливо корисно при створенні складних персонажів або об'єктів, де потрібно досягти високого рівня деталізації.

Програма має інтуїтивно зрозумілий інтерфейс з набором кистей, що нагадує інструменти таких програм, як Photoshop або Krita. Її унікальний набір інструментів дозволяє швидко та ефективно створювати тривимірні форми будь-якої складності. ZBrush також дозволяє створювати базові матеріали для об'єктів, що додає додаткові можливості для реалізації творчих ідей.

Основним недоліком програми є висока вартість ліцензії, однак враховуючи її спеціалізоване призначення та унікальні можливості, цей аспект не можна вважати критичним. ZBrush ефективно виконує своє основне завдання — "ліплення" з цифрової "глини", що зробило його стандартом у галузі [2].

Однак, для завдань, пов'язаних з розробкою архітектури, а також для деяких специфічних задач моделювання, таких як створення тривимірних об'єктів для анімації, що вимагають ретопології, точної масштабності або екструдювання, ZBrush може виявитися не найкращим вибором без використання додаткових інструментів.

3Ds Max, також відомий як 3D Studio Max, є комплексним тривимірним графічним редактором, розробленим компанією Autodesk. Це програмне забезпечення служить універсальним інструментом для тривимірного моделювання, оскільки воно не обмежується лише створенням і редагуванням об'єктів. 3Ds Max дозволяє користувачам займатися анімацією об'єктів,

детальним накладанням матеріалів, розробкою каркасів (ригів), що необхідно для створення рухомих моделей, як-от персонажів.

Основною перевагою цього програмного забезпечення є його багатофункціональність. 3Ds Max пропонує широкий набір інструментів та опцій, що дозволяє безперешкодно реалізувати проекти будь-якої складності, від простих моделей до комплексних сцен анімації. Однак, серед недоліків програми слід відзначити її відносно високу ціну та питання оптимізації, оскільки двигун та функціонал програми довгий час не оновлювалися [3].

Незважаючи на зазначені недоліки, 3Ds Max продовжує залишатися стандартом у галузі візуалізації архітектурних проектів, інтер'єрів, а також у розробці інженерних виробів та інших продукційних моделей. Програма також ефективно використовується для створення анімації, роблячи її одним із переважних виборів серед професіоналів у цій сфері.

Що стосується МАҮА, іншого тривимірного редактора від Autodesk, вона також відзначається схожими перевагами. Проте, МАҮА більш фокусується на анімації та візуальних ефектах, хоча і надає достатньо інструментів для ефективного моделювання. Це робить МАҮА особливо цінною для проектів, які вимагають високої якості анімації та візуальних ефектів, дозволяючи створювати об'єкти та комплекси, які можуть бути використані у власних проектах або навіть пропонуватися для продажу.

Blender представляє собою відкритий програмний пакет для створення 3D графіки, який вирізняється своєю універсальністю. Як справжній "все-в-одному" інструмент, Blender об'єднує можливості моделювання, скульптури, ригінгу (створення каркасу), візуалізації, композитингу та трекінгу руху. Додатково, програма містить вбудований відеоредактор та якісний фізичний двигун для симуляцій.

Оновлення двигуна програми забезпечує високу оптимізацію при роботі з проектами, що містять малу кількість полігонів, але може виявитися менш ефективним у порівнянні з іншими програмами при обробці сцен з великою кількістю деталей. Інтерфейс Blender, хоч і може здатися складним на перший

погляд, є досить зручним після звикання, оскільки функції чітко поділені по категоріях.

Blender, будучи безкоштовним і відкритим програмним забезпеченням, надає користувачам можливість його модифікації та налаштування під індивідуальні потреби. Важливою перевагою є те, що Blender уникає необхідності використання декількох програм для різних аспектів 3D моделювання, хоча і має деякі обмеження порівняно з 3Ds Max і ZBrush, зокрема, в меншому виборі кистей для скульптури та менш розвиненому процедурному моделюванні [4].

Моделювання в Blender можливе завдяки різноманітним методам: від класичного підходу з використанням базових форм і модифікаторів до більш складних технік, таких як скульптинг та робота з кривими. Програма також пропонує засоби для створення анімацій, включаючи детальний ригінг і використання драйверів для контролю руху об'єктів. Ригінг у Blender дозволяє користувачам створювати складні каркаси для моделей, що забезпечують точний контроль над анімацією персонажів або інших об'єктів. Це може включати як інверсну, так і пряму кінематику, що дає можливість реалізовувати натуралістичні рухи та пози.

Додатково, Blender використовує систему частинок і геометричні ноди для додання складності і деталізації сценам, як-от створення волосся, трави або навіть дощу. Нодова система дозволяє з'єднувати різні параметри і ефекти в складні мережі, що впливають на властивості і вигляд об'єктів, забезпечуючи безмежні можливості для креативності.

Blender також містить потужні засоби для візуалізації, дозволяючи користувачам створювати фотореалістичні зображення та анімації завдяки своїм високоякісним рендер-двигунам, таким як Cycles та Eevee. Ці двигуни пропонують різні підходи до рендерингу, з Eevee, який забезпечує швидке візуалізування в реальному часі, та Cycles, що пропонує глибокий рівень реалізму через трасування променів.

Композитинг і пост-продакшн можуть бути виконані прямо всередині Blender, надаючи користувачам інструменти для обробки зображень, додавання візуальних ефектів та зміни кольорової гами, що робить Blender не просто програмою для 3D моделювання, але і повноцінним рішенням для створення завершених візуальних проектів.

Завершуючи, можна сказати, що Blender є інструментом з надзвичайно широким спектром можливостей для 3D художників, аніматорів, інженерів та дизайнерів, що дозволяє втілювати найсміливіші ідеї у реальність, не обмежуючись рамками одного програмного забезпечення. Його безкоштовна ліцензія та активна спільнота користувачів забезпечують постійний розвиток і підтримку, роблячи Blender важливим інструментом у світі цифрового дизайну та анімації [5].

Unity – це потужний інструмент для розробки ігор та інтерактивних додатків, який забезпечує розробникам комплексний набір функцій для створення 2D, 3D, а також віртуальної (VR) та доповненої реальності (AR) проектів. Завдяки своїй гнучкості та доступності, Unity стало одним з найпопулярніших двигунів у галузі відеоігор, використовуючись як незалежними розробниками, так і великими студіями.

Основна перевага Unity полягає у його здатності до крос-платформної розробки, дозволяючи розробникам створювати ігри та додатки, які можуть працювати на понад 20 різних платформах, включаючи Windows, macOS, Android, iOS, Linux, а також ігрові консолі та VR/AR пристрої. Це робить Unity ідеальним вибором для розробки проектів з широкою аудиторією.

Unity пропонує розробникам використання C# для скриптування, що є зручним і потужним мовою програмування, доступним для навчання. Це дозволяє створювати складні механіки та поведінку об'єктів у ігровому світі, від простих інтерактивних дій до складних систем штучного інтелекту.

Unity також включає в себе потужний набір інструментів для візуалізації та анімації, що дозволяє детально налаштовувати графіку і візуальні ефекти. Від реалістично оформлених 3D сцен до художньо стилізованих 2D



персонажів, Unity забезпечує гнучкість для реалізації будь-якого візуального стилю. Завдяки інтеграції з такими інструментами як Adobe Photoshop, розробники можуть легко імпортувати асети та працювати над ними безпосередньо у проекті [6].

Unity не тільки відоме як потужний інструмент для розробки ігор, але й як платформа, що надає широкі можливості для 3D моделювання та анімації. Хоча Unity не містить вбудованих інструментів для створення складних 3D моделей з нуля, як це роблять спеціалізовані програми типу Blender, Maya, чи 3Ds Max, воно ідеально підходить для імпортування, оптимізації, анімації та рендерингу 3D асетів, створених у цих програмах.

Unity підтримує широкий спектр форматів файлів 3D моделей, включаючи популярні FBX, OBJ, DAE та інші. Це дозволяє розробникам легко інтегрувати в свої проекти моделі, створені з використанням зовнішніх інструментів моделювання, і використовувати потужні функції Unity для подальшої роботи з ними.

Однією з ключових переваг Unity в контексті 3D моделювання є його система освітлення та рендерингу, яка дозволяє створювати високоякісні візуальні ефекти та фотореалістичні сцени. Unity пропонує різноманітні типи освітлення, включаючи направлене світло, точкове світло, прожектори та області світла, а також підтримує технології Global Illumination та Real-Time Ray Tracing для надання сценам ще більшої реалістичності.

Крім того, Unity містить інструменти для ригінгу та анімації 3D моделей. З його допомогою можна створювати складні анімаційні секвенції, використовуючи ключові кадри, анімаційні криві та систему механізм для створення плавних та реалістичних рухів персонажів та об'єктів.

Unity також пропонує розширені можливості для роботи з текстурами та матеріалами, дозволяючи розробникам детально налаштовувати вигляд своїх 3D моделей. Завдяки підтримці шейдерів та системі візуального програмування Shader Graph, користувачі можуть створювати складні

матеріали з унікальними властивостями та ефектами, такими як блиск, прозорість, рельєфність та інше.

Незважаючи на те, що Unity не є традиційним інструментом для 3D моделювання, його можливості в інтеграції 3D асетів, анімації, освітлення, та візуалізації роблять його незамінним інструментом для розробників ігор та інтерактивних додатків. Unity дозволяє не просто імпортувати 3D моделі, але й оптимізувати їх для кращої продуктивності в іграх та додатках, надаючи інструменти для LOD (Level of Detail) управління, що є критично важливим для мобільних платформ та VR/AR застосунків.

Додатково, платформа має вбудовану підтримку для фізики, яка дозволяє реалізовувати реалістичну взаємодію об'єктів у віртуальному світі. Система фізики в Unity включає в себе розширені можливості для симуляції рідин, твердих тіл, тканин, а також комплексну взаємодію між об'єктами з різними фізичними властивостями [9].

Unity також відзначається своїми засобами для створення інтерактивних елементів і UI компонентів, що робить його ідеальним рішенням для розробки не тільки ігор, але й едукативних програм, архітектурних візуалізацій, і тренажерів. Інтерфейс користувача може бути легко налаштований та інтегрований з 3D сценами, забезпечуючи зручне взаємодію з користувачем.

Однією з найсильніших сторін Unity є його спільнота. Велика та активна база користувачів постійно ділиться навчальними матеріалами, асетами та плагінами через Unity Asset Store. Це забезпечує розробникам доступ до великої кількості ресурсів, які можуть значно прискорити процес розробки та підвищити якість кінцевого продукту.

Завдяки постійним оновленням та вдосконаленням, Unity продовжує залишатися на передовій технологій ігрової індустрії, пропонуючи розробникам гнучкі та потужні інструменти для реалізації їхніх найсміливіших ідей в області 3D моделювання та інтерактивної анімації.

Unreal Engine, розроблений компанією Epic Games, є одним із найпотужніших та найбільш використовуваних ігрових двигунів у світі. Його

можливості далеко виходять за рамки ігрової індустрії, знаходячи застосування в кіноіндустрії, архітектурі, автомобільному дизайні, віртуальній реальності (VR) та інших сферах, де потрібно високоякісне 3D візуалізування та інтерактивність.

Unreal Engine відомий своєю винятковою графічною якістю та високим рівнем реалізму. Завдяки передовому рендерингу, підтримці реалістичного освітлення через Ray Tracing та потужній системі візуальних ефектів, двигун дозволяє створювати захоплюючі візуальні сцени. Unreal Engine також пропонує великий набір інструментів для створення динамічних ефектів, таких як вогонь, дим, вода, що робить середовище гри ще більш занурюючим.

Unreal Engine надає інтегровані інструменти для імпортування, оптимізації та роботи з 3D моделями. Його сумісність із широким спектром форматів файлів дозволяє легко інтегрувати асети, створені в інших програмах моделювання, таких як Maya, 3Ds Max, або Blender. Unreal Engine пропонує розширені можливості для роботи з текстурами та матеріалами, дозволяючи створювати високодеталізовані поверхні та комплексні шейдери.

Двигун має потужну систему анімації, що дозволяє створювати складні анімаційні секвенції, контрольовані через Blueprints або C++ код. Система ригінгу та анімації персонажів дозволяє детально налаштовувати рухи, міміку та інші аспекти поведінки персонажів. Unreal Engine також пропонує інструменти для створення інтерактивних елементів геймплею, включаючи фізику, штучний інтелект, та систему відстеження подій.

Unreal Engine дозволяє розробникам створювати проекти для широкого спектру платформ, включаючи ПК, консолі, мобільні пристрої, VR та AR системи. Ця універсальність забезпечує, що проекти можуть досягати максимальної аудиторії без потреби в значних змінах коду або асетів для різних платформ. Крос-платформна розробка з Unreal Engine економить час і ресурси, одночасно збільшуючи доступність проекту.

Unreal Engine вирізняється своєю системою візуального програмування Blueprints, яка дозволяє розробникам та дизайнерам створювати складні логіки

і механіки без безпосереднього написання коду. Blueprints зробили процес розробки більш інтуїтивним та доступним для непрограмістів, водночас зберігаючи гнучкість і потужність для реалізації будь-яких задумів проекту.

Unreal Engine підтримується великою та активною спільнотою розробників, а також має велику базу навчальних матеріалів, документації, форумів та відеоуроків. Epic Games регулярно проводить вебінари та конференції для навчання та обміну досвідом серед розробників. Також Unreal Engine Marketplace пропонує велику кількість готових асетів, плагінів та інструментів, які можуть значно прискорити розробку та покращити якість проектів.

### 1.3 Висновки до першого розділу

Застосування Unity в сфері тривимірного моделювання ландшафту та архітектури реального світу розкриває безмежні можливості для архітекторів та дизайнерів. Використання цієї платформи, як детально описано в першому розділі кваліфікаційної роботи, надає професіоналам в області тривимірного моделювання потужний інструментарій для створення реалістичних і детальних візуалізацій ландшафтів та архітектурних об'єктів. Unity дозволяє не тільки виконувати базові маніпуляції з тривимірними моделями, але й працювати як з двовимірним анімаційним, так і з базовим відеоредактором, що розширює можливості візуалізації проектів.

Розгляд додаткових інструментів та вже розроблених моделей, які можуть бути інтегровані в Unity, вказує на стратегії, що дозволяють оптимізувати процес розробки та зосередитися на ключових аспектах дизайну. Такий підхід не тільки сприяє економії часу, але й забезпечує можливість звернення уваги на деталізацію та реалізм сцен без необхідності створення кожного елемента з нуля.

Аналіз анімаційних принципів та правил освітлення в контексті тривимірного моделювання стає ключовим для досягнення високої якості візуалізацій. Відповідне розташування об'єктів та налаштування освітлення дозволяють створювати реалістичні сцени, що є критично важливим для представлення архітектурних проєктів та ландшафтного дизайну.

Це дослідження підтверджує значущість застосування сучасних технологічних рішень, таких як Unity, у тривимірному моделюванні для архітектури та ландшафтного дизайну. Метою роботи є не лише демонстрація ефективності цих методів для створення візуалізацій, але й оптимізація робочого процесу через використання передових інструментів і підходів. Такий підхід дозволяє фахівцям в області архітектури та ландшафтного дизайну ефективно втілювати свої творчі ідеї та концепції в життя. Використання Unity сприяє створенню детальних та реалістичних тривимірних сцен, що дозволяє замовникам та іншим зацікавленим особам краще зрозуміти запропоновані архітектурні та ландшафтні проєкти до їх реалізації. Це, у свою чергу, відкриває нові перспективи для обговорення, корекції та удосконалення проєктів на ранніх етапах розробки.

