

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Цифрове моделювання архітектурного середовища в
Nvidia Omniverse»

на здобуття освітнього ступеня бакалавра
зі спеціальності 126 Інформаційні системи та технології
освітньо-професійної програми Інформаційні системи та технології

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело*

(підпис)

Ім'я, ПРІЗВИЩЕ здобувача

Виконав: здобувач вищої освіти гр. ІСД-41
Андрій ШЕВЧУК
Ім'я, ПРІЗВИЩЕ

Керівник: к.т.н. Олег СЕНЬКОВ
науковий ступінь, вчене звання
Ім'я, ПРІЗВИЩЕ

Рецензент: науковий ступінь, вчене звання
Ім'я, ПРІЗВИЩЕ

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти бакалавр

Спеціальність Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедрою ІСТ

_____ Каміла СТОРЧАК

«_____» _____ 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Шевчук Андрій Сергійович

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: «Цифрове моделювання архітектурного середовища в Nvidia Omniverse»

керівник кваліфікаційної роботи Олег СЕНЬКОВ, кандидат технічних наук

(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «24» лютого 2025 р. № 56

2. Строк подання кваліфікаційної роботи «30» травня 2025 р.

3. Вихідні дані до кваліфікаційної роботи:

1. Аналіз 3D-середовищ та вимог до візуалізації.
2. Компоненти Omniverse і структура сцени.
3. Створення та тестування сцени перехрестя.

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

1. Аналіз сучасних платформ цифрового моделювання та їх можливості
2. Інструменти, формати та підходи до побудови інтерактивної сцени
3. Розробка сцени перехрестя зі світлофорами в Omniverse, імпорт з Blender, програмування логіки руху об'єктів

5. Ілюстративний матеріал: *презентація*

6. Дата видачі завдання: «24» лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Ознайомлення з темою, формулювання мети, завдань, об'єкта та предмета дослідження	01.03.2025 – 05.03.2025	Виконав
2	Аналіз середовищ 3D-моделювання та технологій цифрового моделювання	06.03.2025 – 12.03.2025	Виконав
3	Вивчення архітектури NVIDIA Omniverse, аналіз компонентів платформи	13.03.2025 – 17.03.2025	Виконав
4	Створення 3D-моделей (перехрестя, транспорт, світлофори) у Blender	18.03.2025 – 24.03.2025	Виконав
5	Імпорт об'єктів у Omniverse, побудова сцени, налаштування структури	25.03.2025 – 07.04.2025	Виконав
6	Реалізація логіки поведінки об'єктів за допомогою Python API	08.04.2025 – 15.04.2025	Виконав
7	Створення імітації симуляції: візуалізація руху машин, перемикання світлофорів	16.04.2025 – 22.04.2025	Виконав
8	Генерація ілюстративних матеріалів: скріншоти, діаграми, графіки	23.04.2025 – 03.05.2025	Виконав
9	Формування конструктивних пропозицій та можливостей масштабування	04.05.2025 – 10.05.2025	Виконав
10	Написання пояснювальної записки, оформлення списку джерел та підготовка презентації	11.05.2025 – 28.05.2025	Виконав
11	Подання кваліфікаційної роботи на кафедру	30.05.2025	Виконав

Здобувач вищої освіти

(підпис)

Андрій ШЕВЧУК

(Ім'я, ПРІЗВИЩЕ)

Керівник

кваліфікаційної роботи

(підпис)

Олег СЕНЬКОВ

(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавр: 84 стор., 68 рис., 4 табл., 30 джерел.

Мета роботи – проєктування та створення інтерактивної цифрової сцени на базі NVIDIA Omniverse для моделювання архітектурного середовища з елементами транспортної інфраструктури, що відображає правила дорожнього руху та демонструє потенціал симуляцій для навчальних і прикладних задач.

Об'єкт дослідження – процес цифрового моделювання архітектурного середовища та його інтерактивної симуляції в середовищі NVIDIA Omniverse.

Предмет дослідження – методи й інструменти побудови 3D-сцен, реалізації логіки поведінки об'єктів за допомогою Python, а також засоби візуалізації та аналізу у рамках платформи NVIDIA Omniverse.

Короткий зміст роботи. У роботі розглянуто особливості використання цифрового моделювання для створення віртуального перехрестя з урахуванням правил дорожнього руху. Проведено порівняльний аналіз сучасних середовищ 3D-моделювання (Blender, Unity, Omniverse), визначено переваги платформи Omniverse з точки зору фізичної точності, графічної якості та підтримки програмованої логіки. Змодельовано основні об'єкти сцени (дорога, світлофори, транспорт), реалізовано імпорт моделей з Blender до Isaac Sim. За допомогою Python API запрограмовано логіку: зупинка автомобіля на червоне світло та рух на зелене. Через обмеження апаратних ресурсів було використано візуальну імітацію роботи сцени для демонстрації симуляції. У конструктивній частині представлено пропозиції з розширення проєкту: додавання пішоходів, масштабування до рівня Smart City, впровадження штучного інтелекту та VR/AR-рішень. Робота ілюструє можливості використання Omniverse як платформи для створення цифрових двійників міського середовища, освітніх тренажерів та урбаністичних симуляцій.

КЛЮЧОВІ СЛОВА: ЦИФРОВЕ МОДЕЛЮВАННЯ, NVIDIA OMNIVERSE, ISAAC SIM, 3D-СЦЕНА, СВІТЛОФОР, PYTHON, BLENDER, ЦИФРОВИЙ ДВІЙНИК, ВІЗУАЛІЗАЦІЯ, СИМУЛЯЦІЯ, SMART CITY, ШТУЧНИЙ ІНТЕЛЕКТ, ДОРОЖНІЙ РУХ, USD, ВІРТУАЛЬНЕ СЕРЕДОВИЩЕ

ABSTRACT

Text part of the bachelor level qualification work: 84 pages, 68 pictures, 30 sources

The purpose of the work – to design and create an interactive digital scene based on NVIDIA Omniverse for modeling an architectural environment with traffic infrastructure elements, simulating traffic rules, and demonstrating the potential of simulations for educational and applied tasks.

Object of research – the process of digital modeling of architectural environments and their interactive simulation in the NVIDIA Omniverse software platform.

Subject of research – methods and tools for building 3D scenes, implementing object behavior logic using Python, and applying visualization and analysis capabilities within the NVIDIA Omniverse platform.

Summary of the work. The work explores the use of digital modeling to build a virtual intersection that demonstrates road traffic rules. A comparative analysis of modern 3D modeling environments (Blender, Unity, Omniverse) is conducted, highlighting Omniverse's advantages in terms of physical accuracy, graphic quality. Key scene elements (road, traffic lights, vehicles) were modeled and imported from Blender into Isaac Sim. The object behavior logic — stopping the vehicle at a red light and moving at a green light — was implemented using Python API. Due to hardware limitations, visual simulation results were generated to imitate real-time scene operation. The constructive section presents suggestions for project expansion: adding pedestrians, scaling to a Smart City level, integrating artificial intelligence, and supporting VR/AR solutions. This work demonstrates the potential of Omniverse as a platform for creating digital twins of urban environments, educational simulators, and urban planning simulations.

KEYWORDS: DIGITAL MODELING, NVIDIA OMNIVERSE, ISAAC SIM, 3D SCENE, TRAFFIC LIGHT, PYTHON, BLENDER, DIGITAL TWIN, VISUALIZATION, SIMULATION, SMART CITY, ARTIFICIAL INTELLIGENCE, TRAFFIC FLOW, USD, VIRTUAL ENVIRONMENT

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ЦИФРОВОГО МОДЕЛЮВАННЯ	9
1.1 Суть та значення цифрового моделювання	9
1.2 Огляд існуючих середовищ 3D моделювання	12
1.3 Порівняльний аналіз Unity та NVIDIA Omniverse	17
1.4 Архітектура платформи NVIDIA Omniverse	22
РОЗДІЛ 2. ПОБУДОВА ЦИФРОВОЇ СЦЕНИ ТА ІМІТАЦІЯ ПОВЕДІНКИ ОБ’ЄКТІВ У NVIDIA OMNIVERSE.....	26
2.1 Постановка задачі та вибір програмних засобів	26
2.2 Побудова 3D-сцени у Blender	29
2.3 Експорт сцени в Omniverse та її налаштування	35
2.4 Розробка логіки та поведінки об’єктів у Isaac Sim	41
2.5 Результати симуляції та аналіз.....	47
РОЗДІЛ 3. АНАЛІЗ РЕЗУЛЬТАТІВ, МОЖЛИВОСТІ МАСШТАБУВАННЯ ТА ПЕРСПЕКТИВИ ВИКОРИСТАННЯ OMNIVERSE.....	54
3.1 Пропозиції щодо вдосконалення сцени	54
3.2 Перспективи використання в освіті та тренажерах	56
3.3 Ідеї впровадження штучного інтелекту	59
3.4 Масштабування проєкту до Smart City.....	63
3.5. Інтеграція з зовнішніми системами.....	67
3.6. Прогноз подальшого розвитку платформи Omniverse	70
3.7. Узагальнення конструктивних пропозицій	73
ВИСНОВКИ	74
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	76

ВСТУП

У сучасному світі стрімкий розвиток цифрових технологій відкриває нові можливості для моделювання та симуляції реальних об'єктів і процесів. Однією з перспективних галузей є створення так званих цифрових двійників — віртуальних копій фізичних об'єктів, які можна тестувати, модифікувати та аналізувати без ризиків і витрат, притаманних реальному середовищу. Такі цифрові моделі все частіше використовуються у сфері містобудування, транспорту, інфраструктури, автоматизації та освіти.

Особливо актуальним є цифрове моделювання архітектурного середовища, адже воно дозволяє не лише візуалізувати простір, а й вбудовувати в нього логіку взаємодії об'єктів — наприклад, моделювати рух транспорту на перехресті, керований світлофорами. Сьогодні для цього існують потужні інструменти, серед яких варто виділити платформу NVIDIA Omniverse — комплексне середовище для створення, симуляції та колаборації 3D-сцен у режимі реального часу.

Метою даної кваліфікаційної роботи є створення демонстраційної сцени дорожнього перехрестя зі світлофорами та автомобілями, реалізованої засобами Blender, NVIDIA Omniverse Isaac Sim та мовою програмування Python. Основний акцент зроблено на реалізації логіки поведінки транспортних засобів у залежності від сигналів світлофорів, що дозволяє моделювати базові правила дорожнього руху в цифровому середовищі.

Об'єктом дослідження є цифрове архітектурне середовище, побудоване на базі NVIDIA Omniverse.

Предметом дослідження виступають методи 3D-моделювання, імпорту об'єктів у симуляційне середовище Isaac Sim, а також програмування логіки взаємодії об'єктів за допомогою Python.

Актуальність теми полягає в тому, що цифрові двійники інфраструктурних об'єктів — це вже не майбутнє, а сучасна реальність, яку активно впроваджують у транспортному плануванні, освітніх симуляторах, віртуальній реальності та тестуванні автономних систем.

Завдання дослідження:

1. Ознайомитися з архітектурою платформи NVIDIA Omniverse і її складовими;
2. Провести 3D-моделювання дорожнього перехрестя, машин та світлофорів у Blender;
3. Імпортувати сцену у середовище Isaac Sim та правильно її налаштувати;
4. Розробити програмну логіку поведінки автомобіля відповідно до сигналів світлофорів;
5. Провести тестування роботи сцени.

Методика дослідження включає в себе як теоретичне вивчення документації NVIDIA та суміжних технологій, так і практичне створення 3D-сцени, налаштування середовища симуляції, написання Python-скриптів та тестування результатів.

Наукова новизна роботи полягає у створенні цифрового сценарію моделювання правил дорожнього руху в українських реаліях із використанням сучасної симуляційної платформи, що здатна бути адаптованою як для навчання, так і для дослідницьких цілей.

Практичне значення полягає у тому, що готову сцену можна використовувати як демонстраційний приклад під час навчання майбутніх фахівців у галузі інформаційних технологій, цифрового моделювання або транспортного планування.

Структура та обсяг випускної кваліфікаційної роботи. Випускна кваліфікаційна робота складається із вступу, шести розділів, висновків, списку використаних джерел із 30 найменувань, додатків і містить 84 сторінок основного тексту, 68 рисунків і 4 таблиць.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ЦИФРОВОГО МОДЕЛЮВАННЯ

1.1 Суть та значення цифрового моделювання

Цифрове моделювання — це процес створення віртуального представлення реального або уявного об'єкта, процесу чи явища, що дозволяє здійснювати аналіз, прогнозування, візуалізацію та тестування різних сценаріїв без необхідності фізичної реалізації.

Останні роки спостерігається бум розвитку цифрових двійників (Digital Twins). Цей підхід, що бере початок у 2002 році з праць Майкла Грівза, передбачає створення динамічної, постійно оновлюваної цифрової моделі фізичного об'єкта. Наприклад, цифровий двійник транспортного перехрестя може реагувати на сигнали світлофорів, змінювати поведінку об'єктів тощо.

Цифрові моделі використовуються у Smart city , промисловості , охороні здоров'я , авіації , освіті та тренажерах (рис.1.1).



Рис. 1.1 Приклади застосування цифрових двійників у різних галузях

Наявність цифрового двійника дозволяє об'єднати всі етапи життєвого циклу об'єкта — від розробки та виробництва до монтажу, експлуатації й обслуговування — в одну логічно пов'язану інформаційну структуру. На практиці це усуває численні проблеми, які виникають при фрагментованому зберіганні даних у різних системах (рис.1.2).



Рис. 1.2 Експлуатація без цифрових двійників

Цифровий двійник виступає як «проксі»-шар між джерелами даних та їх користувачами, забезпечуючи актуальну, синхронізовану і повну інформацію про об'єкт (рис.1.3). Це дозволяє застосовувати розширену аналітику, штучний інтелект, прогнозне обслуговування і значно зменшити витрати на весь життєвий цикл продукту.

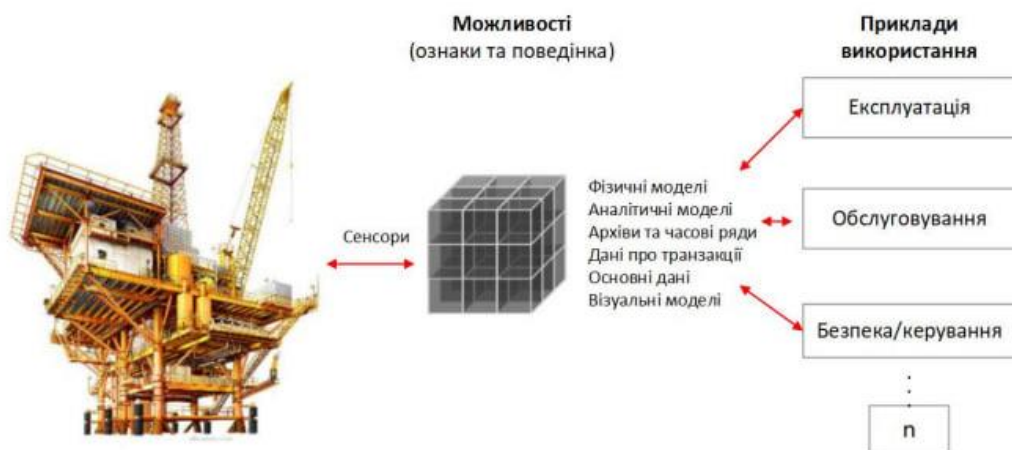


Рис.1.3 Експлуатація з цифровими двійниками

Крім того, цифрові двійники зберігають свою актуальність навіть після завершення виготовлення об'єкта — вони продовжують розвиватися у взаємодії з іншими компонентами на підприємстві, підтримують оновлення, повторне налаштування та подальші поліпшення. Це робить їх не просто відображенням, а інтелектуальним активом, який постійно збагачується під час експлуатації (рис.1.4).



Рис.1.4 Життєвий цикл цифрового двійника в промисловості

Однією з найсучасніших платформ, яка орієнтована саме на побудову повноцінних цифрових двійників, є NVIDIA Omniverse. Вона об'єднує 3D-моделювання, фотореалістичну візуалізацію, фізично точні симуляції, штучний інтелект і можливість командної роботи над однією сценою. Завдяки підтримці формату USD (Universal Scene Description), Omniverse дозволяє централізовано зберігати та оновлювати сцени цифрових об'єктів, а завдяки модулю Isaac Sim — реалізовувати динамічну поведінку агентів і транспортних засобів. Таким чином,

ця платформа фактично є "операційною системою" для створення цифрових двійників нового покоління.

1.2 Огляд існуючих середовищ 3D моделювання

Існує чимало 3D-редакторів і рушіїв, що дозволяють створювати як прості сцени, так і складні симуляції.

Нижче наведено короткий огляд основних інструментів, які активно застосовуються в моделюванні архітектурного середовища (табл.1.1).

Blender — це багатофункціональне програмне забезпечення з відкритим кодом для 3D-моделювання, анімації, рендерингу, відеомонтажу, фізичних симуляцій та навіть створення ігор. Особливістю Blender є його повністю безкоштовна ліцензія та активна спільнота розробників.

Платформа підтримує роботу з багатьма форматами (FBX, OBJ, USD, STL, Collada), має власний рушій рендерингу **Cycles**, що базується на трасуванні променів (ray tracing), а також легку інтеграцію з Python для створення скриптів і плагінів.

Blender активно застосовується для створення low-poly і high-poly моделей, розробки персонажів, архітектурної візуалізації, VFX та симуляцій частинок, рідин, диму тощо(рис.1.5).