## РОЗДІЛ 2. СТВОРЕННЯ ВІРТУАЛЬНИХ СЕРЕДОВИЩ НА ОСНОВІ ВІДКРИТИХ ГЕОПРОСТОРОВИХ ДАНИХ ТА ІГРОВОГО РУШІЯ UNITY.

### 2.1 3D-модельовання ландшафту

Тривимірні представлення реального світу на комп'ютерах та мобільних телефонах широко використовуються як у віртуальних глобусах, так і в іграх. Віртуальні глобуси, які включають різноманітні графічні комп'ютерні додатки, що відображають поверхню Землі (наприклад, Google Earth) та і відкриті програмні інтерфейси (API) з відомими назвами World Wind та Cesium, спочатку були запущені для можливості розповсюдження та візуалізації тривимірних геопросторових даних. Після їхнього введення віртуальні глобуси відкрили багато можливостей для експертів та користувачів, щоб обмінюватися та поширювати тривимірні світові дані для багатьох областей застосування (наприклад, охорона навколишнього середовища та реагування на надзвичайні ситуації) і як платформа для співпраці.

Тривимірні представлення Землі в іграх спочатку використовувалися для створення реалістичних сцен, але розвиток комп'ютерних ігрових рушіїв дозволив розробникам використовувати їх як багатоплатформенний інструмент для створення інтерактивних візуалізацій.

Порівняно з віртуальними глобусами, ігрові рушії відкривають більше можливостей для збагачення взаємодій та вражень користувачів. Однією з ключових вбудованих функцій ігрових рушіїв є можливість перегляду від першої особи. Це дозволяє користувачам спостерігати за віртуальним світом з певної перспективи, що відрізняється від їхнього фактичного місця розташування. Це також надає можливість контролювати рух та виконувати різноманітні дії та взаємодіяти з різними об'єктами, створюючи інтенсивні враження від взаємодії з віртуальним середовищем [11].

Розробники вже давно використовують ігрові рушії для візуалізації віртуальних світів, доступних для дослідження гравцями. Однак використання реальних геопросторових даних у гральних рушіях залишається завданням з високим ступенем складності, оскільки більшість гральних рушіїв обмежуються у підтримці геопросторових інформацій.

Гральні рушії давно відомі своєю можливістю надавати користувачеві враження, схожі на реальні ситуації. Серйозні ігри, тобто ігри, які мають попит не лише для розваг, але й для інших цілей, використовуються в різних сферах, таких як освіта, симуляція та навчання на основі ігор. Прикладами є ігри для навчання в умовах надзвичайних ситуацій, військові симуляції, візуалізації міст та різноманітні інші застосування. Стратегії гейміфікації використовуються для розробки ігрових додатків із серйозною метою, такою як збір даних та розповсюдження інформації для охорони навколишнього середовища.

Розглянемо Unity3D для візуалізації великомасштабних топографічних даних з різних джерел моделей наземних лазерних сканерів та даних топографічних карт.

## 2.2 Методи обробки топографічних даних з Unity.

Одним з методів створення 3D в Unity є використання MapBox.

Mapbox є потужним інструментом для розробників, який дозволяє інтегрувати карти та геопросторові дані в додатки та ігри. Коли стверджується про використання Mapbox з Unity, це означає створення інтерактивних і геопросторових додатків, які використовують реальні дані про місцевість і навігацію в 3D ігровому середовищі. Unity - це популярний рушій для створення ігор, який дозволяє розробникам створювати складні і високоякісні ігри та інтерактивні дослідження (рис. 2.1).

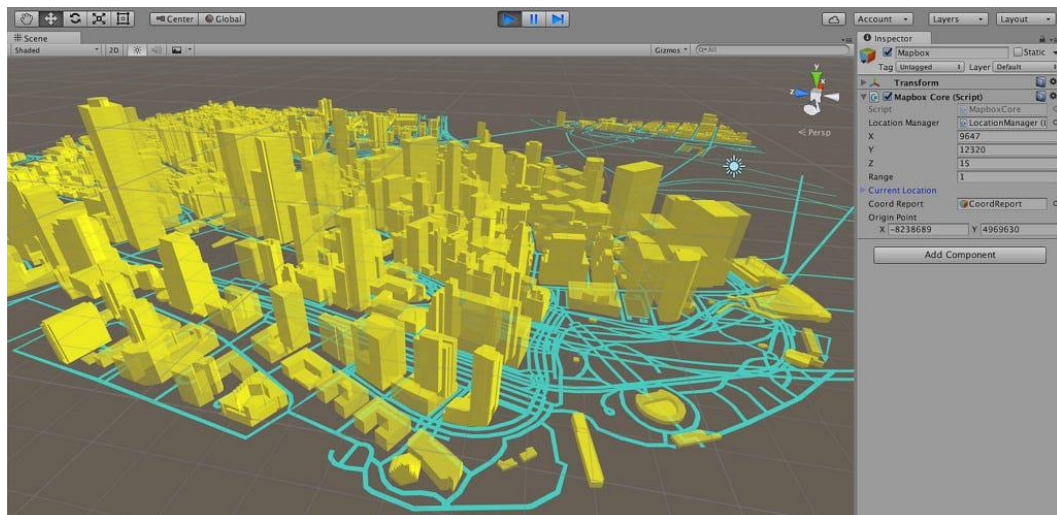


Рисунок 2.1 – Використання MapBox, для моделювання міста

Mapbox дозволяє розробникам інтегрувати реальні картографічні дані в ігри та інтерактивні додатки, створюючи реалістичні віртуальні середовища, засновані на реальних місцевостях. Розробники можуть налаштувати вигляд карт, вибираючи різні стилі та типи даних, що відображаються, включаючи дороги, будівлі, парки та інші об'єкти. Unity дозволяє розробникам створювати деталізовані 3D моделі місцевостей та об'єктів, використовуючи дані Mapbox для створення реалістичних сценаріїв та ігрових елементів. Також Mapbox та Unity разом дозволяють створювати інтерактивні картографічні досвіди, де користувачі можуть взаємодіяти з картами, змінювати масштаб, переглядати різні типи даних та виконувати інші дії.

Mapbox Maps SDK for Unity - це набір інструментів для побудови Unity-додатків з реальними картографічними даними. Він складається з потужного API для взаємодії з веб-сервісами Mapbox і перетворення ресурсів карт у гральні об'єкти, а також з графічного інтерфейсу користувача, побудованого на основі платформи Unity [12].

Основний функціонал Mapbox Maps SDK реалізовується в базовому класі AbstractMap (рис. 2.2).



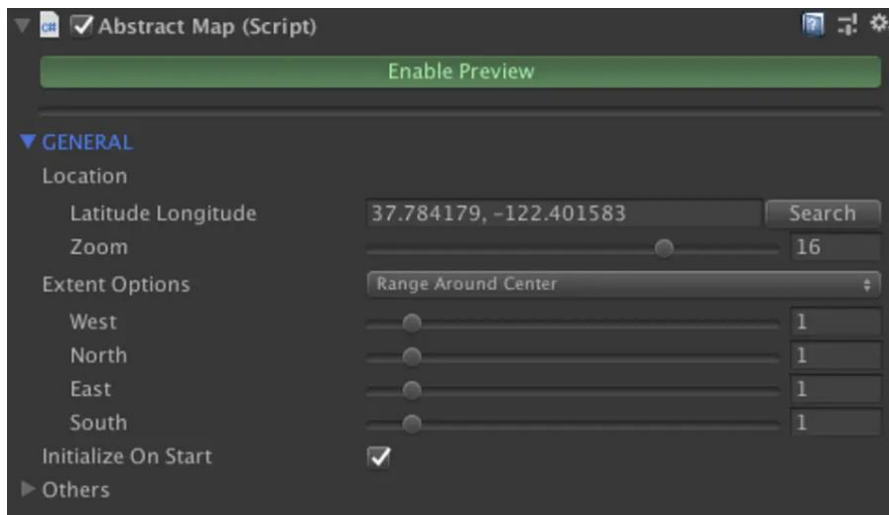


Рисунок 2.2 – Базовий клас AbstractMap

AbstractMap - це базовий клас, який відповідає за створення і оновлення карти в сцені. Він контролює візуалізацію картографічних даних. AbstractMap об'єднує зображення, рельєф і векторні джерела і надає централізований інтерфейс для керування візуалізацією карти. Його методи встановлюють центр карти за географічними координатами, корегують масштаб карти, встановлюють властивості для шару зображень карти та рельєфу [15].

Для роботи з картами бібліотека надає окремий графічний редактор в середині Unity - Map editor (рис. 2.3).

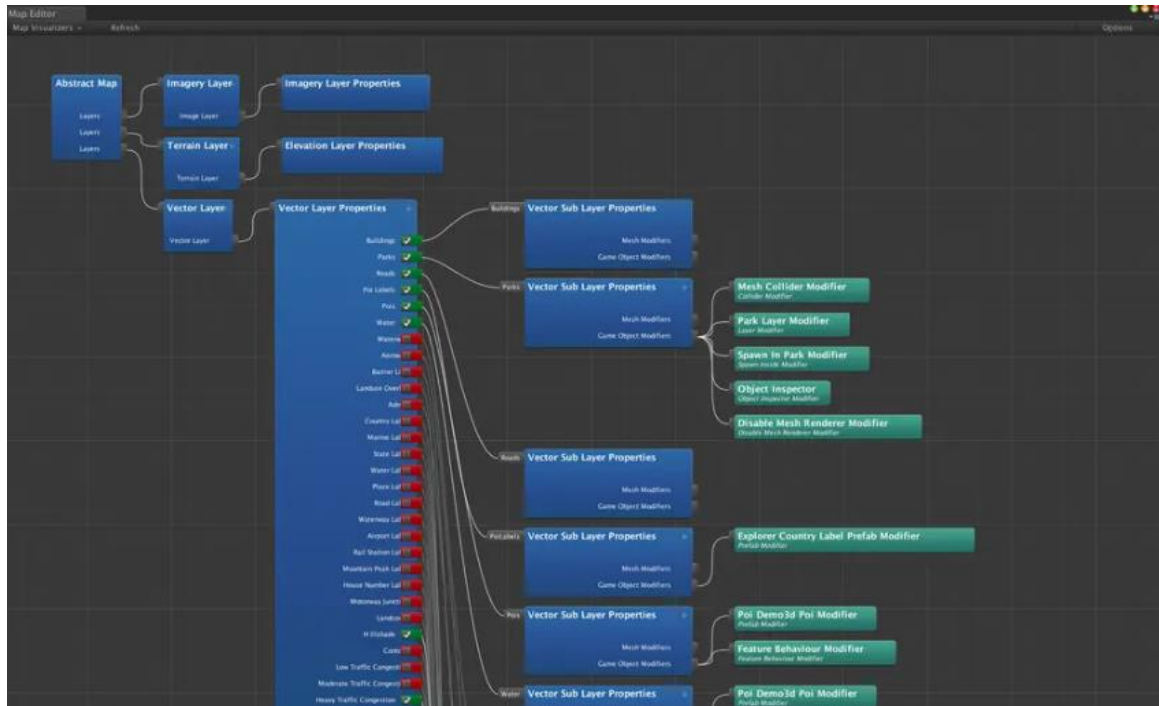


Рисунок 2.3 – Редактор Map editor

Map editor - це графічний інтерфейс користувача для проектування карти і її візуалізації в Unity, знаходиться в меню **Mapbox > Map Editor**. Він дозволяє створювати модульні, деревоподібні вузли, які представляють різні функції карти, такі як будівлі, супутникові зображення, геодані і інше. Даний редактор дозволяє з'єднувати вузли між собою і налаштовувати їх параметри в панелі Inspector. Map editor допомагає розуміти і керувати структурою карти

Серед прикладів можна розглянути ігри, подібні до Pokémon Go, Minecraft Earth (рис. 2.4) де геопросторові дані використовуються для створення ігрового світу, що взаємодіє з реальним світом гравця.



Рисунок 2.4 – Гра на основі геоданих Pokémon Go

Серед інших методів є використання OpenStreetMap. OpenStreetMap: OSM - це вільна та відкрита база геоданих, яку генерують та редагують користувачі спільноти. Дані у OSM можуть бути змінені та оновлені в реальному часі користувачами (рис. 2.5).

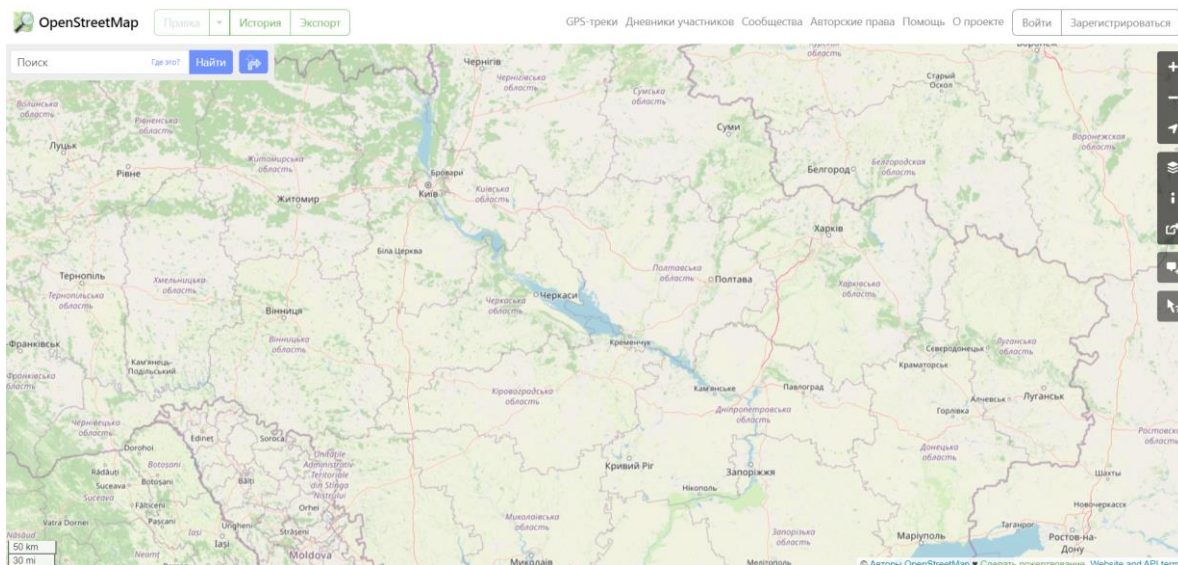


Рисунок 2.5 – OpenStreetMap

Для роботи з OSM в Unity, можна використовувати сторонні бібліотеки та інструменти, такі як Open Street Map Client. Open Street Map Client - це інструмент інтеграції, який дозволяє завантажувати і відображати карти OpenStreetMap в Unity. Він підтримує різні формати даних, такі як XML, PBF, GeoJSON і інші. Він також дозволяє редагувати карти в режимі реального часу (рис 2.6).