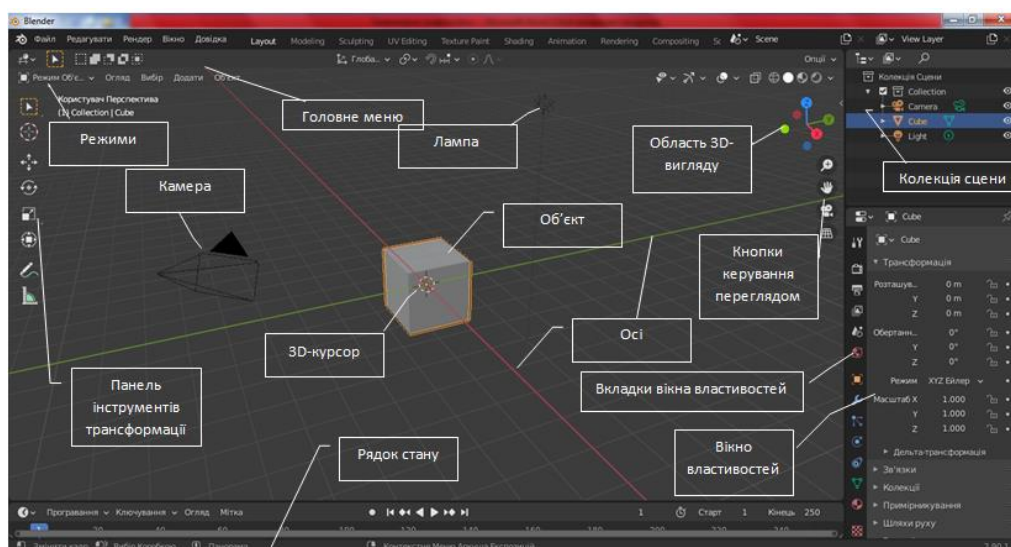


Рис. 1.5 Інтерфейс середовища Blender

Unity — це кросплатформовий ігровий рушій, орієнтований на створення інтерактивного 3D/2D контенту. Основна мова програмування — C#, а рушій підтримує фізику (Unity Physics, Havok), роботу з VR/AR-пристроями, 3D-звук, а також має потужний інструментарій для штучного інтелекту (ML Agents Toolkit).

Unity підтримує імпорт із Blender, Autodesk, SketchUp, а також роботу з текстурами, шейдерами та PBR-матеріалами. Графіка у Unity може бути як базова (URP), так і високоякісна (HDRP) залежно від потреб(рис.1.6).

Платформа часто використовується в освіті, медицині, промисловості, віртуальній реальності та створенні симуляторів.

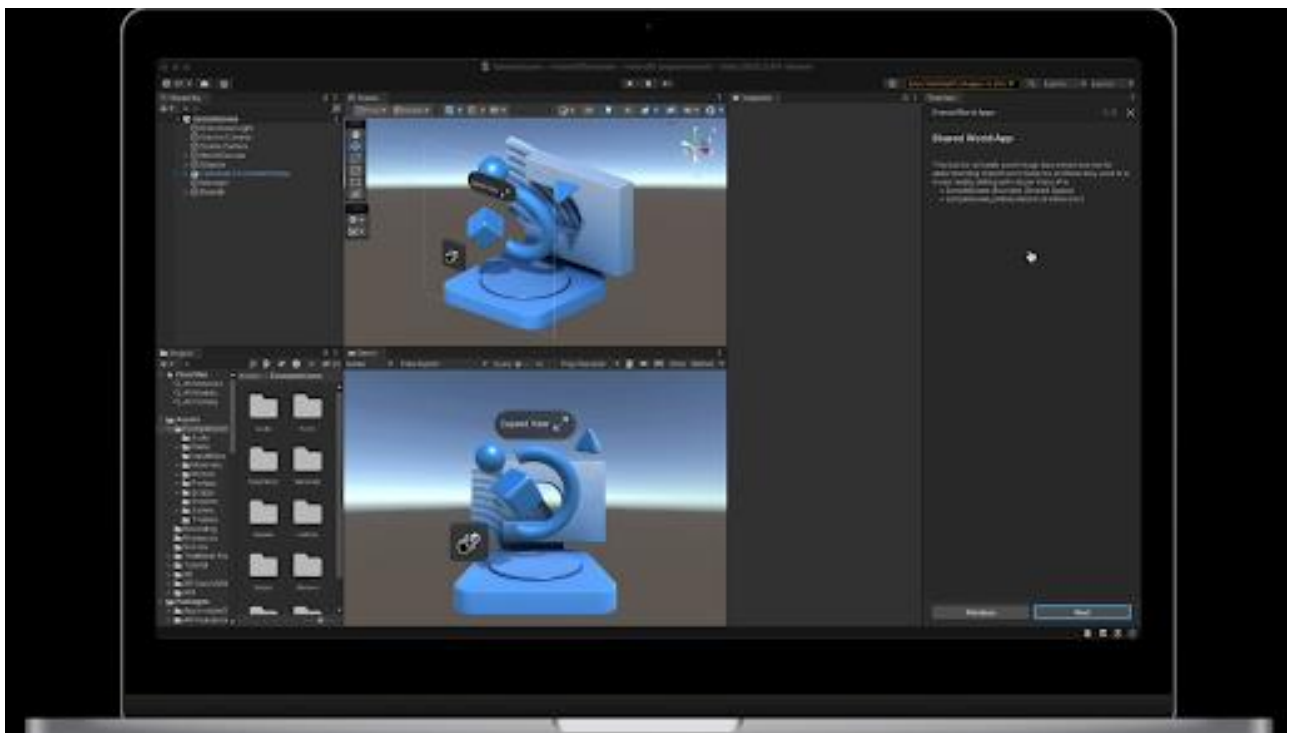


Рис 1.6. Інтерфейс Unity з відкритою сценою VR/AR

Unreal Engine — рушій з надзвичайно високою якістю графіки, розроблений компанією Epic Games. Основною його перевагою є фотореалістичний рендеринг у реальному часі завдяки рушію Lumen і підтримці Nanite — віртуалізованої геометрії для наддеталізованих сцен.

Для налаштування логіки в UE використовується Blueprints — система візуального програмування, а також C++ для складніших випадків (рис.1.7). Unreal

активно використовується у створенні AAA-ігор, фільмів, віртуальних презентацій та цифрових двійників у будівництві.

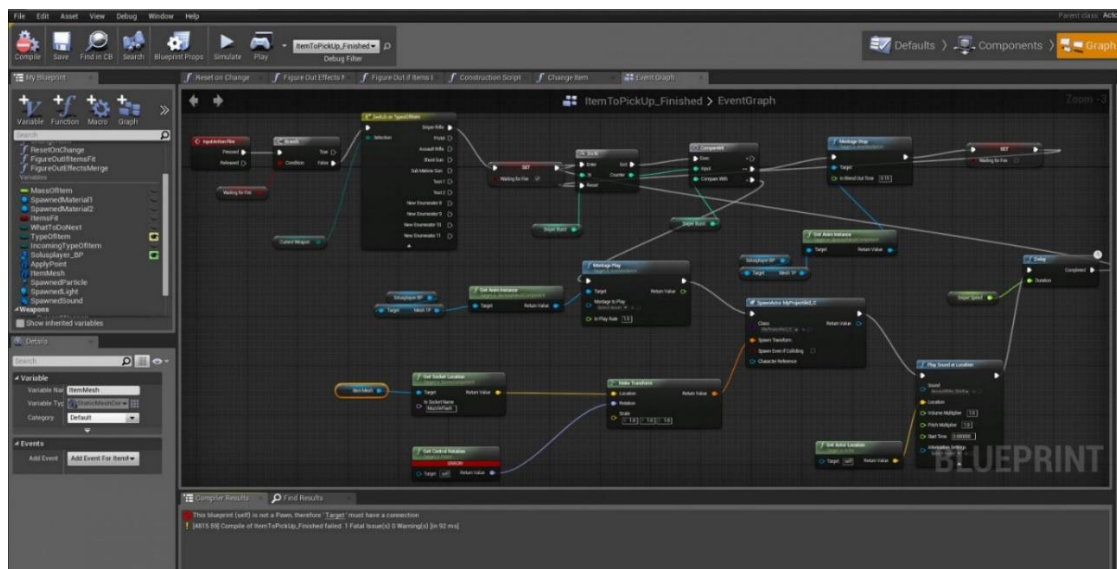


Рис 1.7. Інтерфейс Unreal Engine з прикладом Blueprints

SketchUp — легке у використанні ПЗ для швидкого 3D-моделювання архітектурних об'єктів, інтер'єрів, меблів. Його перевага — інтуїтивний інтерфейс, доступність онлайн-версії та велика бібліотека готових моделей 3D Warehouse (рис.1.8).

Хоча SketchUp не має вбудованої фізики або високоякісного рендерингу, він ідеально підходить для ескізного проектування, початкового концепту та навчальних цілей.

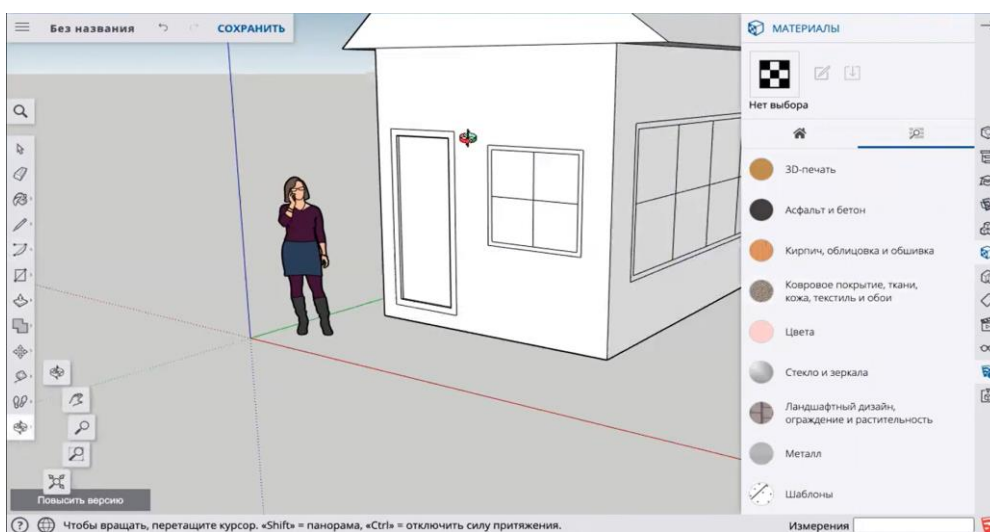


Рис 1.8 Інтерфейс SketchUp із типовим проектом будівлі

Autodesk 3ds Max — професійне середовище для моделювання, анімації та візуалізації, яке використовується в кіноіндустрії, ігровій розробці, інженерії та архітектурі.