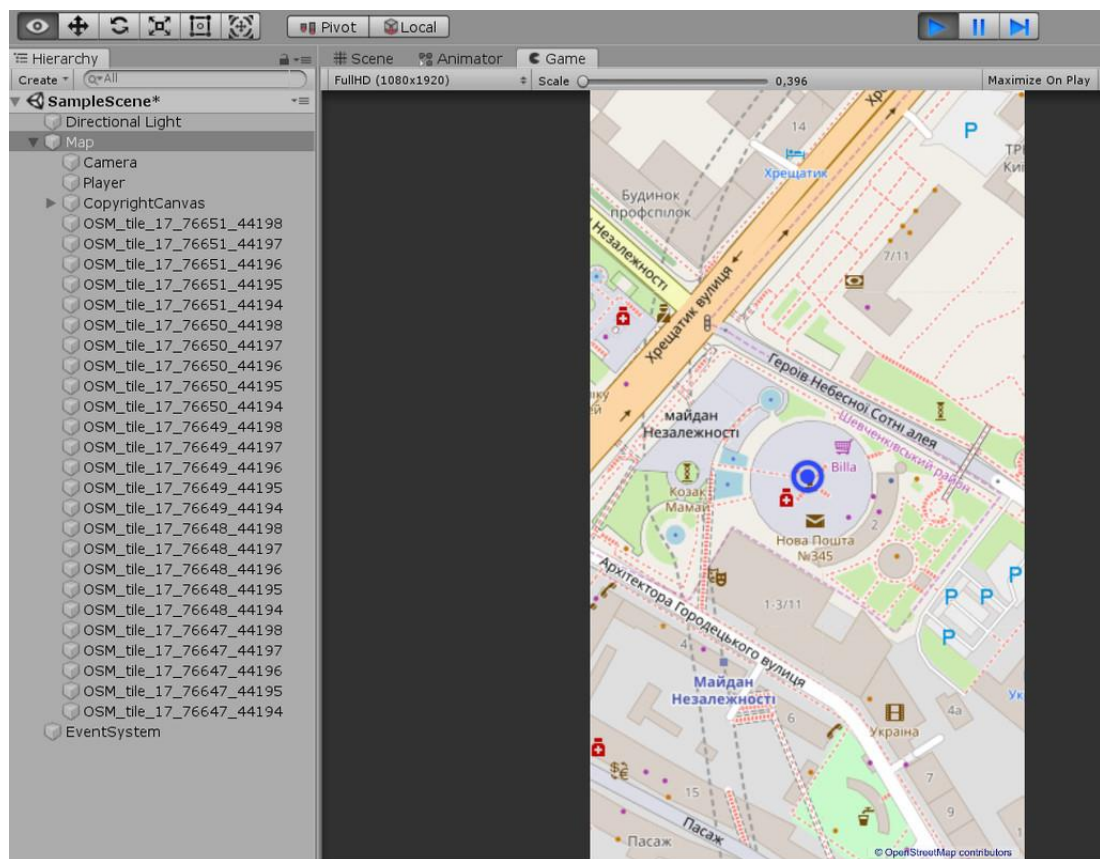


Рисунок 2.6 – Використання Overpass API в Unity

Якщо потрібно більше функціональності, може знадобитися використовувати додаткові дані або інші сервіси на відміну від MapBox [16, 17].

Дані OSM доступні за ліцензією Open Database License (ODbL), яка дозволяє використовувати та розповсюджувати дані, але із зазначенням джерела та збереженням вільності внесення змін. Серед інструментів які надає Open Street Map Client Unity можна виділити наступні:

- OSM Map - це компонент, який додається до грального об'єкта і відповідає за завантаження і відображення карти. Він має властивості, такі як OSM File Name, Map Style, Map Scale, Map Offset і інші, які дозволяють налаштувати параметри карти.

- OSM Editor - це редактор, який дозволяє редагувати карти в режимі реального часу. Він має панелі, такі як Map, Elements, Tags, Search і інші, які дозволяють додавати, видаляти і змінювати елементи карти, такі як точки, лінії, полігони, теги і інші.

- OSM Data - це клас, який зберігає інформацію про дані карти, такі як версія, границі, елементи і теги. Він має методи, такі як Load, Save, Clear, GetElement, GetTag і інші, які дозволяють працювати з даними карти.

Пакет Terrain Tools - це пакет, який покращує роботу з створення екосистем Terrain в Unity (рис. 2.7). Він додає більш розширені функції, такі як інструменти ерозії, скульптури і фільтри масок пензлів. Він також містить Terrain Toolbox, який дозволяє створювати нові Terrain з файлів пресетів.

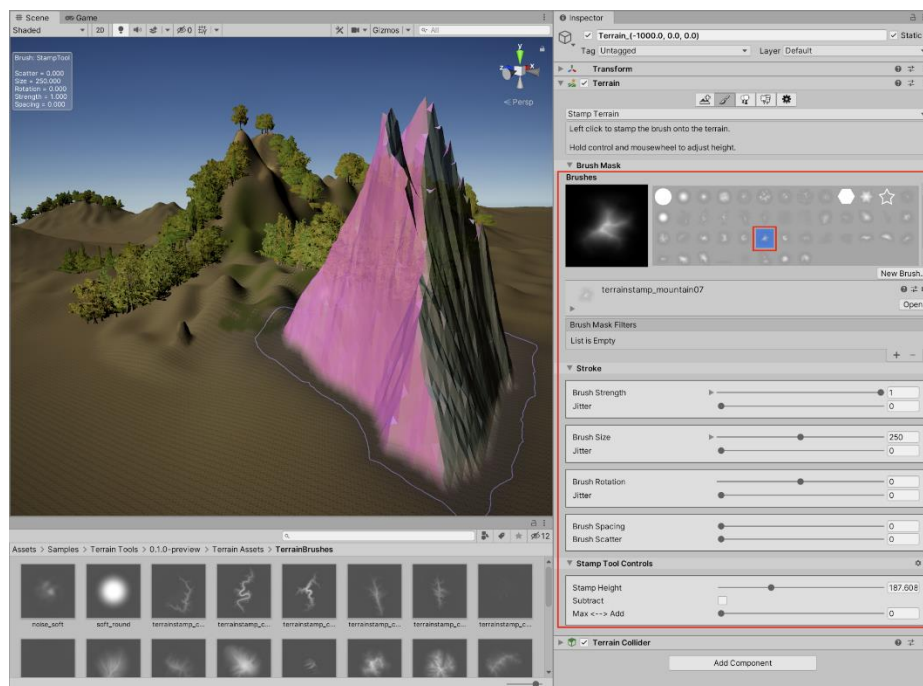


Рисунок 2.7 - Terrain Tools

Terrain Toolbox - це інструмент, який можна відкрити в редакторі Unity, який містить корисні інструменти для полегшення роботи з Terrain. Він дозволяє створювати нові Terrain з налаштувань пресетів або імпортованих висотних карт, змінювати налаштування на декількох плитках Terrain і імпортувати/експортувати рельєф і висотні карти за допомогою інструментів з Toolbox [18].

Terrain Toolbox має три інструменти: Create New Terrain, Terrain Settings і Terrain Utilities. Він також містить меню Terrain Visualization, яке дозволяє аналізувати Terrain (рис. 2.8).

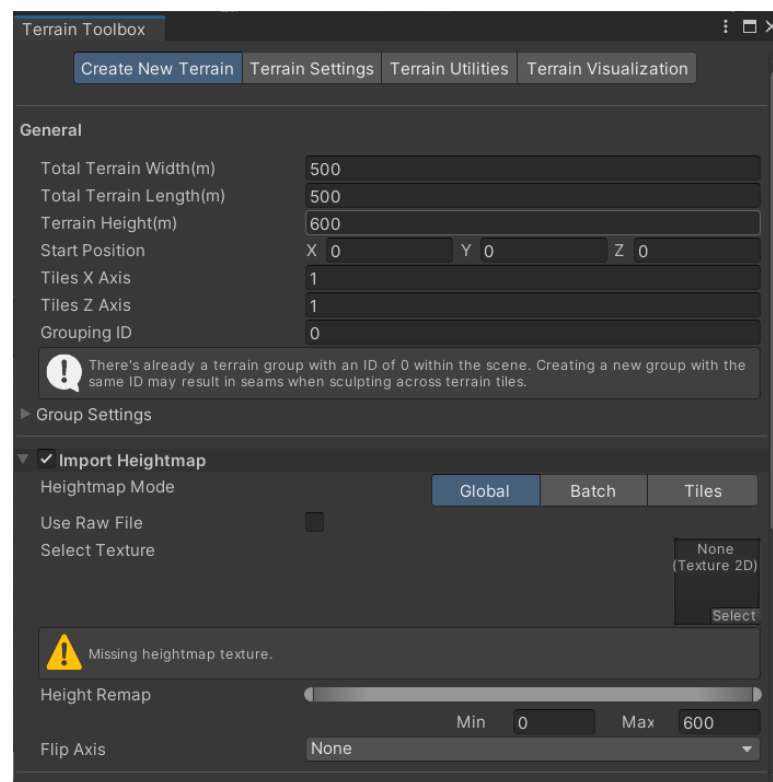


Рисунок 2.8 - Terrain Toolbox

Create New Terrain - це інструмент, який дозволяє створювати нові Terrain з файлів пресетів або імпортованих висотних карт. Тобто вибрати тип Terrain (одиничний або плитковий), розмір, роздільну здатність, висоту і інші параметри та вибрати джерело висотних карт (файл, шум, крива або інше) і налаштувати його опції (рис. 2.9).

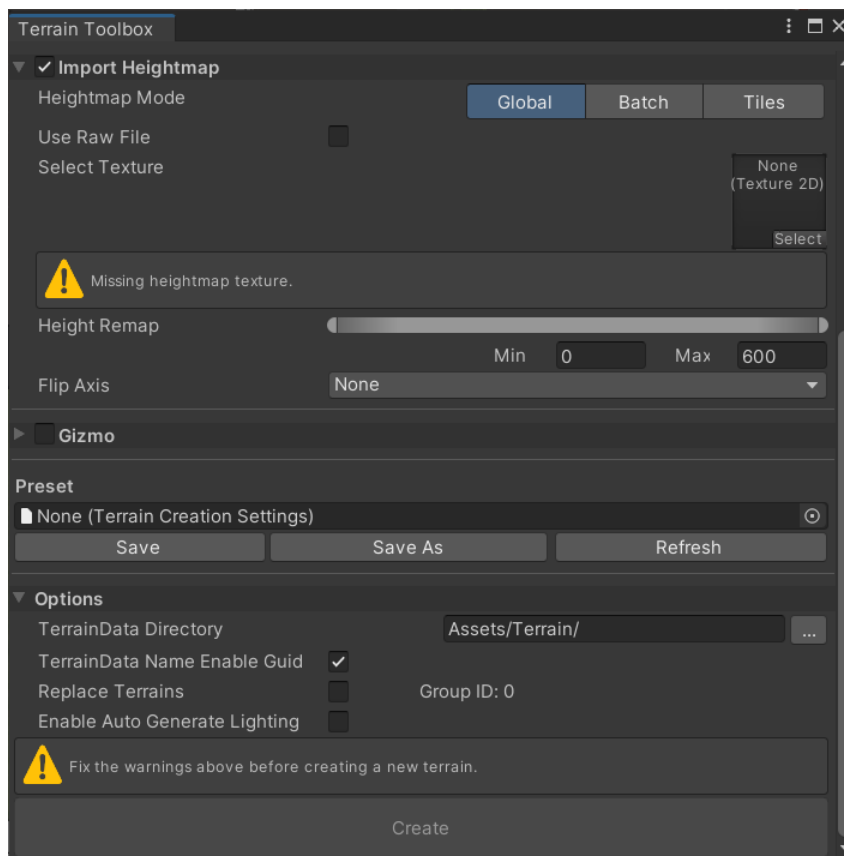


Рисунок 2.9 – Інструмент create new terrain

Terrain Settings - це інструмент, який дозволяє змінювати налаштування на декількох плитках Terrain (рис. 2.10).

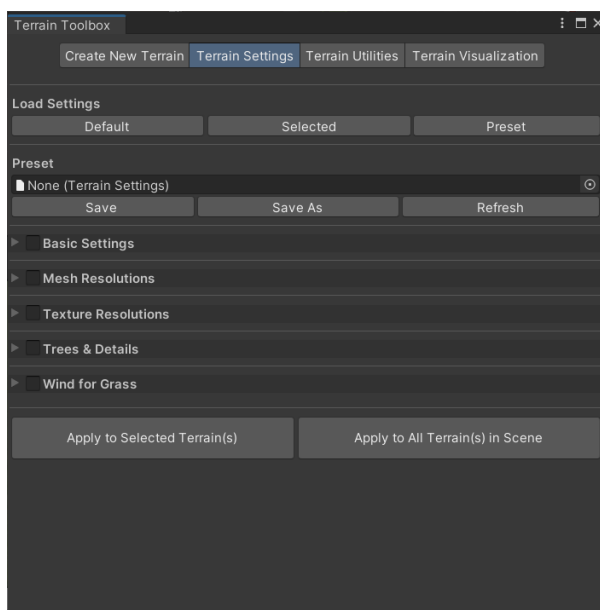


Рисунок 2.10 – Інструмент Terrain Settings

Terrain Utilities - це інструмент, який дозволяє імпортувати/експортувати висотні карти з. В Terrain Utilities можна вибрати плитки Terrain, які потрібно імпортувати/експортувати, і вказати шлях до файлів (рис. 2.11).

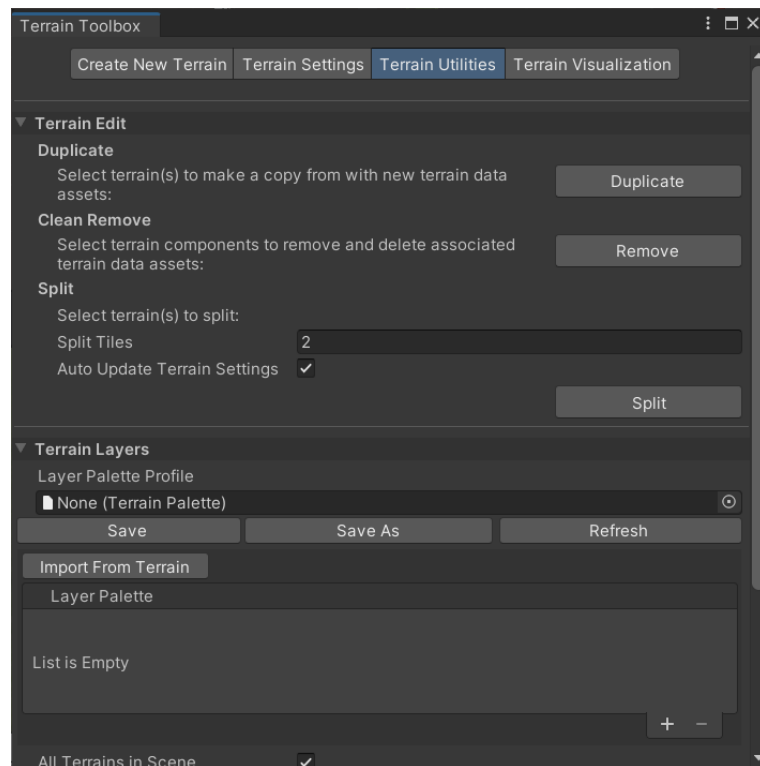


Рисунок 2.11 – Інструмент Terrain Utilities

### 2.3 Покращення методу моделювання

Моделювання геоданих доволі складне завдання, через те що впроваджувати реальні дані буває дуже складно через їх формат. Не всі розширення підтримує Unity: harefile, GeoJSON, KML та інші, що вимагає перетворення та адаптації для використання в Unity. для будування 3D Ландшафту.

Тому ефективним способ впроваджень, з оптимізувати роботу 3Dмоделювання таким чином, щоб якомога менше було зовнішніх інструментів для конвертації (рис 3.12).





Рисунок 2.12 – Кроки для роботи з геоданими

Скорочення кроків від завантаження до безпосередньо редагування самої моделі, може покращити швидкість роботи з геоданими для їх створення в 3D просторі.

Для оптимізації рішення, потрібно використовувати такі джерела, що надають можливість отримувати потрібний формат серед таких джерел можна використати джерело heightmap. Джерело геоданих дозволяє відразу завантажувати дані в потрібний для Unity формат, у вигляді фото розширення типу png (рис. 2.13).