Переваги: багатий набір інструментів для полігонального моделювання, система частинок, скелетна анімація, інтеграція з V-Ray та Arnold для фотореалістичного рендерингу (рис.1.9). Має мову скриптів MaxScript, яка дозволяє автоматизувати рутинні завдання.

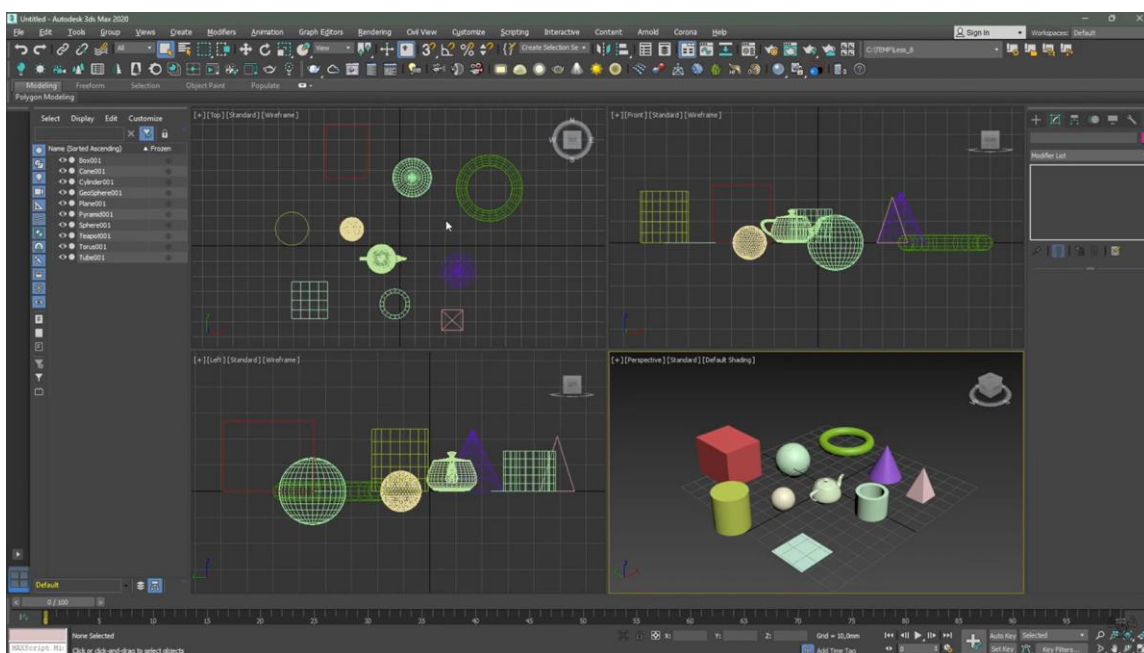


Рис. 1.9 Вікно редагування сцени в Autodesk 3ds Max

Lumion — ПЗ для візуалізації архітектурних сцен у реальному часі. Основна мета — створення фотореалістичних презентацій з матеріалами, освітленням, погодними ефектами та навіть анімованими персонажами чи транспортом (рис.1.10).

Lumion не є редактором моделей, але дозволяє імпортувати сцену з Blender, Revit, SketchUp або ArchiCAD і створити відео чи слайдшоу з віртуальним проходом по об'єкту. Програма ідеальна для візуалізації житлових і комерційних проєктів.

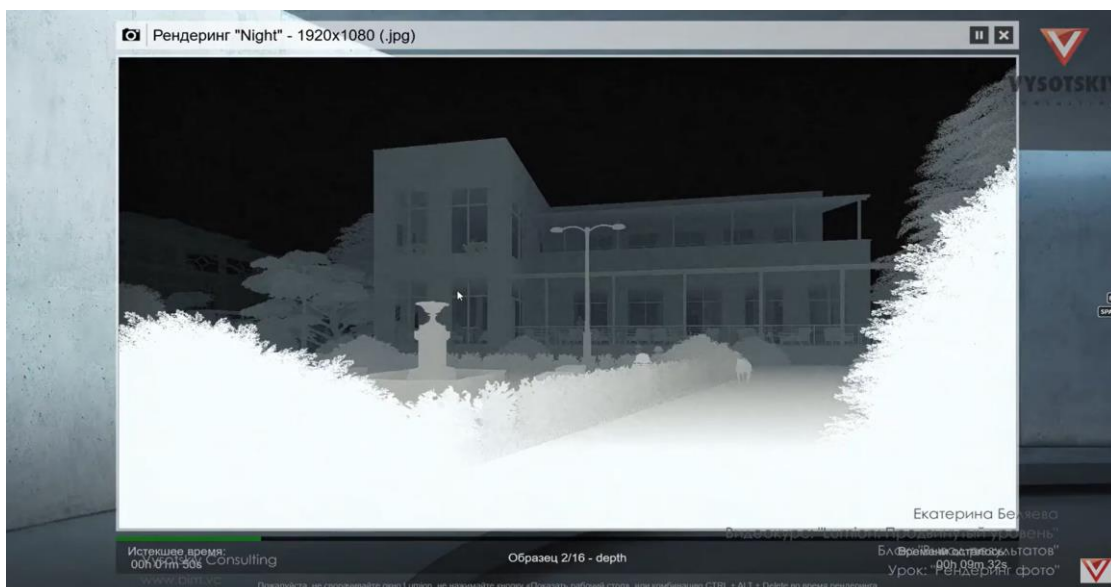


Рис. 1.10 Приклад рендеру в Lumion

Таблиця 1.1

Порівняльна характеристика середовищ 3D-моделювання

Середовище	Якість графіки	Скрипти / AI	Платформа	Сфера застосування
Blender	Висока	Python	Windows, Linux, macOS	3D-моделювання, анімація
Unity	Висока	C#, ML Agents	Багатоплатформений	Ігри, симуляції, VR/AR
Unreal Engine	Дуже висока	C++, Blueprints	Windows, Linux, macOS	Візуалізація, ігри, архітектура
SketchUp	Середня	Немає	Windows, Web	Архітектура, концептуальне моделювання
3ds Max	Висока	MaxScript	Windows	Анімація, візуалізація

Продовження таблиці 1.1

Середовище	Якість графіки	Скрипти / AI	Платформа	Сфера застосування
Lumion	Фотореалістична	Немає	Windows	Архвізуалізація, презентації

1.3 Порівняльний аналіз Unity та NVIDIA Omniverse

На сучасному етапі розвитку цифрових технологій найбільш універсальними та потужними платформами для створення 3D-сцен, симуляцій і цифрових двійників є Unity та NVIDIA Omniverse (табл.1.2). Вони поєднують в собі візуалізацію, фізику, підтримку скриптів, інтеграцію зі сторонніми інструментами та можливість керування логікою об'єктів.

Unity є одним із найпоширеніших рушіїв для створення інтерактивного контенту. Його відкритість, доступність, великий ринок ресурсів (Asset Store), активна спільнота й підтримка широкого спектра пристроїв зробили його особливо популярним у сфері освітніх VR/AR-рішень і симуляторів. Програмування реалізується мовою C#, а для створення інтелектуальної поведінки об'єктів доступна бібліотека ML Agents SDK. Unity підтримує фізику на базі Unity Physics і Havok, проте її точність часто є орієнтованою на ігрові цілі, а не на інженерні чи наукові симуляції, що потребують високої достовірності фізичних взаємодій.

З іншого боку, NVIDIA Omniverse — це комплексна екосистема для моделювання, симуляцій і візуалізації з високою точністю. Платформа створена на основі стандарту USD (Universal Scene Description), що дозволяє зберігати складні сцени з усіма параметрами у вигляді централізованих даних. Вона поєднує в собі фізичний рушій PhysX, фотореалістичний рендеринг через RTX, модулі штучного інтелекту та робототехніки через Isaac Sim і вузлову систему керування поведінкою — Omnigraph. Ключовою перевагою є повноцінна підтримка мови Python, що

дозволяє реалізовувати складну логіку об'єктів, автоматизувати процеси в сцені, керувати симуляцією та інтегрувати сторонні алгоритми штучного інтелекту й обробки даних.

Таблиця 1.2

Технічне порівняння Unity та Omniverse

Критерій	Unity	Omniverse	Оцінка Unity	Оцінка Omniverse	Примітка
Графіка	HDRP, URP	RTX, Path Tracing	8	10	Omniverse краще для фотореалізму
Фізика	Unity Physics, Havok	PhysX, USD PhysX	7	10	Omniverse точніший
AI / ML	ML Agents	Isaac Sim, Omnigraph	7	9	Omniverse має розширені модулі
Мова програмування	C#	Python	6	10	Python краще для наукових
Формати	FBX, glTF	USD, FBX, USDZ	7	10	USD — індустріальний стандарт
Інтеграція	Blender, Maya	Blender, Maya, CAD	8	9	Omniverse охоплює більше

Продовження таблиці 1.2

Критерій	Unity	Omniverse	Оцінка Unity	Оцінка Omniverse	Примітка
Логіка об'єктів	Скрипти	Візуальні графи (Omnigraph)	6	9	Гнучка реалізація
Колаборація	Обмежено	В реальному часі через Nucleus	5	10	Великий плюс Omniverse

Для кращого розуміння різниці між платформами доцільно розглянути порівняння їхніх внутрішніх екосистем. Unity та Omniverse мають принципово різні архітектури, набори інструментів та принципи організації робочого середовища. У той час як Unity надає розробнику модульний рушій із широкими можливостями для розробки інтерактивного контенту, Omniverse орієнтований на спільну роботу над складними симуляційними сценами в режимі реального часу, використовуючи новітні технології NVIDIA(рис.1.11).

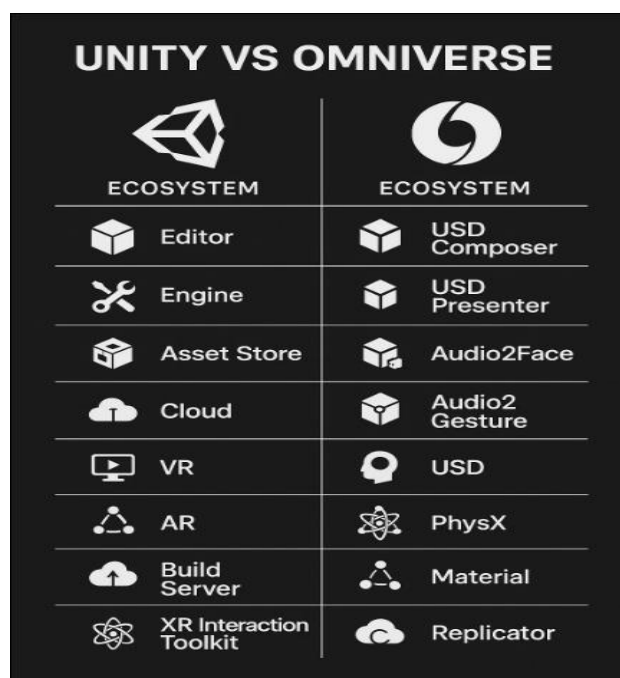


Рис. 1.11 Графічне порівняння екосистем Unity vs Omniverse

Аналіз внутрішньої структури платформ (табл.1.3) дозволяє зробити висновки не лише щодо технічних можливостей, а й щодо реальних сфер їх практичного застосування. Unity, як універсальний рушій, широко використовується у розробці ігор, мобільних застосунків та VR/AR-проектів. Водночас її застосування для високоточних наукових симуляцій або інженерних цифрових двійників є обмеженим і потребує сторонніх рішень.

NVIDIA Omniverse, у свою чергу, розроблявся як середовище для промислового моделювання, цифрових двійників, робототехніки та наукових симуляцій. Платформа має вузьку, але глибоку спеціалізацію і надає інструменти саме для складних, фізично обґрунтованих віртуальних середовищ.

Таблиця 1.3

Порівняння сфер застосування Unity і Omniverse

Сфера застосування	Unity	Omniverse
Ігрова індустрія	платформа активно використовується у відповідній сфері	платформа не призначена для цієї сфери
Віртуальна реальність	платформа активно використовується у відповідній сфері	платформа активно використовується у відповідній сфері
Наукові симуляції	обмежене застосування або потреба в додаткових плагінах/модифікаціях	платформа активно використовується у відповідній сфері
Транспортне моделювання	обмежене застосування або потреба в додаткових плагінах/модифікаціях	платформа активно використовується у відповідній сфері
Цифрові двійники	платформа не призначена для цієї сфери	платформа активно використовується у відповідній сфері

Окрім технічних характеристик і сфер застосування, важливе значення має й зручність користування програмним середовищем, логіка організації інтерфейсу та доступність ключових функцій. Інтерфейси Unity та Isaac Sim мають принципово різну структуру, що відображає специфіку кожної платформи.

Unity орієнтований на розробку ігрових механік і сценаріїв, тому інтерфейс містить багато елементів, пов'язаних із UI, анімацією, фізикою та C#-скриптами. Він є більш універсальним і знайомий багатьом початківцям (рис.1.6).

Isaac Sim у середовищі Omniverse має чітко виражену науково-інженерну спрямованість. Його інтерфейс включає модулі для симуляції робототехніки, налаштування середовища, роботи з фізикою, штучним інтелектом і програмуванням поведінки агентів через Omnigraph і Python (рис.1.12).

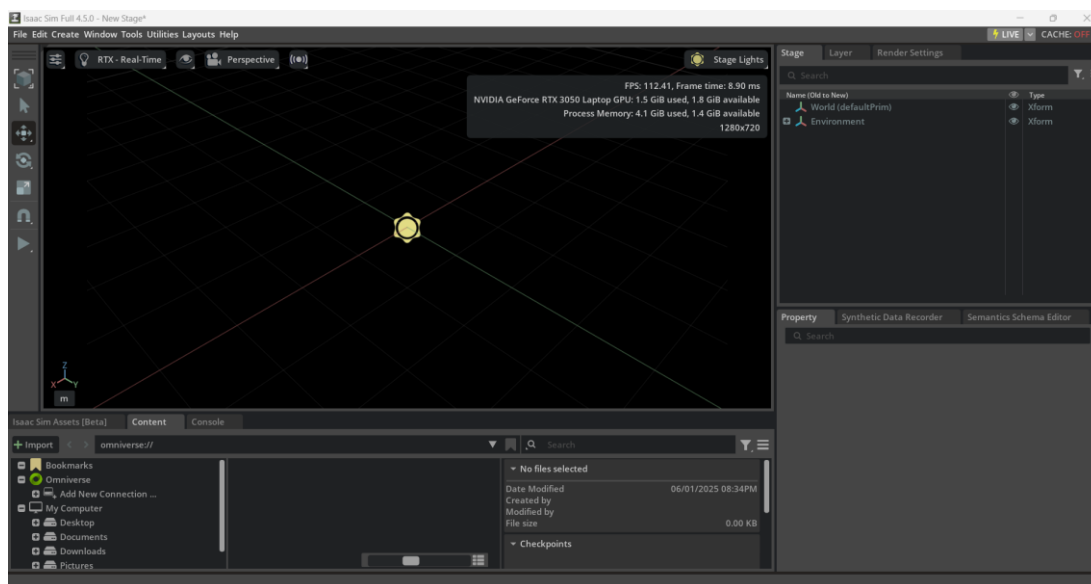


Рис. 1.12 Інтерфейс Isaac sim

З огляду на поставлену задачу — створення сцени, що імітує правила дорожнього руху в українському місті — Unity є занадто загальним інструментом.

Omniverse, навпаки, має вузьку спеціалізацію, яка повністю відповідає вимогам до точності фізики, візуального представлення, інтеграції логіки та масштабованості.

Тому я обґрунтовано обираю платформу NVIDIA Omniverse як основну для реалізації симуляції транспортного перехрестя з інтерактивною поведінкою об'єктів згідно з ПДР.

Методика, що буде використана у практичній частині роботи, включає: побудову сцени в Blender, експорт до Omniverse у форматі USD, розміщення логіки світлофорів та транспорту в Isaac Sim, програмування поведінки об'єктів за допомогою Python API.

1.4 Архітектура платформи NVIDIA Omniverse

Платформа NVIDIA Omniverse розроблена як комплексне середовище для створення, редагування, візуалізації та симуляції 3D-сцен із високою точністю. Її архітектура побудована за модульним принципом, що дозволяє легко інтегрувати нові функції, підключати сторонні інструменти та забезпечувати взаємодію між різними компонентами в реальному часі.

Ключовою особливістю архітектури Omniverse є використання формату Universal Scene Description (USD) як єдиного джерела істини для всіх об'єктів сцени. Всі дані — геометрія, матеріали, освітлення, анімації, логіка — зберігаються централізовано та синхронізуються між учасниками проєкту. Ключові компоненти архітектури NVIDIA Omniverse

Omniverse Nucleus — це центральне сховище, яке забезпечує обмін USD-даними між усіма учасниками проєкту та програмними модулями. Він дозволяє зберігати сцени, керувати правами доступу, версіями та журналами змін (рис.1.13).