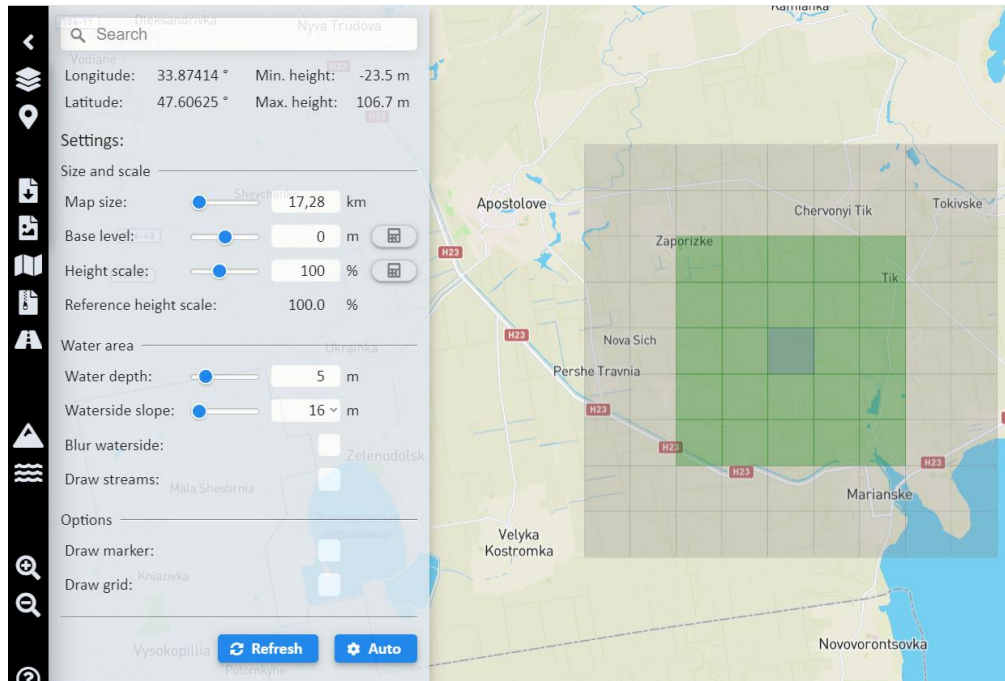


Рисунок 2.13 – Heightmap

Крім, вибору джерела даних, також потрібно обрати API, що надасть змогу без додаткових конвертацій отримувати ландшафт. Розумним рішенням є використання Terrain Tools. Дана базується на основних інструментах Unity, та лише доповнює їх, не додаючи при цьому нагромадження на проект та редактор. Таким чином можна скоротити кроків інтегрування карт в Unity (рис. 2.14).

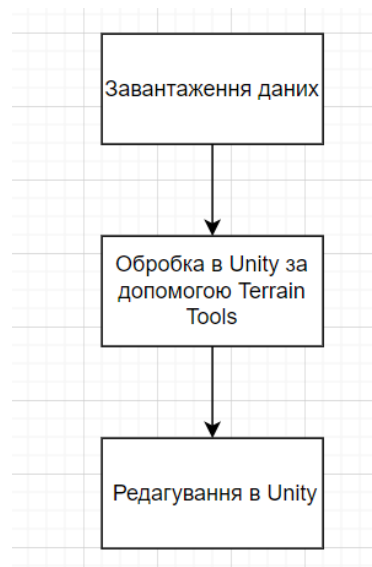


Рисунок 2.14 – Додавання карт за допомогою Terrain Tools

Можливим також покращенням, є інтегрування в реальному часі дані та додавати їх безпосередньо в програму, що дозволило обробляти ландшафт не тільки перед самим завантаженням проекту, а й під час її роботи.

## 2.4 Висновок розділу

Методи обробки 3D моделей карт та принципи їхнього моделювання стають ключовими у сучасних технологіях для віртуальних та розширених реальностей. 3D моделювання карт - це процес створення тривимірних візуальних представлень місцевості, будівель, об'єктів або інших сцен за допомогою спеціального програмного забезпечення. 3D моделювання карт має багато застосувань у різних галузях, таких як геодезія, географія, архітектура, інженерія, геологія, геоінформаційні системи, ігрова індустрія, кіно, анімація, освіта та інші. Обробка 3D моделей карт вимагає розуміння різних форматів геоданих. Забезпечення сумісності та перетворення даних з форматів, таких як Shapefile, GeoJSON чи KML, відкриває можливості для використання цих даних у віртуальних середовищах. Unity надає вбудовану систему ландшафту та можливості програмування шейдерів. Це дозволяє реалізувати деталізовані та реалістичні 3D моделі карт, враховуючи рельєф, текстури та ефекти світла.

## РОЗДІЛ 3. РОЗРОБКА 3D МОДЕЛІ ПОБУДОВИ ВОДОПОСТАЧАННЯ.

### 3.1 Актуальність розробки 3D моделювання завдання.

Актуальність розробки 3D моделі водопостачання виникає внаслідок зруйнування Каховської гідроелектростанції, що сталося в результаті руйнування її Російською Федерацією. Подібні події можуть призвести до серйозних проблем у системі водопостачання та вимагають невідкладних заходів для відновлення функціональності системи.

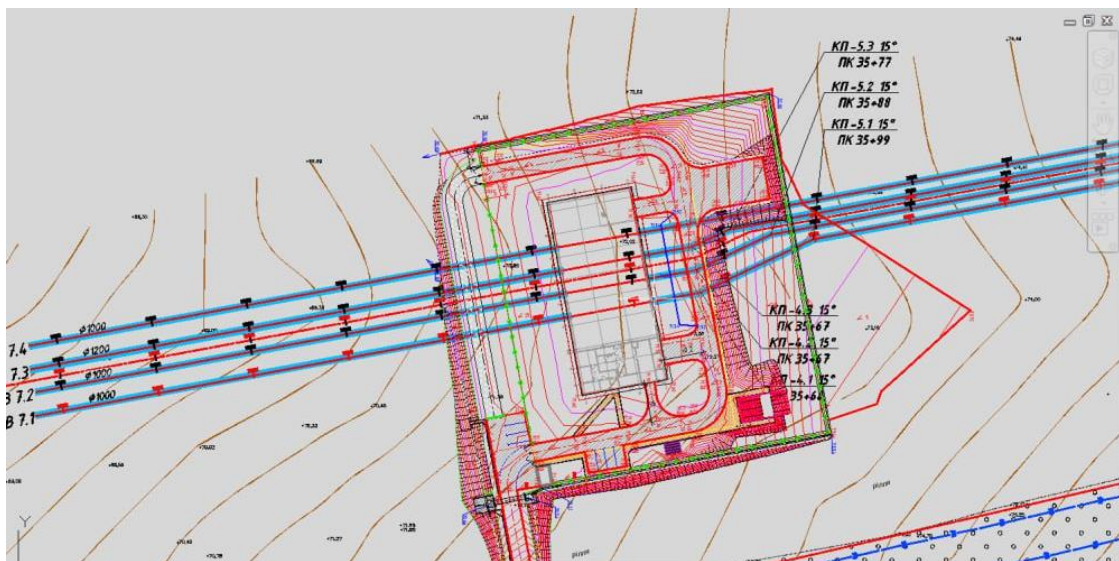


Рисунок 3.1– Прокладання магістралі

Зруйнування гідроелектростанції спричинило перерву у водопостачанні для прилеглих населених пунктів та промислових об'єктів. Розробка 3D моделі водопостачання стає критичною для ефективного відновлення інфраструктури, адаптації до нових умов та мінімізації можливих негативних наслідків для населення та економіки регіону.

Застосування 3D моделювання дозволить не лише швидко та точно відобразити пошкодження системи водопостачання, але і врахувати

різноманітні фактори, такі як територіальні особливості, географічні умови та розподіл населення, для оптимізації процесу відновлення та покращення стійкості системи в майбутньому.

Отже, розробка 3D моделі водопостачання після зруйнування Каховської гідроелектростанції є невід'ємною частиною стратегії відновлення інфраструктури та забезпечення надійного та ефективного водопостачання для населення та промислових підприємств в умовах непередбачуваних подій.

Переваги 3D моделювання у розробці системи водопостачання виявляються в надзвичайній можливості візуалізації та аналізу майбутньої інфраструктури. Спочатку, технології 3D моделювання дозволяють точно та реалістично відтворити місцевість, де буде розбудовуватися система водопостачання. Це дозволяє враховувати усі особливості ландшафту, географічні аспекти та просторові обмеження, що може впливати на оптимальний розміщення трубопроводів та інших інфраструктурних об'єктів (рис. 3.2).



Рисунок 3.2 – Створення 3D модель місцевості

Крім того, 3D моделювання дозволяє вирішити питання ефективності та безпеки майбутньої магістралі водопостачання. Вирішення цих питань на етапі віртуального моделювання дозволяє уникнути можливих проблем та недоліків на реальному об'єкті. За допомогою 3D моделювання можна провести умовне побудовування магістралі, оцінити її взаємодію з навколишнім середовищем, визначити оптимальний маршрут та врахувати фактори, такі як перешкоди та екологічні особливості, що сприяє покращенню проекту в цілому.

Такий підхід дозволяє максимально оптимізувати інженерний процес, зменшити витрати та час на реалізацію проекту, а також гарантує високу надійність та стійкість майбутньої системи водопостачання. Тим самим, 3D моделювання виявляється необхідним інструментом для ефективного та передбачуваного розвитку інфраструктурних проектів в сфері водопостачання.

### 3.2 Інструменти розробки.

Для моделювання 3D місцевості та взаємодії з ним використовується рушій Unity.

Інструмент розробки Unity володіє вражаючим набором можливостей для моделювання ландшафту, надаючи розробникам інструменти для створення реалістичних та візуально привабливих 3D середовищ. Unity дозволяє моделювати ландшафт з високою ступенем деталізації, використовуючи вбудовані та сторонні інструменти.

Для моделювання ландшафту в Unity зазвичай використовують інструменти, такі як Terrain, який надає можливість створення рельєфу з високою точністю, і Terrain Composer, що спрощує процес та робить його

більш інтуїтивним. Крім того, можна використовувати текстури та матеріали для надання поверхні ландшафту реалістичного вигляду.

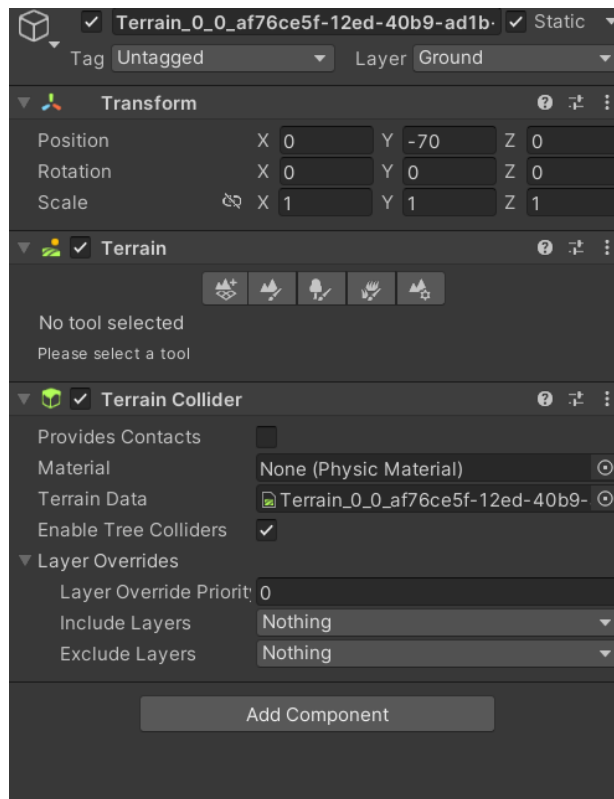
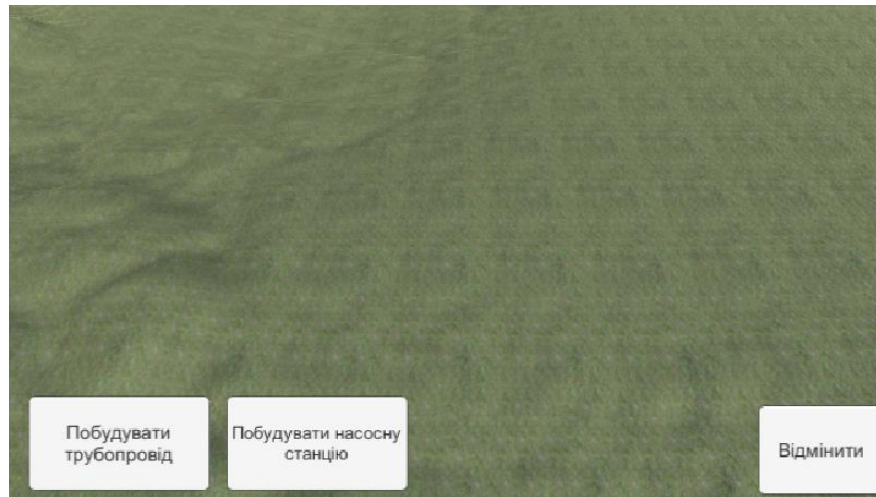


Рисунок 3.3 – Інструмент Terrain

Unity також підтримує використання шейдерів, що дозволяє створювати різні ефекти, такі як реалістичне водосховище, трава та інші деталі [7].

Щодо інтерактивного планування побудови водопостачання, використання Unity є доцільним завдяки його здатності створювати інтерактивні 3D моделі та візуалізації. Інтерактивність грає ключову роль у плануванні, оскільки дозволяє розробникам та планувальникам взаємодіяти з моделлю в реальному часі. Це відзначається в створенні користувацького інтерфейсу, інтерактивних елементів, можливість будувати споруди на 3D ландшафті (рис. 3.4).



Рисунку 3.4 – Користувацький інтерфейс

Застосування Unity для реалізації інтерактивного планування побудови водопостачання дозволяє створити інтерактивні стратегії для вирішення різних викликів. Рушій Unity забезпечує можливість взаємодії з об'єктами, сценаріями та симуляцією в режимі реального часу. Використання цього рушія дозволяє розробникам та планувальникам ефективно моделювати та тестувати різні стратегії, що сприяє оптимізації планування та уникненню можливих проблем у реальних умовах побудови водопостачальних об'єктів.

Також потрібно звернути увагу на використання в Unity Nvidia PhysX. Nvidia PhysX і Unity - це два потужні інструменти для розробки ігор з фізичними симуляціями. Nvidia PhysX - це відкритий рушій реальної фізики, який дозволяє створювати реалістичні та динамічні ефекти. Unity інтегрує PhysX як свій вбудований 3D-фізичний рушій. Це дозволяє розробникам легко використовувати фізичні можливості PhysX в своїх проектах Unity. Також існує експериментальний пакет Unity, який дозволяє отримати доступ до Nvidia PhysX SDK 4 зсередини Unity. Це надає розробникам більше контролю та гнучкості над фізичними симуляціями. Nvidia PhysX і Unity - це сильна комбінація для створення захоплюючих ігор з високоякісною фізикою. Nvidia PhysX надає методи та події для обробки зіткнень між об'єктами, такі як `OnCollisionEnter`, `OnCollisionStay`, `OnCollisionExit`. Основні аспекти використання Nvidia PhysX в Unity включають використання компоненту



Rigidbody використовуються для надання об'єктам маси, фізичних властивостей та взаємодії.

Для створення інтерактивних елементів, їх 3D моделей використовувався інструмент MagicaVoxel (рис. 3.5).

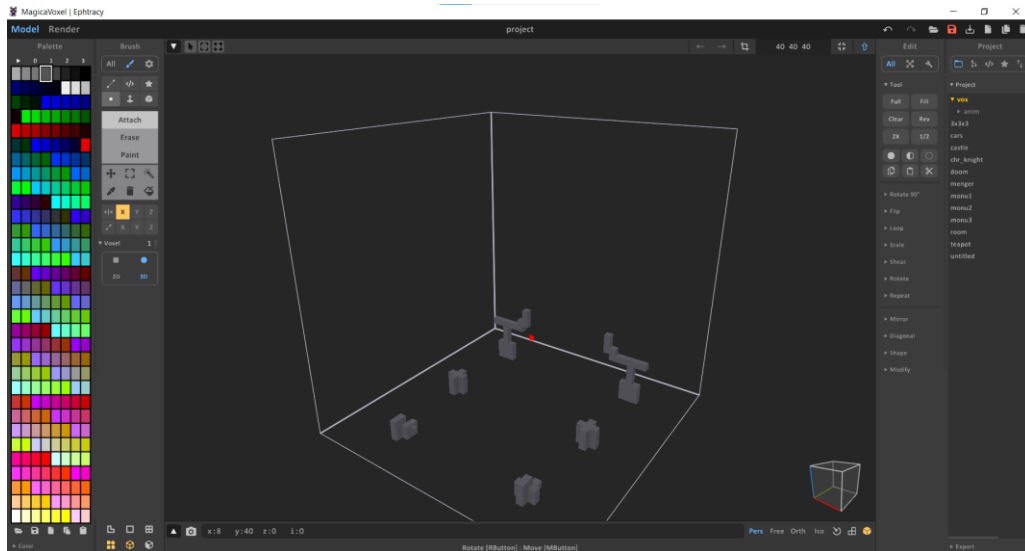


Рисунок 3.5 – Інтерфейс інструменту MagicaVoxel

MagicaVoxel – це популярний інструмент для створення 3D моделей за допомогою простою воксельної графіки, який виділяється своєю простотою використання та потужними можливостями. Особливість цього редактора полягає в його спрощеному інтерфейсі, що дозволяє навіть початківцям швидко освоювати процес моделювання.