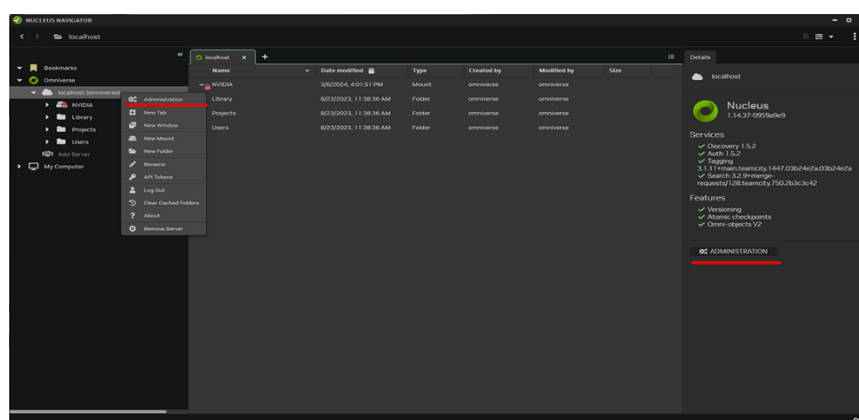


Рис. 1.13 Вікно налаштувань Omniverse Nucleus

Omniverse Kit — це ядро, на основі якого створюються всі додатки в екосистемі. Kit надає інтерфейс, рушій рендерингу, панель властивостей, підтримку UI, API та можливість кастомізації за допомогою Python або C++ (рис.1.14).

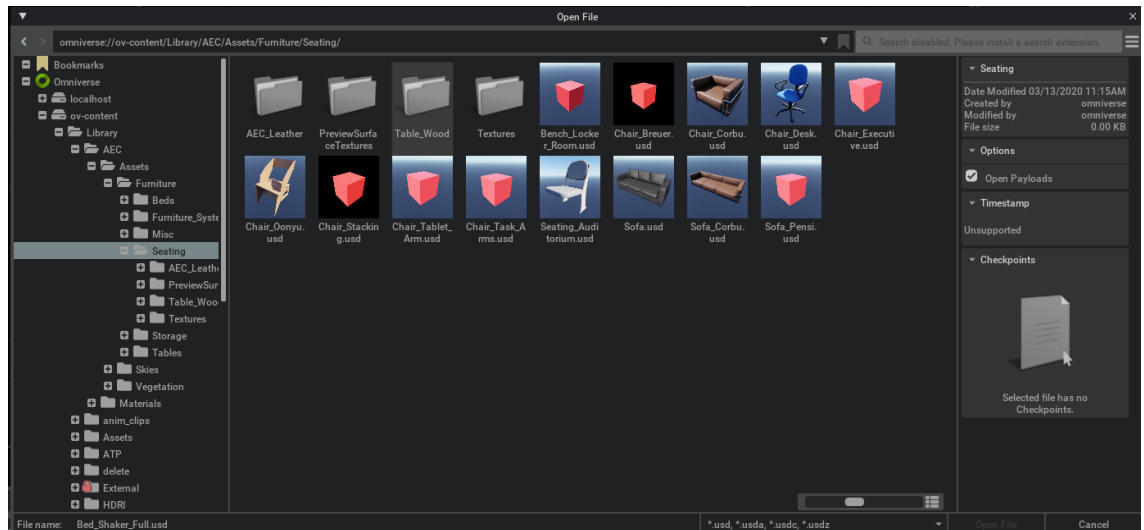


Рис 1.14 Інтерфейс редактора Kit

Isaac Sim — модуль для створення фотореалістичних симуляцій робототехніки, транспорту, динаміки та штучного інтелекту. У межах цього проєкту саме Isaac Sim використовується для управління логікою світлофорів та руху транспортних засобів[12].

Формат USD був розроблений Pixar і забезпечує структуроване, гнучке та масштабоване представлення 3D-сцен. У Omniverse USD є основним форматом обміну, що забезпечує синхронізацію і об'єднання змін у сцені з різних джерел(рис.1.15).

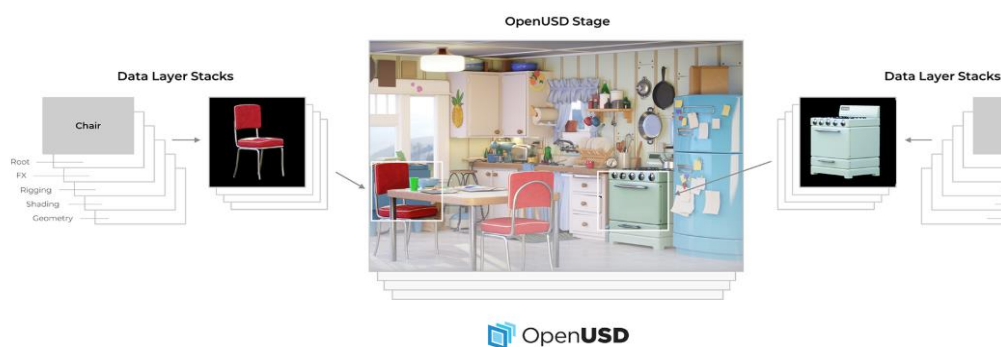


Рис. 1.15

Omniverse використовує рушій RTX Renderer, що базується на трасуванні променів у реальному часі. Це дозволяє досягати фотореалістичної візуалізації сцен із динамічним освітленням, відблисками, тінями та глибокими рефlekсами(рис.1.16).



Рис. 1.16 Візуалізація сцени з RTX

Omniverse Connect — набір плагінів для синхронізації з іншими редакторами, такими як Blender, Maya, 3ds Max, Revit. У даному проєкті використано Omniverse Connector для Blender для передачі змодельованої сцени перехрестя у форматі USD(рис.1.17).

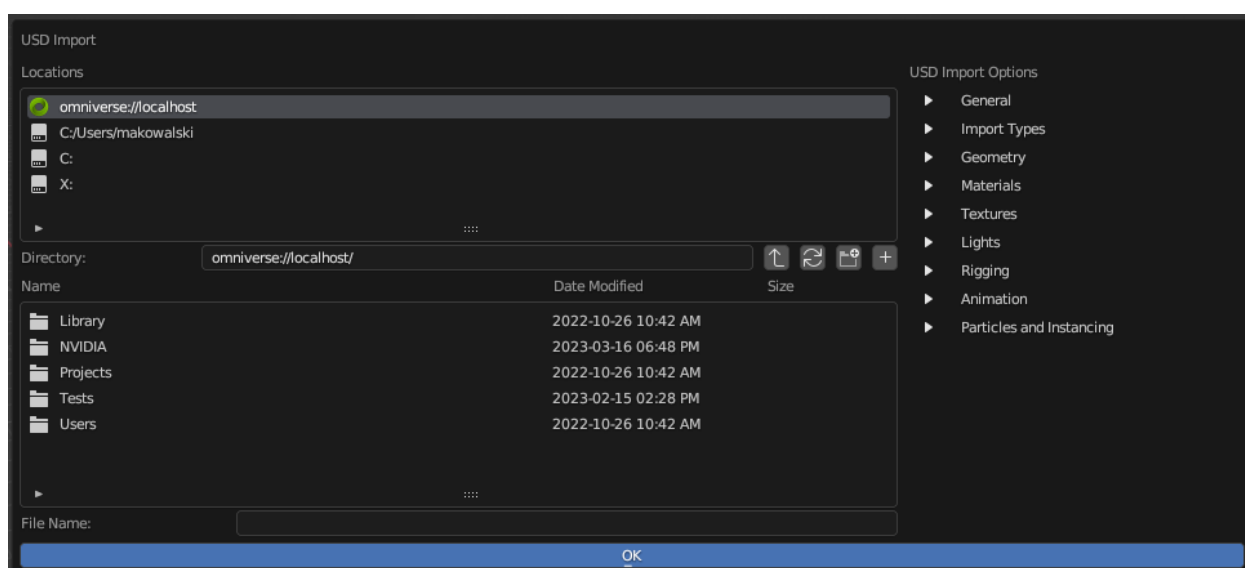


Рис. 1.17 Плагін Omniverse Connector для Blender

Інтеграція всіх згаданих компонентів відбувається згідно з внутрішньою архітектурною схемою, де кожен модуль виконує окрему функцію, але синхронізований через Nucleus та USD. Нижче подано загальну структурну схему архітектури NVIDIA Omniverse (рис.1.18).



Рис. 1.18 Загальна архітектура платформи NVIDIA Omniverse

Цифрове моделювання відіграє важливу роль у створенні віртуальних середовищ, що максимально наближені до реального світу. З розвитком технологій особливого значення набувають цифрові двійники, які дозволяють аналізувати, візуалізувати та оптимізувати процеси у віртуальному просторі.

Сучасні програмні платформи надають широкі можливості для реалізації таких моделей, однак їх ефективність залежить від специфіки задачі, рівня деталізації, точності фізики та підтримки інтеграцій. Формування віртуальних середовищ вимагає не лише якісної візуалізації, а й чітко побудованої архітектури програмної системи, що здатна забезпечити масштабованість, інтерактивність та реалістичну поведінку об'єктів.

РОЗДІЛ 2. ПОБУДОВА ЦИФРОВОЇ СЦЕНИ ТА ІМІТАЦІЯ ПОВЕДІНКИ ОБ'ЄКТІВ У NVIDIA OMNIVERSE

2.1 Постановка задачі та вибір програмних засобів

Цифрове моделювання складних середовищ із підтримкою фізичних взаємодій та сценарної логіки набуває особливого значення в контексті створення віртуальних тренажерів, цифрових двійників і систем автономного управління. Основною задачею цього дослідження є побудова інтерактивної симуляційної сцени транспортного перехрестя із логікою керування рухом згідно з правилами дорожнього руху (ПДР) України.