MagicaVoxel дозволяє користувачам легко створювати візуально захоплюючі 3D об'єкти в стилі voxel art. Цей редактор оптимізований для створення блочних моделей, де кожен елемент визначається окремим вокселем. Він підтримує різні інструменти для моделювання, текстур і кольорів, що робить його ефективним інструментом для розробки невеликих та середніх 3D сцен.

Більш того, MagicaVoxel підтримує експорт у різні формати файлів, що спрощує процес інтеграції створених моделей в різноманітні програми та ігрові рушії, такі як Unity. Його популярність серед розробників графіки та

геймдевелоперів обумовлена його ефективністю, інтуїтивністю та можливістю швидко створювати вражаючі 3D об'єкти.

Таож варто враховувати мову програмування C#. Мова програмування C# є однією з ключових мов для розробки в середовищі Unity, і її значення в цьому контексті можна розглядати з різних точок зору.

C# є офіційною мовою програмування для Unity, і це означає високий рівень інтеграції з Unity API. Розробники можуть звертатися до всіх можливостей Unity, таких як робота з об'єктами сцени, фізикою, анімацією, графікою та іншими, за допомогою C#.

C# є об'єктно-орієнтованою мовою програмування (ООП), що дозволяє розробникам використовувати принципи ООП для створення більш чистого та структурованого коду. Це особливо важливо при розробці складних ігор та додатків [8].

Використання C# у Unity відкриває доступ до широкого спектру бібліотек та інструментів, які можна використовувати для покращення функціональності та продуктивності розробки.

### 3.3 Розробка додатку.

Для роботи додатку використовуються наступні класи проекту:

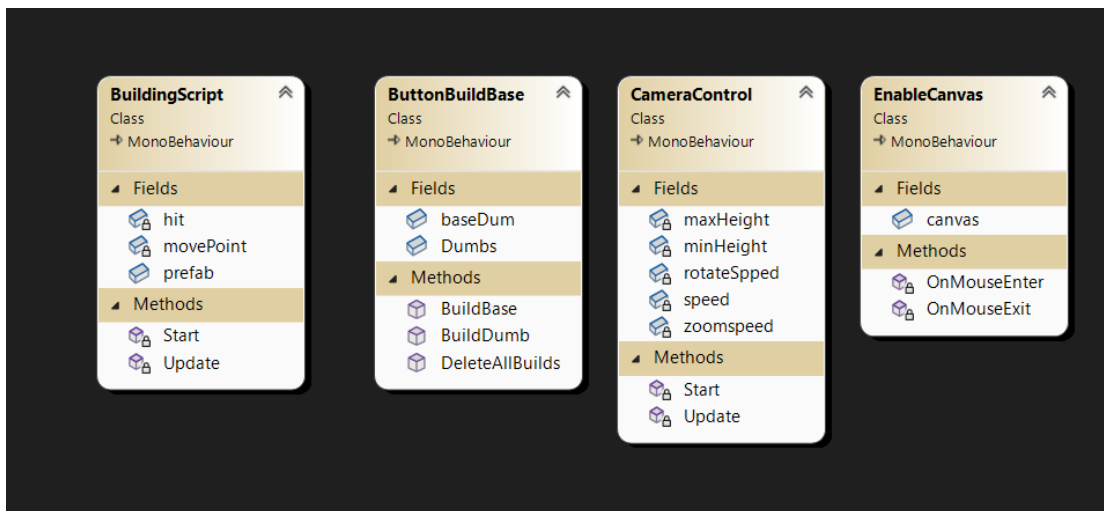


Рисунок 3.6 – Діаграма класів

Розглянемо кожен з них. Перший клас пов'язаний з управлінням камери CameraControl.

```

public class CameraControl : MonoBehaviour
{
    // Start is called before the first frame update
    float speed = 0.06f;
    float zoomspeed = 10.0f;
    float rotateSpped;

    float maxHeight = 120f;
    float minHeight = 1f;

    @ Unity Message | 0 references
    void Start()
    {
    }

    // Update is called once per frame
    @ Unity Message | 0 references
    void Update()
    {
        if(Input.GetKey(KeyCode.LeftShift))
        {
            speed = 0.06f;
            zoomspeed = 20.0f;
        }
        else
        {
            speed = 0.035f;
            zoomspeed = 10.0f;
        }

        float hsp =transform.position.y * speed * Input.GetAxis("Horizontal");
        float vsp =transform.position.y * speed * Input.GetAxis("Vertical");
        float scrolSp =Mathf.Log(transform.position.y) * -zoomspeed * Input.GetAxis("Mouse ScrollWheel");

        if((transform.position.y >= maxHeight) && (scrolSp >0))
        {
            scrolSp = 0;
        }
        else if((transform.position.y <= minHeight) && (scrolSp<0))
        {
            scrolSp = 0;
        }
    }
}

```

Рисунок 3.7 – Клас CameraControl

Даний код представляє клас CameraControl, який відповідає за управління камерою в ігровому середовищі з використанням Unity. Він містить змінні для налаштування швидкості руху, зуму та обертання камери (рис. 3.8).

```

Unity Script (1 asset reference) | 0 references
public class CameraControl : MonoBehaviour
{
    // Start is called before the first frame update
    float speed = 0.06f;
    float zoomspeed = 10.0f;
    float rotateSpded;

    float maxHeight = 120f;
    float minHeight = 1f;

    Unity Message | 0 references
    void Start()
    {

```

Рисунок 3.8 – Змінні для налаштування камери

Клас обробляє події вводу, такі як натискання клавіші Shift, і відповідно змінює швидкості руху та зуму.

У методі Update виконується перевірка, чи натиснута клавіша Shift, та налаштовуються швидкості в залежності від цього. Далі розраховуються величини для руху по горизонталі, вертикалі та зуму камери на основі введення користувача клавіш (рис. 3.9).

```

void Update()
{
    if(Input.GetKey(KeyCode.LeftShift))
    {
        speed = 0.06f;
        zoomspeed = 20.0f;
    }
    else
    {
        speed = 0.035f;
        zoomspeed = 10.0f;
    }

    float hsp =transform.position.y * speed * Input.GetAxis("Horizontal");
    float vsp =transform.position.y * speed * Input.GetAxis("Vertical");
    float scrolSp =Mathf.Log(transform.position.y) * -zoomspeed * Input.GetAxis("Mouse ScrollWheel");

    if((transform.position.y >= maxHeight) && (scrolSp >0))
    {
        scrolSp = 0;
    }
    else if((transform.position.y <= minHeight) && (scrolSp <0))
    {
        scrolSp = 0;
    }

    Vector3 verticalMove = new Vector3(0, scrolSp, 0);
    Vector3 lateralMove = hsp * transform.right;
    Vector3 forwardMove = transform.forward;
    forwardMove.y = 0;
    forwardMove.Normalize();
    forwardMove *= vsp;

    Vector3 move = verticalMove + lateralMove + forwardMove;

    transform.position += move;
}

```

Рисунок 3.9 – Метод Update

Код також містить перевірки максимальної та мінімальної висоти камери, і якщо ці умови виконані, відсікається можливість додаткового зуму вгору або вниз.

Зміщення камери обчислюється на основі величин руху по різних координатних осях, і камера зміщується відповідно до цих значень в кожен кадр гри.

Наступним класом є BuildingScript.

```

public class BuildingScript : MonoBehaviour
{
    RaycastHit hit;
    Vector3 movePoint;
    public GameObject prefab;
    // Start is called before the first frame update
    Unity Message | 0 references
    void Start()
    {
        Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
        Debug.Log(Physics.Raycast(ray, out hit, 50000.0f, (1 << 8)));

        if (Physics.Raycast(ray, out hit, 50000.0f, (1<<8)))
        {
            transform.position = hit.point;
        }
    }

    // Update is called once per frame
    Unity Message | 0 references
    void Update()
    {
    }
}

```

Рисунок 3.10 – Клас BuildingScript

Клас BuildingScript відповідає за взаємодію з об'єктами, які можна будувати в ігровому середовищі використовуючи Unity. У цьому випадку, клас реагує на введення користувача та реалізує можливість розміщення об'єкта будівництва в ігровому просторі.

У методі Start при старті гри використовується Ray для визначення позиції, де користувач хоче розмістити будівлю. Використовується Physics.Raycast, щоб перевірити, чи промінь стикається з об'єктом на певному шарі (в даному випадку, шар з індексом 8). Якщо умова виконується, то позначена позиція об'єкта будівництва встановлюється на місці зіткнення з об'єктом (рис. 3.11).

```

Unity Message | 0 references
void Start()
{
    Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
    Debug.Log(Physics.Raycast(ray, out hit, 50000.0f, (1 << 8)));

    if (Physics.Raycast(ray, out hit, 50000.0f, (1<<8)))
    {
        transform.position = hit.point;
    }
}

```

Рисунок 3.11 – Метод Start

У методі Update відслідковується натискання лівої кнопки миші. Якщо це відбувається, то створюється новий об'єкт (інстанція) на поточній позиції будівництва, використовуючи вказаний шаблон, та вихідний об'єкт (тобто, той, який має цей скрипт) знищується (рис. 3.12).

```

Unity Message | 0 references
void Update()
{
    Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
    if(Physics.Raycast(ray,out hit, 50000.0f, (1 << 8)))
    {
        transform.position = hit.point;
    }
    if(Input.GetMouseButtonDown(0))
    {
        Instantiate(prefab, transform.position, transform.rotation);
        Destroy(gameObject);
    }
}

```

Рисунок 3.12 – Метод Update

Відслідковувати взаємодію та розміщення об'єктів забезпечується завдяки методу Physics.Raycast. `Physics.Raycast` є методом в Unity, який використовується для визначення, чи перетинає луч (ray) об'єкт у 3D просторі гри. Цей метод може бути використаний для взаємодії з об'єктами в грі, такими як визначення точки зіткнення, отримання інформації про зіткнений об'єкт та інше.

Основні параметри Physics.Raycast:

- Ray: промінь, який визначає напрямок та початкову позицію для перевірки зіткнення.
- HitInfo: Об'єкт RaycastHit, який зберігає інформацію про зіткнення, таку як точку зіткнення, нормаль до поверхні, об'єкт, з яким відбулося зіткнення і т.д.
- MaxDistance: Максимальна відстань, на яку буде виявлено зіткнення.
- LayerMask: Вказує на якому шарі об'єктів буде виявлено зіткнення.

Результат виклику `Physics.Raycast` - це булеве значення, яке підтверджує чи не відбулося зіткнення. Якщо луч перетинає який-небудь об'єкт на вказаному шарі в межах максимальної відстані, результат буде `true`, інакше - `false`.

В контексті ігрового програмування цей метод часто використовується для реалізації взаємодії гравця з об'єктами, такими як визначення місця кліку мишею, реалізації системи вибору об'єктів та інших сценаріїв, де необхідно виявляти зіткнення променю з об'єктами в грі.

Наступний клас `ButtonBuildBase` виконує функції управління будівництвом та видаленням об'єктів в ігровому середовищі використовуючи. Нижче подано короткий опис методів цього класу (рис. 3.13):

```

Unity Script (1 asset reference) | 0 references
public class ButtonBuildBase : MonoBehaviour
{
    public GameObject baseDum;
    public GameObject Dumbs;

    0 references
    public void BuildBase()
    {
        Instantiate(baseDum);
    }

    0 references
    public void BuildDumb()
    {
        Instantiate(Dumbs);
    }

    0 references
    public void DeleteAllBuilds()
    {
        GameObject[] list = GameObject.FindGameObjectsWithTag("BuildPrint");
        foreach (GameObject go in list)
        {
            Destroy(go);
        }
    }
}

```

Рисунок 3.13 – Клас `ButtonBuildBase`

- `BuildBase()`: Цей метод відповідає за створення (інстанціювання) об'єкту `baseDum` у грі за допомогою функції `Instantiate`.
- `BuildDumb()`: Цей метод аналогічний `BuildBase()`, але створює об'єкт `Dumbs`.
- `DeleteAllBuilds()`: Цей метод видаляє всі об'єкти, які мають тег `"BuildPrint"` з гри. Він використовує `GameObject.FindGameObjectsWithTag("BuildPrint")`, щоб отримати масив усіх об'єктів з вказаним тегом, а потім використовує цикл `foreach` для знищення кожного об'єкту в масиві за допомогою `Destroy(go)`.

Цей клас використовується для реалізації кнопок управління будівництвом та видаленням об'єктів у грі, і він має доступ до об'єктів `baseDum` та `Dumbs`, які призначені для будівництва в грі.

Функціонал даного класу залежить від типів `GameObject`. В мові програмування Unity, змінна типу `'GameObject'` представляє собою посилання на об'єкт у грі. Об'єкти в Unity - це основні будівельні блоки гри, які можуть включати в себе гравців, об'єкти оточення, світло, камери та інше.

Змінна типу `GameObject` дозволяє програмісту отримувати доступ та взаємодіяти з різними об'єктами у грі. Це може бути корисно при розробці скриптів для управління об'єктами, такими як створення, знищення, переміщення, отримання доступу до їхніх компонентів та інше.

Наприклад, в коді Unity можна використовувати змінну `GameObject` для зберігання посилання на об'єкт будь-якого типу та взаємодії з ним через різноманітні методи та властивості, що надає Unity API.

Ігрові об'єкти в Unity можна створювати на різні способи, і основними методами є використання інтерфейсу Unity Editor та динамічне створення об'єктів через скрипти. В основному в Unity об'єкти створюються з допомогою редактора (рис. 3.14):



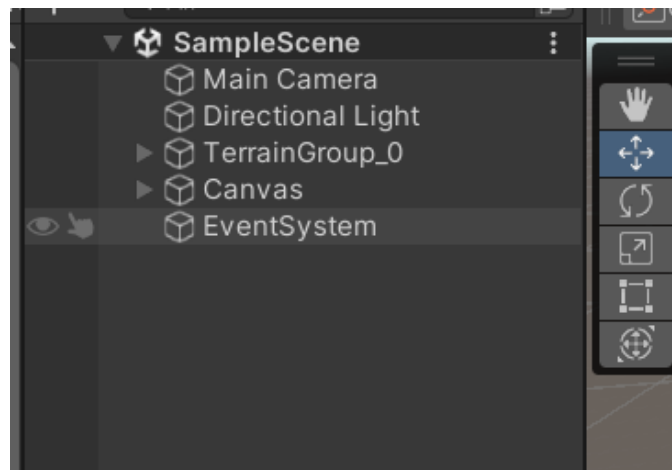


Рисунок 3.14 – Створення об'єкту в Unity

Для створення ландшафтів використовується інструмент Terrain ToolBox. Даний компонент дозволяє створювати різноманітні ландшафти враховуючи яку текстуру надає розробник (рис. 3.15):

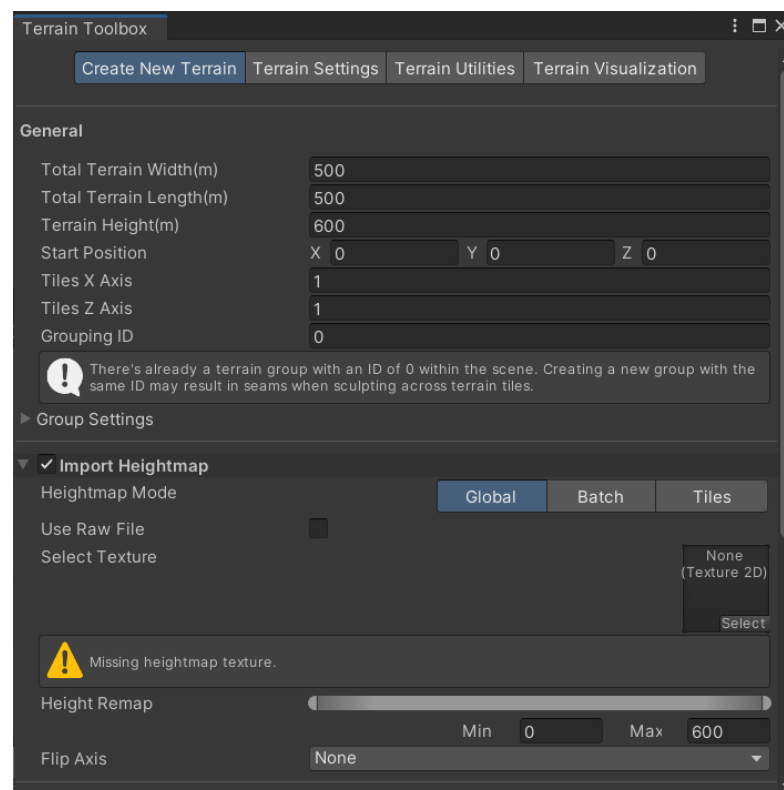


Рисунок 3.15 – Інструмент Terrain ToolBox

Відповідно, щоб створити ландшафт використовується відповідний шаблон, це може бути файл формату png, jpg.

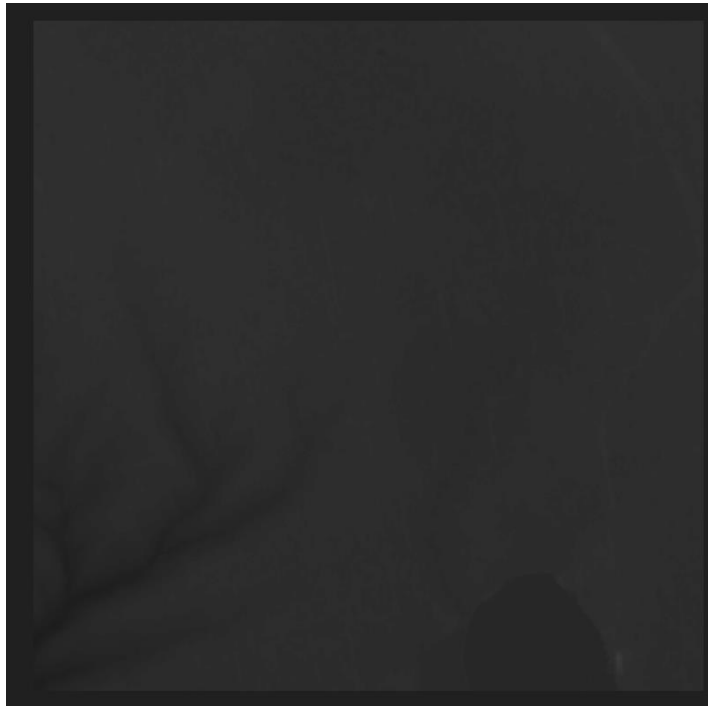


Рисунок 3.16 – Шаблон ландшафту.

Щоб отримати такий ландшафт потрібно використати відповідний ресурс: <https://heightmap.skydark.pl/>

Даний сайт надає можливість у будь-якій точці світу отримати шаблон ландшафту для побудови у вигляді 3D моделі.

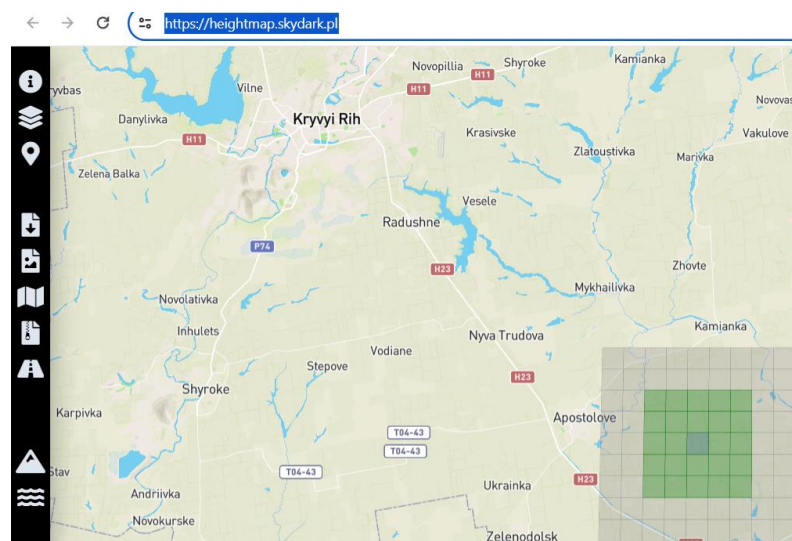


Рисунок 3.17 – Вибір відрізка ландшафту

Після обрання шаблону на сцені створюється 3D ландшафт:

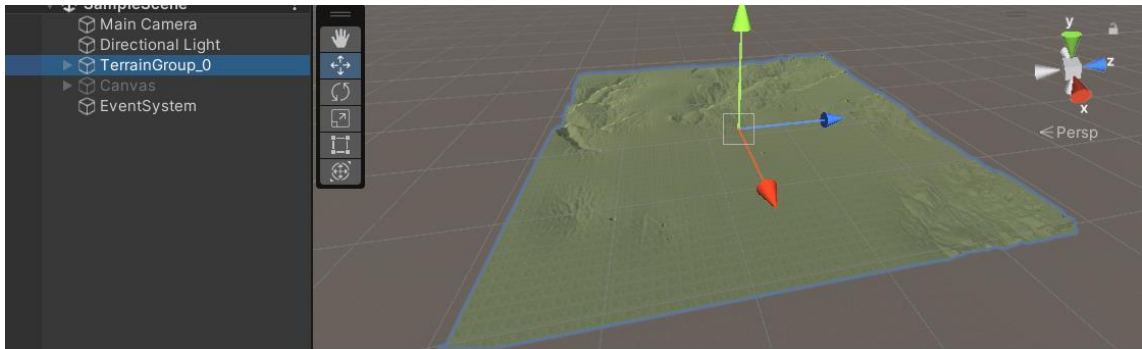


Рисунок 3.18 – Створення 3D ландшафту

Тепер з даним об'єктом можна повноцінно взаємодіяти, встановлювати різноманітні об'єкти завдяки вже зазначеному Physics.Raycast. Це працює завдяки компоненту Terrain Collider.

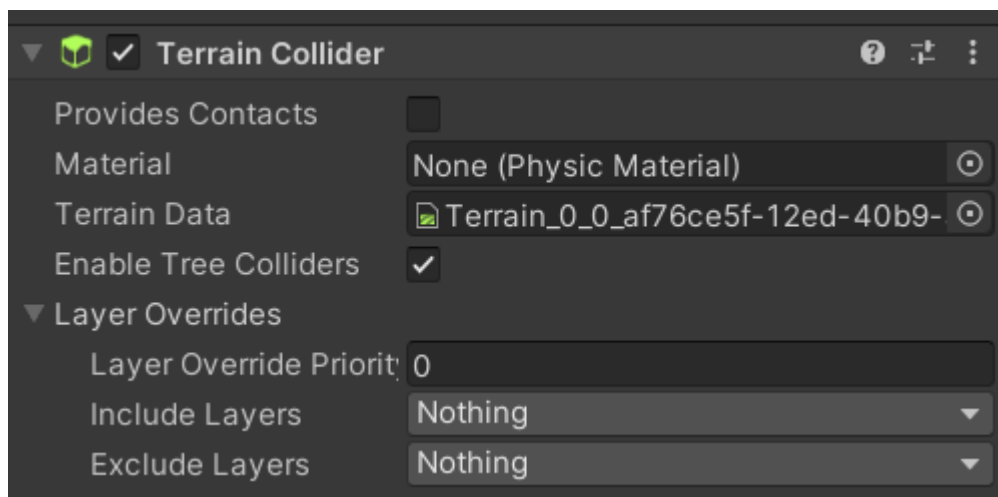


Рисунок 3.19 – Компонент Terrain Collider

Компонент Terrain Collider в Unity є важливим елементом для обробки колізій на ландшафті. Використовується в основному для оптимізації обробки фізичних взаємодій з рельєфом, що значно спрощує і покращує продуктивність в іграх, де присутні великі ландшафти.

Завдяки можливостям Unity можна корегувати ландшафт за допомогою різноманітних текстур, які можна завантажити з інтернету (рис. 20).

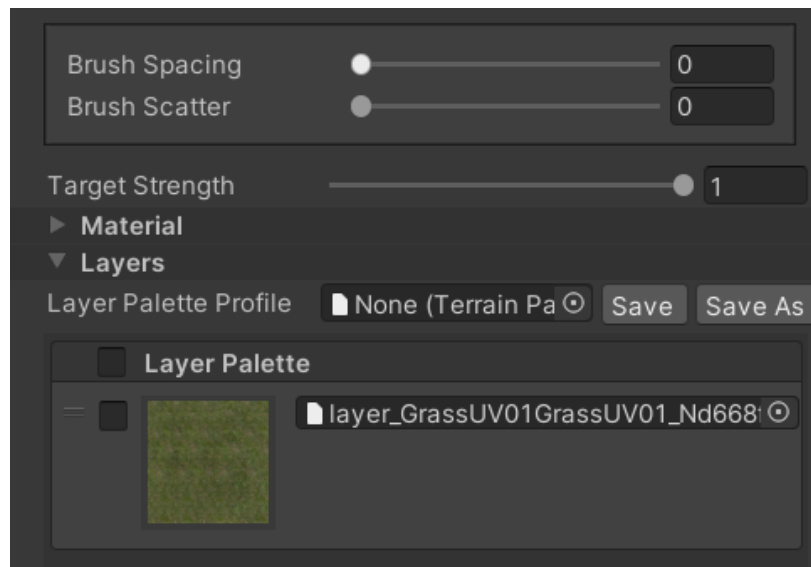


Рисунок 3.20 – Коригування ландшафту за допомогою текстур

Наступним етапом це створення ігрових об'єктів Unity, які містять в собі 3D текстури та скрипти для побудови будівлі на ландшафті. Всього 4 об'єкти розглянемо кожен з них (рис. 3.21).

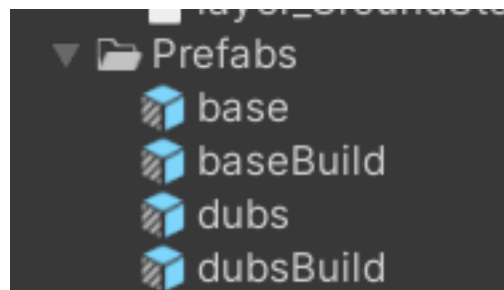


Рисунок 3.21 – Об'єкти шаблони

Дані шаблони використовуються для подальшого відтворення безпосередньо під час самої гри. Умовно дані шаблони поділені на 2 види це об'єкти з припискою Build, дані об'єкти мають лише певний силует будівлі яку потрібно побудувати та інший вид це самі будівлі.

Наприклад об'єкт baseBuild має наступний вигляд (рис. 3.22):

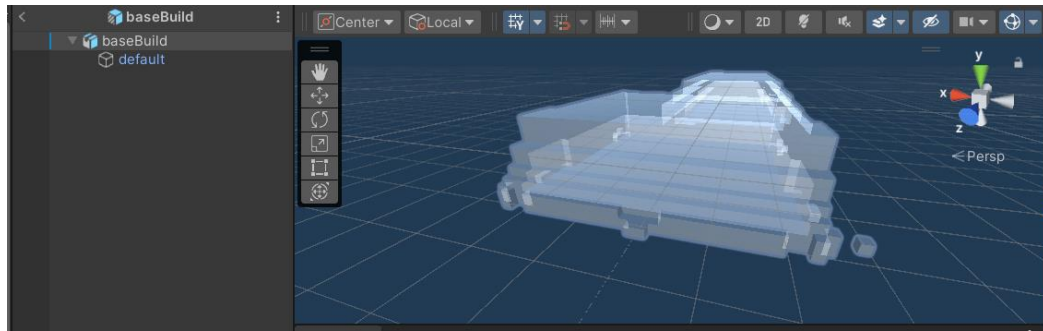


Рисунок 3.22 – Об'єкт baseBuild

Даний об'єкт має такий вигляд завдяки наступному матеріалу:

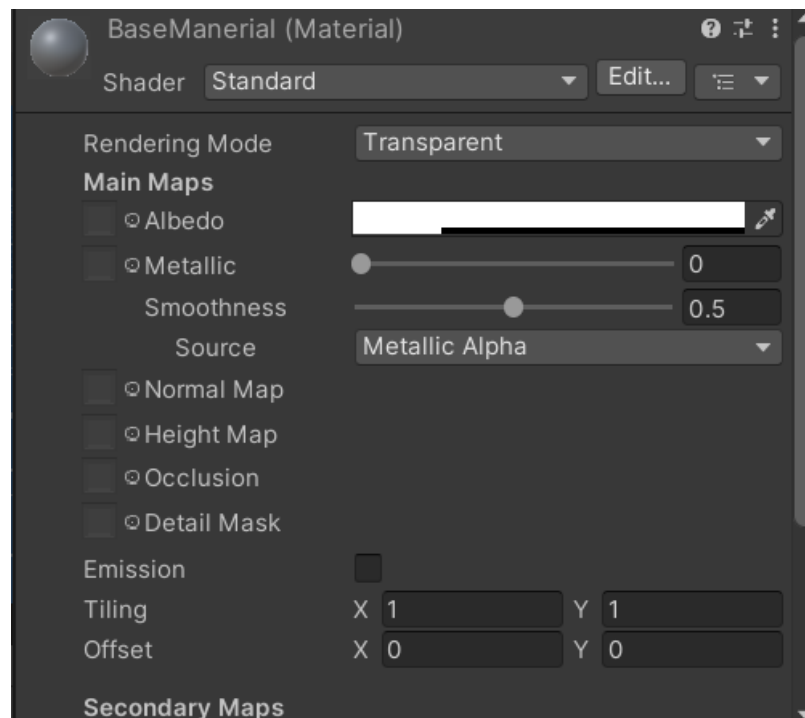


Рисунок 3.23 – Матеріал BaseMaterial

Компонент Material у Unity є ключовим елементом для визначення вигляду об'єктів у грі. Він визначає зовнішній вигляд об'єкта, включаючи кольори, текстури, блиск і інші властивості, які впливають на те, як об'єкт відображається на екрані. Основна функція матеріалу - визначення кольорів і взаємодія з освітленням. Можливість задавати освітлення, тіні та інші параметри дозволяє створювати різноманітні ефекти відображення на поверхні об'єктів.

Матеріал може містити текстури, які накладаються на поверхню об'єкта. Це може бути текстура кольору, текстура нормалей для деталізації, текстура відбитків світла, і т.д. Текстури надають додаткову реалістичність та деталізацію об'єктам.

Матеріали можуть використовувати шейдери. Вони дозволяють реалізовувати різноманітні ефекти, такі як водяний вигляд, рельєфність, мерехтіння та інші.

Також більшості випадків матеріали взаємодіють з фізичним двигуном гри, враховуючи фактори, такі як тривалість відскоку, тертя і т.д.

Використання компонента Material в Unity дозволяє розробникам ефективно контролювати візуальний аспект об'єктів у грі, створюючи різноманітні та реалістичні сцени.

Аналогічно створений об'єкт dubsBuild (рис. 3.24).

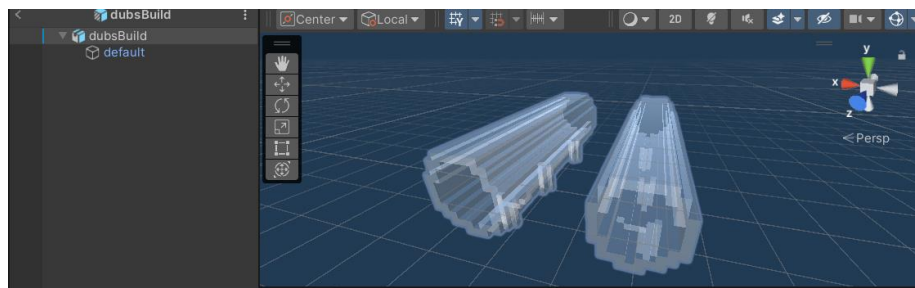


Рисунок 3.24 – Об'єкт dubsBuild

Кожен з цих силуетів змінюється вже на повністю відрендерені об'єкти з яким можна взаємодіяти.