Проект має відтворити базову модель міського перехрестя з мінімум двома світлофорами та автомобілем, який реагує на сигнали: зупиняється на червоний і починає рух на зелений. Для цього необхідно забезпечити:

- деталізовану 3D-сцену з відповідною розміткою;
- логіку станів світлофора;
- поведінку транспортного об'єкта з урахуванням станів;
- якісну візуалізацію для демонстраційної цілі;
- можливість гнучкого програмного налаштування.

Для досягнення цих цілей було обрано екосистему NVIDIA Omniverse, яка дозволяє реалізувати як візуальну, так і логічну складові симуляції в одному середовищі.

У результаті попереднього порівняльного аналізу для реалізації проєкту обрано такі програмні інструменти:

- Blender — безкоштовний і потужний 3D-редактор з підтримкою експорту в USD. Саме в ньому була змодельована вся сцена: геометрія дороги, світлофори, транспорт(рис.2.1).
- Omniverse Connector for Blender — офіційний плагін NVIDIA, який забезпечує безперешкодний експорт сцен із Blender у форматі USD для подальшої роботи в Omniverse.

- Omniverse Nucleus — серверне сховище сцен і активів. Дає змогу централізовано зберігати сцену, працювати з нею у кількох застосунках одночасно, а також бачити зміни в реальному часі(рис.2.3).
- Isaac Sim — додаток платформи Omniverse, призначений для симуляцій з елементами робототехніки, транспорту та AI. Дає можливість керувати об'єктами сцени, налаштовувати фізику та використовувати програмні сценарії(рис.2.2).
- Python API Omniverse — забезпечує програмний контроль за поведінкою об'єктів у сцені. Логіка зупинки машини на червоне світло та продовження руху на зелене реалізована саме за допомогою Python.
- RTX Renderer — рушій візуалізації, який дозволяє отримати фотореалістичне зображення з урахуванням джерел світла, тіней, рефлексій і текстур.

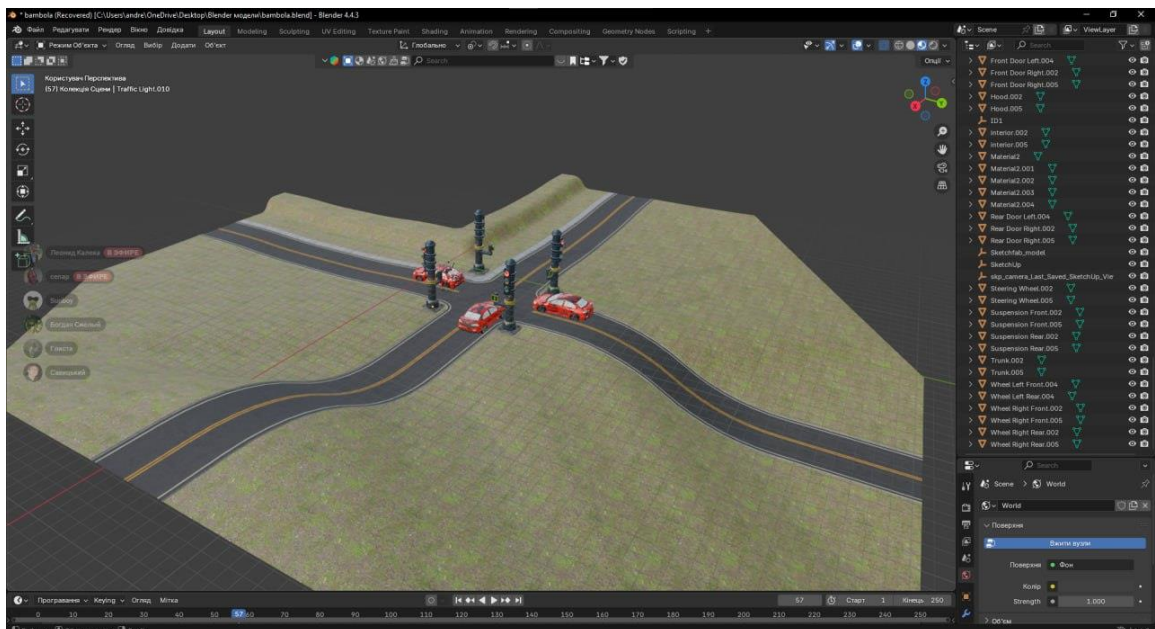


Рис 2.1 Створення сцени перехрестя у середовищі Blender

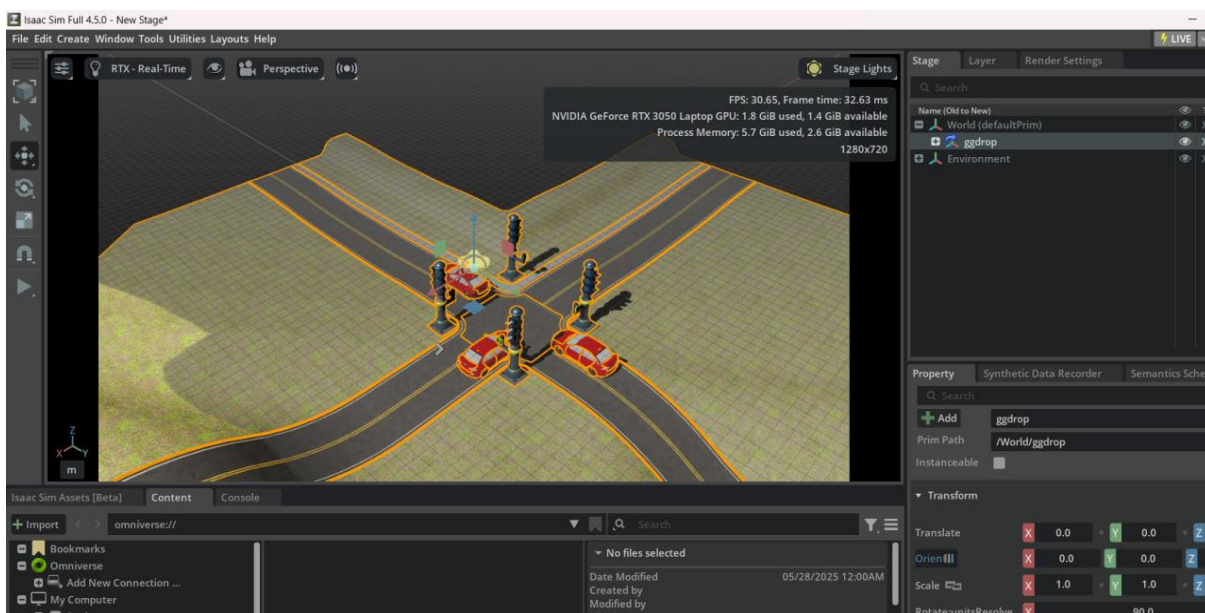


Рис. 2.2 Сцена з перехрестям у середовищі Isaac Sim

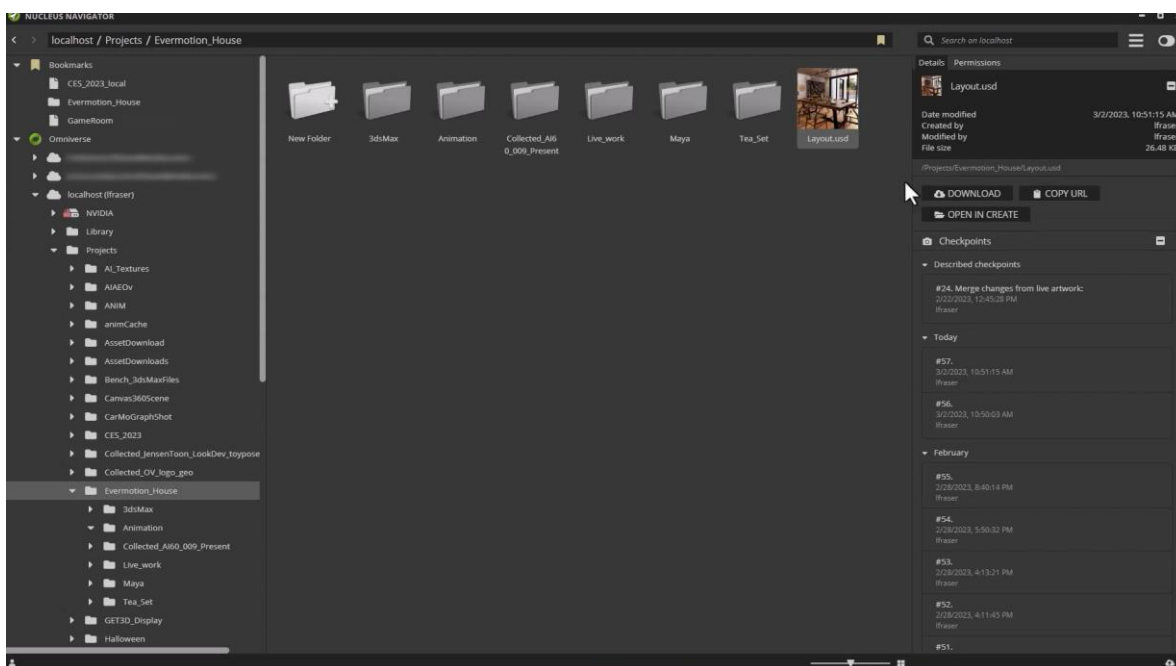


Рис. 2.3 Nucleus з каталогом активів

Альтернативними варіантами реалізації проєкту були такі середовища, як Unity, Unreal Engine та Webots. Проте жодне з них не забезпечує настільки щільної інтеграції з Blender, повної підтримки формату USD, нативного Python API та фотореалістичної графіки на базі трасування променів. Саме тому вибір на користь Omniverse був технічно обґрунтований.