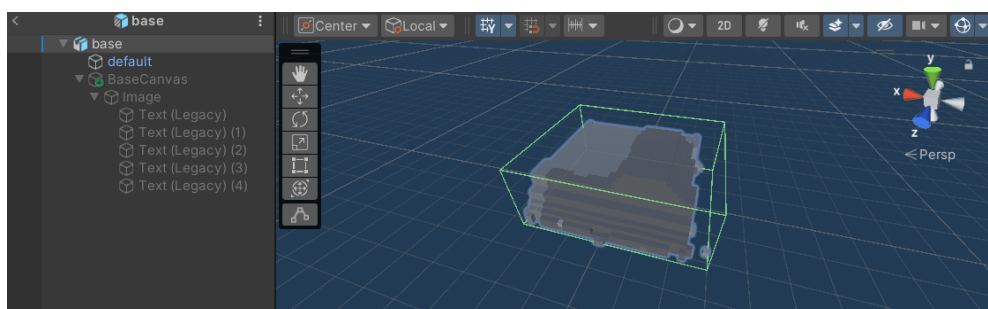


Рисунок 3.25 – Об'єкт base.

Для чіткого розуміння яким чином створюються дані об'єкти на місцевості потрібно розробити інтерфейс, який надавав коротку інформацію про об'єкт та етапи його побудови.

Для цього потрібно створити об'єкт Canvas з налаштуванням Render Mode: World Space.

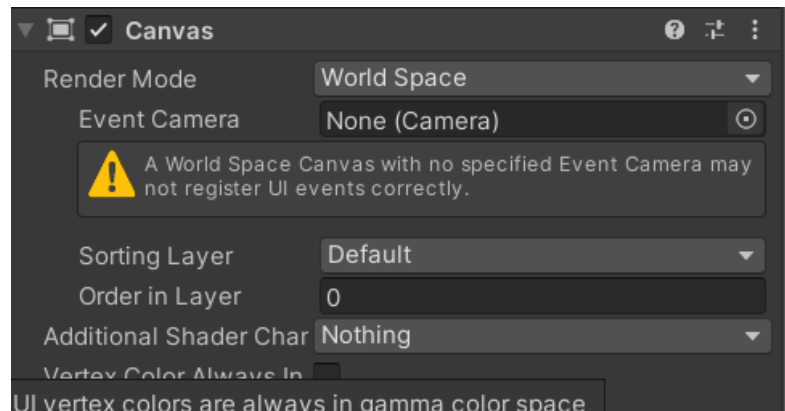


Рисунок 3.26 – Налаштування об'єкту Canvas

Завдяки даним налаштуванням, є можливість створювати елементи користувацького інтерфейсу безпосередньо в самому об'єкті, а не тільки як окремі елементи (рис. 3.27).

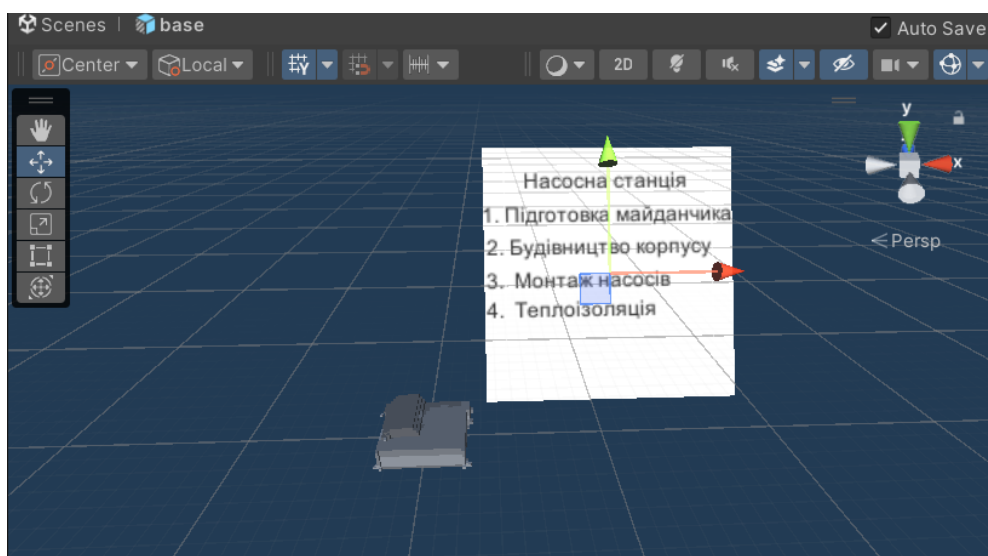


Рисунок 3.27 – UI елементи в ігровому об'єкті

Розробка Користувацького інтерфейсу передбачає створення декількох кнопок для встановлення моделей об'єктів на ландшафт.

Для роботи створено 3 кнопки. Першою кнопкою є «Побудувати трубопровід», який дає можливість створювати ігрові об'єкти dubs.

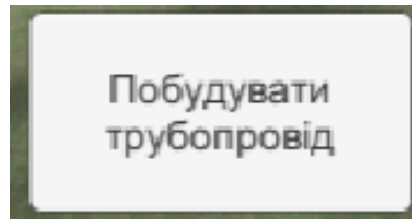


Рисунок 3.28 – Кнопка для побудови трубопроводу

Наступна кнопка «Побудувати насосну станцію», використовується для побудови моделі насосної станції.

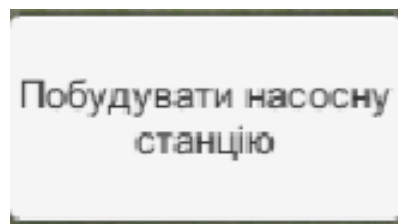


Рисунок 3.29 – Кнопка для побудови насосної станції

Завдяки даним елементам інтерфейсам користувач може обирати категорію будівель яку він хоче побудувати на ландшафті.

Остання кнопка це відміна будівництва:

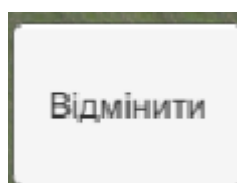


Рисунок 3.30 – Кнопка для відміни будівництва



Використовується для відміни будь-яких дій пов'язаних з відображенням силуетів об'єктів, щоб припинити будівництво.

### 3.4 Тестування створеного продукту

Створення моделі, що може бути орієнтиром в подальшому для будівництва реальної водоносної системи.

Для початку користувачу потрібно завантажити та запустити програму. Користувач спостерігатиме наступний інтерфейс (рис. 3.31).

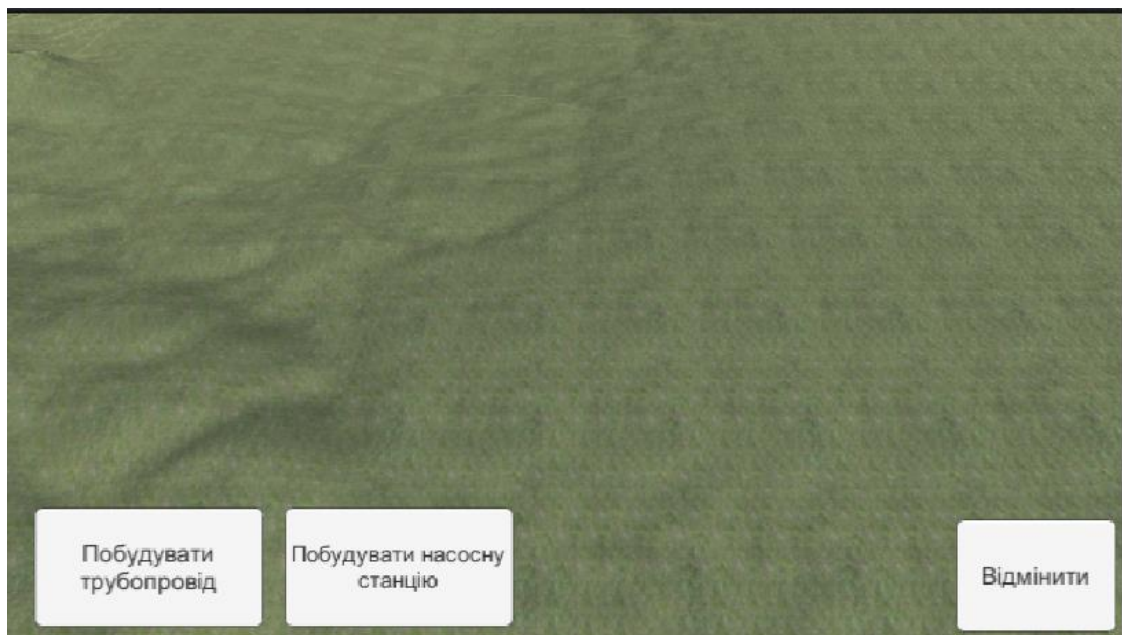


Рисунок 3.31 – Завантаження програми

Перш ніж перейти до побудови моделі системи водопостачання для початку користувач має перевірити роботу переміщення камери. Для цього користувачеві потрібно натискати на кнопки «W», «A», «D», «S».

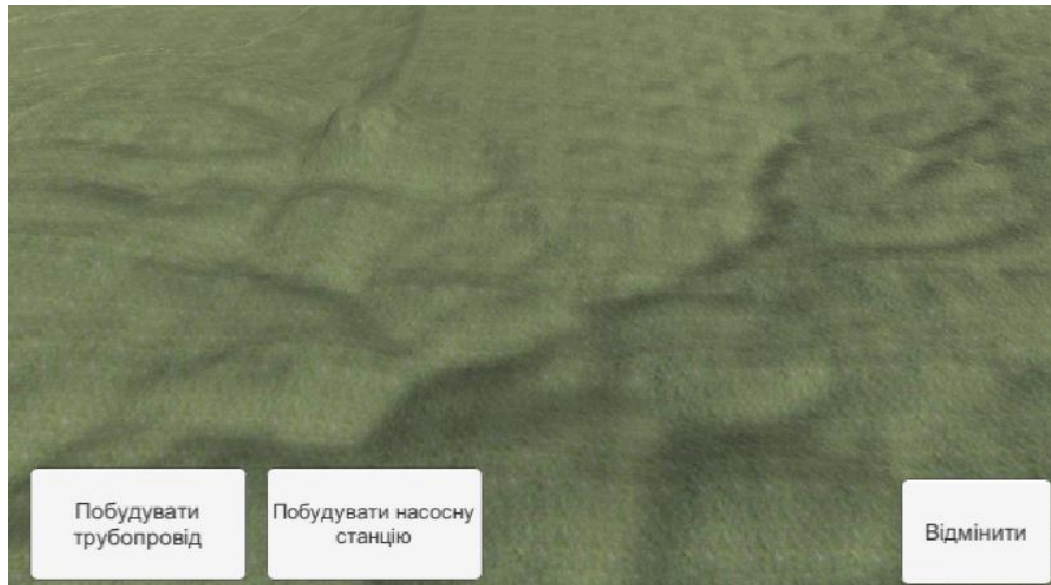


Рисунок 3.32 – Переміщення камери за допомогою клавіш

Якщо переміщення екрану відбулося без помилок, то також є можливість підняти камеру, щоб збільшити розглянути більший ландшафт та об'єкти (рис. 3.33):

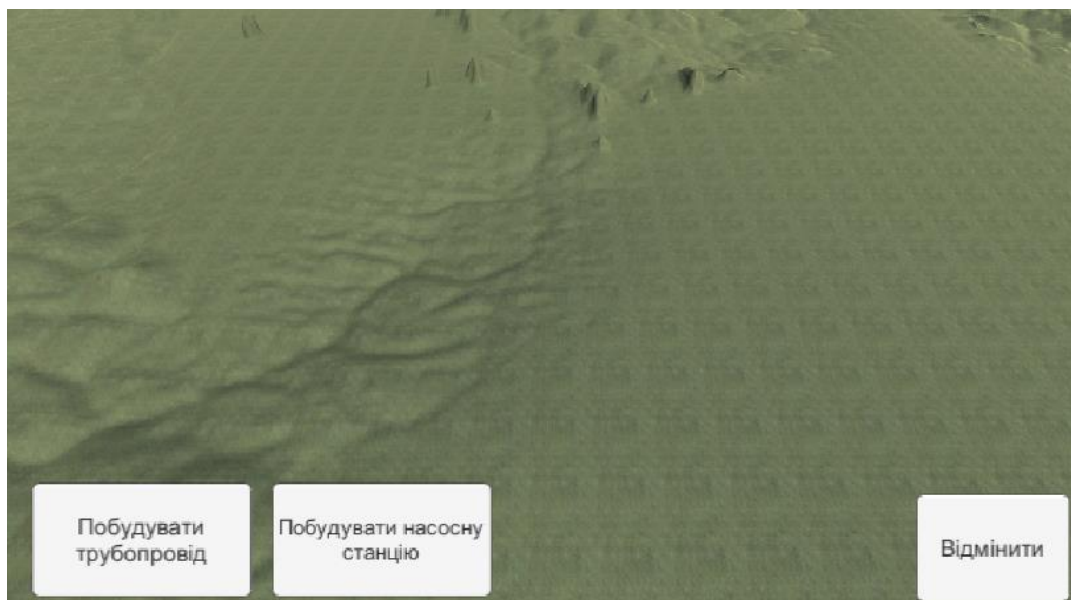


Рисунок 3.33 – Переміщення камери в верх

Переміщення камери в гору відбулося успішно, це можна спостерігати, порівняно з попереднім рисунком на 3.33 збільшилося область при якій можна розглянути ландшафт.

Додаток акож дозволяє прискорити переміщення камери за допомогою клавіши «Shift».

Далі користувачеві надається можливість встановити об'єкти на поверхню ландшафту. Для цього потрібно обрати одну з категорій будівель, наприклад «Побудувати трубопровід», та натиснути на відповідну кнопку (рис. 3.34).

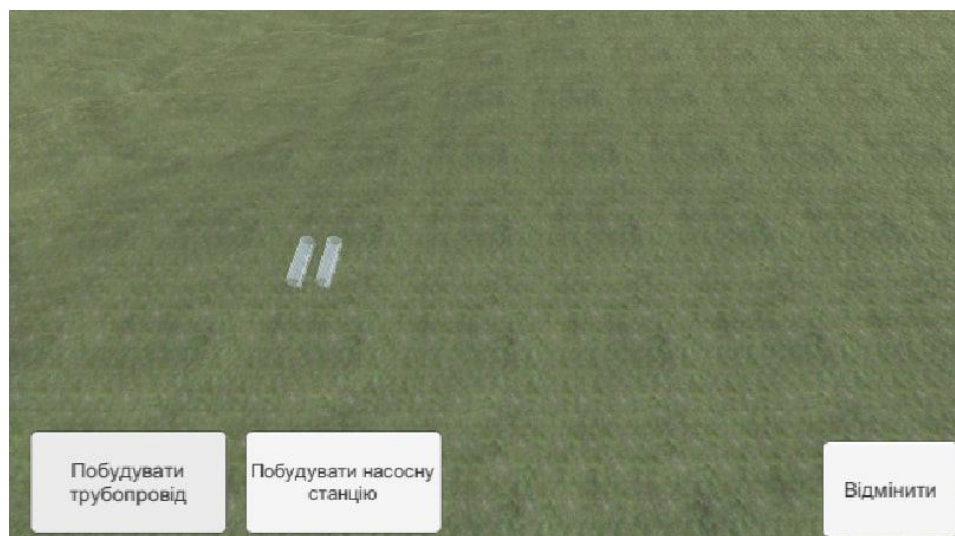


Рисунок 3.34 – Будівництво трубопроводу

Користувач спостерігає, що з'явився певний силует об'єкту, який він бажає побудувати. Далі якщо переміщати курсор миші або камеру то відповідно і об'єкт також рухатиметься за ним на ландшафті (рис. 3.35).



Рисунок 3.35 – Переміщення об'єкту для будівництва

Щоб встановити об'єкт потрібно натиснути на ліву кнопку миші на те місце де користувач бажає розмістити об'єкт (рис. 3.36).

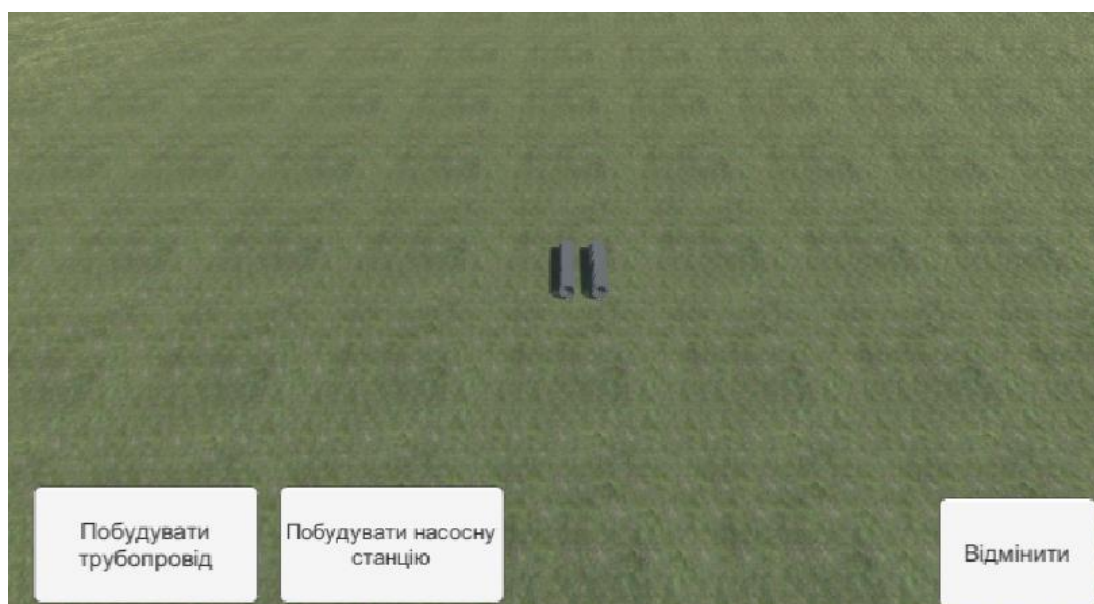


Рисунок 3.36 – Розміщення об'єкту на рельєфі

Звісно, що користувач може побудувати декілька таких об'єктів (рис. 3.37).

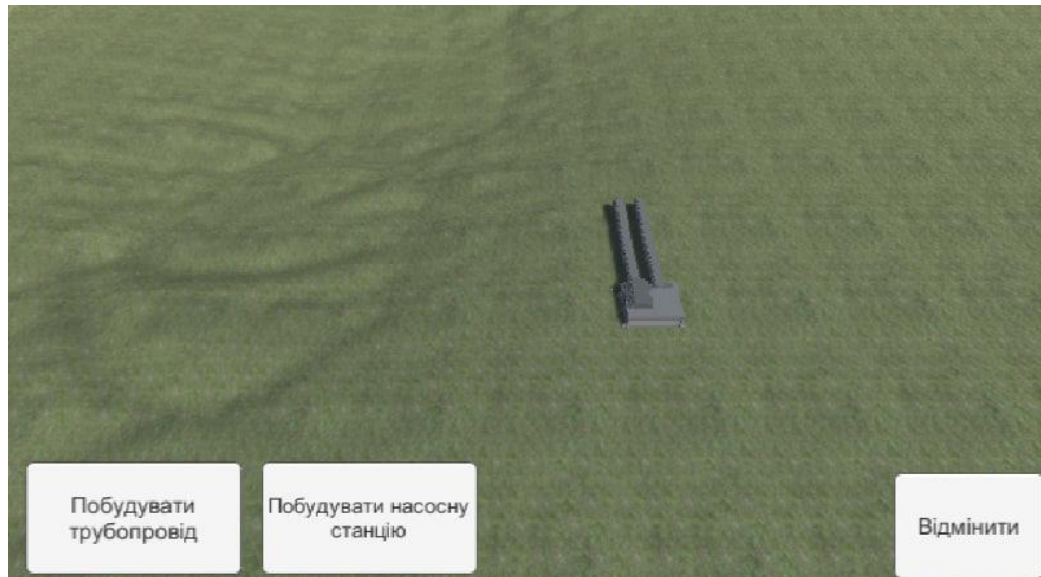


Рисунок 3.37 – Створення декількох об'єктів

Також користувач може ознайомитися на план робіт кожної зі споруд яку потрібно побудувати. Для цього потрібно навести курсор миші на об'єкт, і після цього виведеться вікно інтерфейсу із зазначеним планом робіт які потрібні для побудови даного об'єкту.

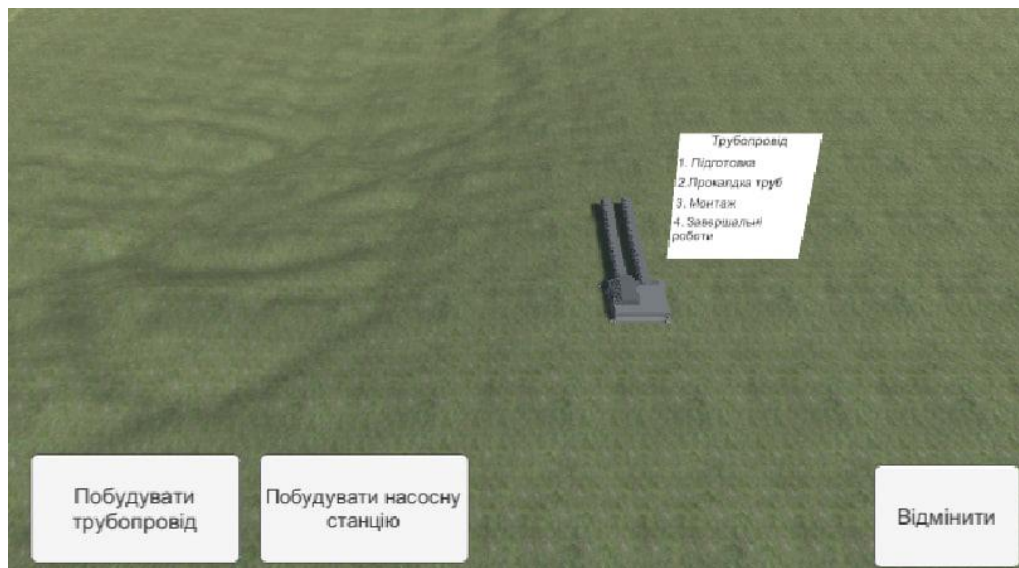


Рисунок 3.38 – Відображення плану робіт об'єкту

### 3.5 Висновок розділу

Отже, результатом розробки є додаток, який відкриває широкі можливості для моделювання завдань побудови водопостачальної системи в місті. Шляхом створення реалістичної 3D-моделі поверхні та ефективного розміщення моделей об'єктів, програма надає зручний та ілюстративний інструмент для візуалізації вигляду системи постачання води. Цей додаток становить значний крок у напрямку вдосконалення процесів планування та дизайну водопостачання, сприяючи оптимізації рішень та підвищенню ефективності водних інфраструктурних проєктів. Результати роботи вказують на високий потенціал використання цього додатку як інструменту для проєктування, що може значно полегшити завдання фахівців у галузі інженерії та водних ресурсів.

## ВИСНОВОК

Unity являє собою важливий інструмент для архітекторів, дизайнерів, інженерів та фахівців у галузі тривимірного моделювання, відкриваючи широкі можливості для створення реалістичних і детальних візуалізацій архітектурних об'єктів, ландшафтного дизайну, картографічних матеріалів та інженерних систем, таких як водопостачальні системи. Застосування Unity дозволяє не лише виконувати базові маніпуляції з тривимірними моделями, але й інтегрувати додаткові інструменти та розроблені моделі, оптимізуючи процес розробки та дозволяючи зосередитися на ключових аспектах дизайну. Це сприяє економії часу та засобів, дозволяючи одночасно підвищити деталізацію та реалізм візуалізацій.

Використання Unity в галузі обробки 3D моделей карт та їх моделювання відкриває нові можливості для створення деталізованих та реалістичних візуальних представлень місцевості, що має застосування у багатьох сферах від геодезії до ігрової індустрії. Вбудована система ландшафту та можливості програмування шейдерів у Unity дозволяють реалізувати складні проекти, враховуючи рельєф, текстури та ефекти світла.

Також Unity відіграє ключову роль у розробці спеціалізованих інженерних програм, таких як системи водопостачання. Створення реалістичних 3D-моделей поверхні з ефективним розміщенням об'єктів надає потужний інструмент для візуалізації та планування, сприяючи оптимізації рішень і підвищенню ефективності проектів.

Використання Unity, таким чином, сприяє ефективному втіленню творчих ідей та концепцій у реальність, відкриваючи нові перспективи для детального обговорення, корекції та удосконалення проектів на ранніх етапах розробки. Це робить Unity незамінним інструментом у широкому спектрі галузей, від архітектури до інженерії, дозволяючи фахівцям ефективно реалізовувати комплексні та інноваційні проекти. Платформа не тільки

забезпечує потужний візуальний інструментарій для деталізації та реалізму візуалізацій, але й надає можливість для глибшого аналізу та оцінки проектних рішень через інтеграцію різноманітних даних і моделювання сценаріїв в реальному часі.

У підсумку, Unity стає важливим інструментом у руках професіоналів, який надає величезні можливості для інновацій, підвищення ефективності та досягнення високої якості проектів у різноманітних галузях. Його застосування сприяє не лише технічному прогресу в архітектурі, інженерії та дизайні, але й відкриває нові горизонти для креативності та творчого вираження, роблячи процес проектування більш інтерактивним, доступним та вражаючим.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Найкраще програмне забезпечення для 3D моделювання в 2022 році [Електронний ресурс] – Режим доступу: <https://zenfolio.com/blog/3dmodeling-software/>.
2. Головна сторінка ZBrush [Електронний ресурс] – Режим доступу: <https://pixologic.com/>.
3. Головна сторінка Autodesk [Електронний ресурс] – Режим доступу: <https://www.autodesk.eu/>.
4. Головна сторінка Blender [Електронний ресурс] – Режим доступу: <https://www.blender.org/>.
5. Кроністер Д. Blender Basics 3rd edition / Джеймс Кроністер., 2010. – 153с.
6. Використання 3D графіки у створенні концептів [Електронний ресурс] – Режим доступу до ресурсу: <https://evenant.com/using-3d-in-concept-art/>.
7. Документація Unity3D [Електронний ресурс]. – Режим доступу: <https://docs.unity.com>
8. Документація C# [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/ru-ru/dotnet/csharp/>
9. Training content comparison of future architects in economic conditions in Poland and Ukraine // National Economic Reform: experience of Poland and prospects for Ukraine – Collective monograph. / Inna Birillo et al.– Vol. 1. “Izdevnieciba “Baltija Publishing”, 2016. – p. 61-77.
10. Фірсов О. Д., Ульяновська, Ю. В., Мормуль, М. Ф., Пікулін, Д. О.ПРОЕКТУВАННЯ АРХІТЕКТУРИ ВЕБ-ЗАСТОСУНКУ ДЛЯ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ УПРАВЛІННЯ ТРАНСПОРТНИМИ ПОТОКАМИ.// Системи та технології, Том 63 № 1 (2022), с 70-87. <https://doi.org/10.32782/2521-6643-2022.1-63.6>

11. Review of Micro-Simulation Models. [Электронный ресурс] – Режим доступа: <http://www.its.leeds.ac.uk/projects/smartertest/append3d.html#a3>.

12. Mapbox Maps SDK for Unity [Электронный ресурс] – Режим доступа: <https://docs.mapbox.com/mapbox-unity-sdk/>

13. Maps SDK for Unity [Электронный ресурс] – Режим доступа: <https://docs.mapbox.com/unity/guides/>

14. MapBox Documentation [Электронный ресурс] – Режим доступа: <https://docs.mapbox.com/help/getting-started/unity/>

15. Class AbstractMap MapBox [Электронный ресурс] – Режим доступа: <https://docs.mapbox.com/mapbox-unity-sdk/api/unity/Mapbox.Unity.Map.AbstractMap.html>

16. OpenStreetMap, [Электронный ресурс] – Режим доступа: <https://www.openstreetmap.org/#map=6/48.578/31.113>

17. Overpass API [Электронный ресурс] – Режим доступа: [https://wiki.openstreetmap.org/wiki/Overpass\\_API](https://wiki.openstreetmap.org/wiki/Overpass_API)

Overpass API User's Manual [Электронный ресурс] – Режим доступа: <https://dev.overpass-api.de/overpass-doc/en/>

18. Open Street Map Client [Электронный ресурс] – Режим доступа: <https://assetstore.unity.com/packages/tools/integration/open-street-map-client-143711>

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class EnableCanvas : MonoBehaviour
{
    public GameObject canvas;
    private void OnMouseEnter()
    {
        canvas.SetActive(true);
    }
    private void OnMouseExit()
    {
        canvas.SetActive(false);
    }
}

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ButtonBuildBase : MonoBehaviour
{
    public GameObject baseDum;
    public GameObject Dumbs;

    public void BuildBase()
```

```

    {
        Instantiate(baseDum);
    }
public void BuildDumb()
{
    Instantiate(Dumbs);
}
public void DeleteAllBuilds()
{
    GameObject[] list = GameObject.FindGameObjectsWithTag("BuildPrint");
    foreach (GameObject go in list)
    {
        Destroy(go);
    }
}
}
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class BuildingScript : MonoBehaviour
{
    RaycastHit hit;
    Vector3 movePoint;
    public GameObject prefab;
    // Start is called before the first frame update
    void Start()
    {

```

```
Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
Debug.Log(Physics.Raycast(ray, out hit, 50000.0f, (1 << 8)));
```

```
if (Physics.Raycast(ray, out hit, 50000.0f, (1<<8)))
{
    transform.position = hit.point;
}
}
```

// Update is called once per frame

```
void Update()
{
    Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
    if(Physics.Raycast(ray,out hit, 50000.0f, (1 << 8)))
    {
        transform.position = hit.point;
    }
    if(Input.GetMouseButtonDown(0))
    {
        Instantiate(prefab, transform.position, transform.rotation);
        Destroy(gameObject);
    }
}

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
```

```
public class CameraControl : MonoBehaviour
{
    // Start is called before the first frame update
    float speed = 0.06f;
    float zoomspeed = 10.0f;
    float rotateSppeed;

    float maxHeight = 120f;
    float minHeight = 1f;

    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        if(Input.GetKey(KeyCode.LeftShift))
        {
            speed = 0.06f;
            zoomspeed = 20.0f;
        }
        else
        {
            speed = 0.035f;
            zoomspeed = 10.0f;
        }
    }
}
```

```

float hsp =transform.position.y * speed * Input.GetAxis("Horizontal");
float vsp =transform.position.y * speed * Input.GetAxis("Vertical");
float scrolSp =Mathf.Log(transform.position.y) * -zoomspeed *
Input.GetAxis("Mouse ScrollWheel");

if((transform.position.y >= maxHeight) && (scrolSp >0))
{
    scrolSp = 0;
}
else if((transform.position.y <= minHeight) && (scrolSp<0))
{
    scrolSp = 0;
}
Vector3 verticalMove = new Vector3(0, scrolSp, 0);
Vector3 lateralMove = hsp * transform.right;
Vector3 forwardMove = transform.forward;
forwardMove.y = 0;
forwardMove.Normalize();
forwardMove *= vsp;

Vector3 move = verticalMove + lateralMove + forwardMove;

transform.position += move;
}
}

```