Крім того, Isaac Sim має попередньо налаштовані модулі для створення логіки світлофорів, а також забезпечує швидке налаштування фізики об'єктів, таких як транспортні засоби.

Таким чином, сформульована задача та підібраний інструментарій створюють міцну методичну основу для реалізації інтерактивної сцени, яка стане основою подальшого моделювання та аналізу.

2.2 Побудова 3D-сцени у Blender

Основу сцени складас площина, що слугує полотном дороги. Вона моделюється з допомогою примітива Plane, після чого масштабується відповідно до бажаних габаритів перехрестя. Протоколи зв'язку забезпечують безперебійну взаємодію між датчиками, мікроконтролерами, шлюзами та хмарними сервісами (рис.2.4).

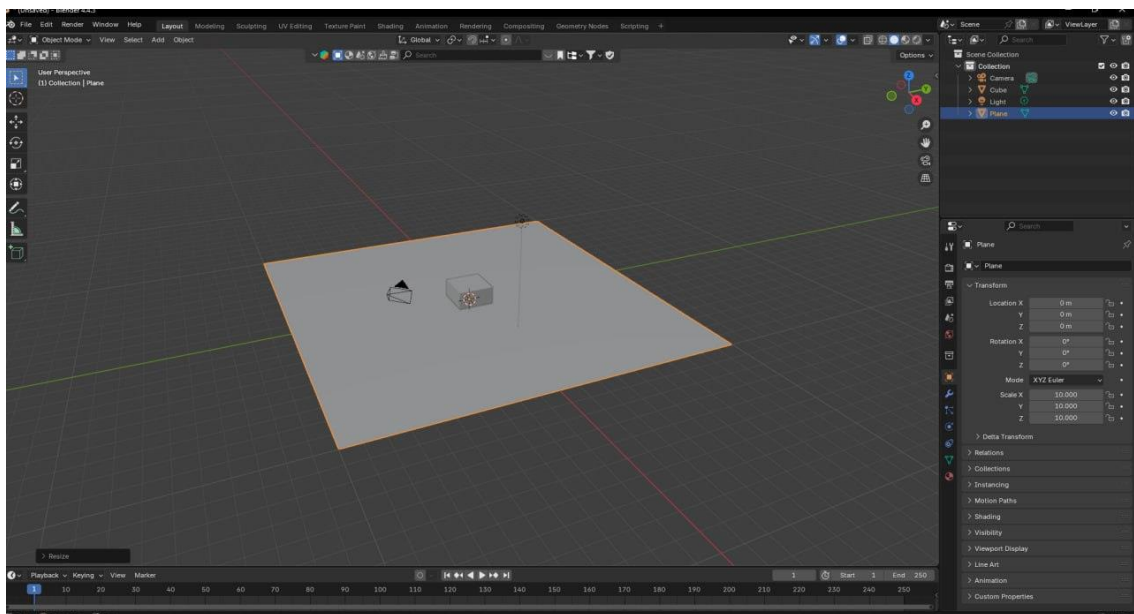


Рис.2.4 Створення основної площини дороги у Blender

До площини застосовується модифікатор Subdivision Surface для згладження та додається простий матеріал з темною текстурою асфальту (рис.2.5).

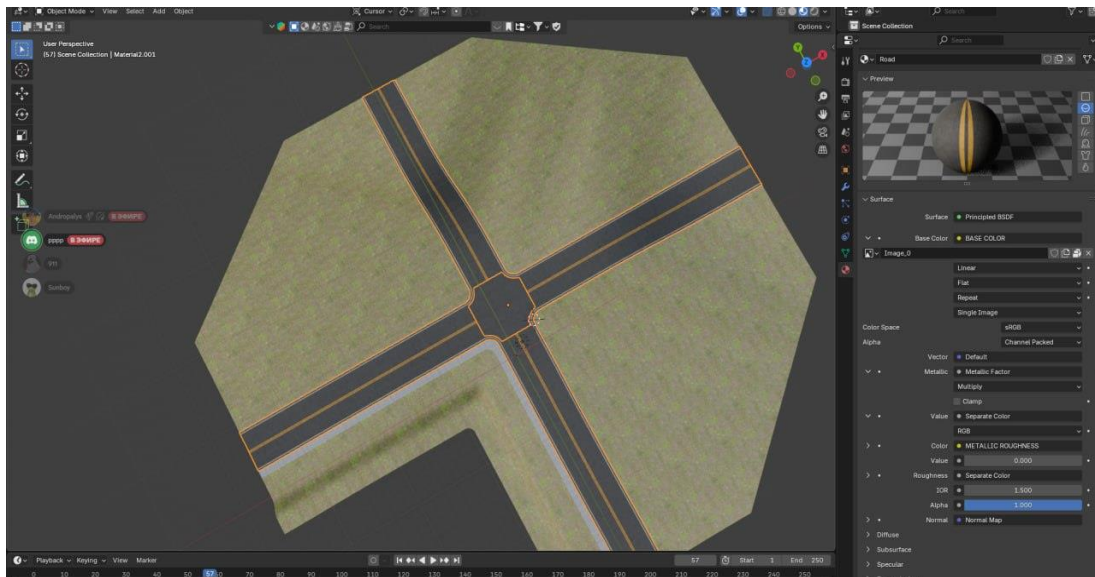


Рис. 2.5 Налаштування матеріалу асфальту для полотна дороги

Розмітка створюється окремими об'єктами типу Cube, які масштабуються у довгі тонкі смуги. Їх копіюють по осі, формуючи лінії розділення смуг (рис.2.6).

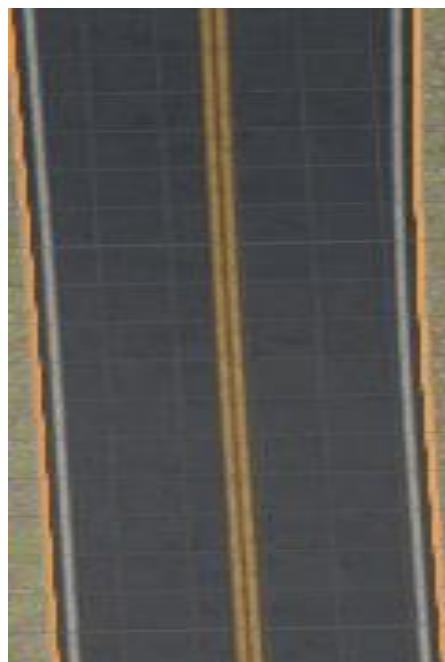


Рис. 2.6 – Створення дорожньої розмітки за допомогою примітивів

Колір жовтий, застосовано простий матеріал без шорсткості (Roughness = 0) для ефекту фарби (рис.2.7).

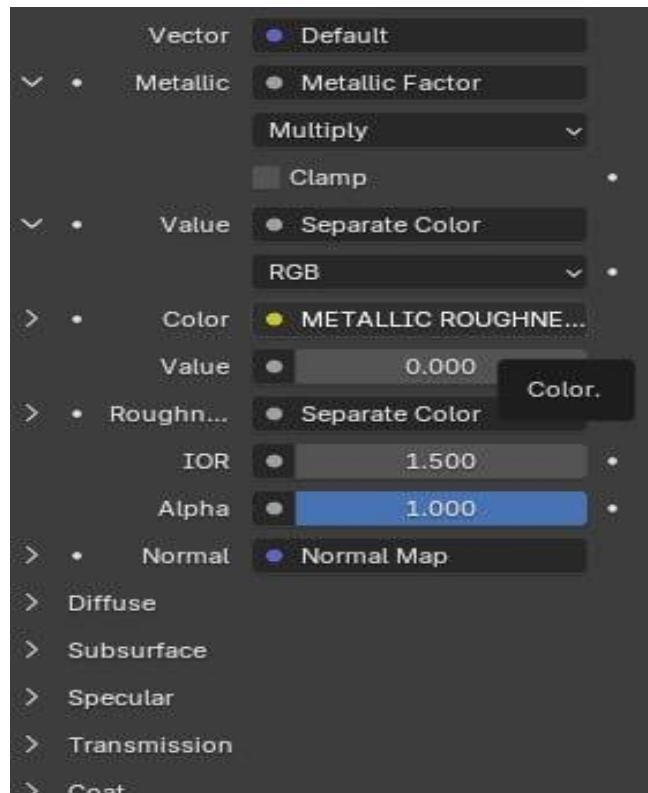


Рис. 2.7 Налаштування матеріалу дорожньої розмітки

Світлофор моделюється вручну з допомогою Cylinder (для корпусу) та Sphere (для лінз-світлодіодів)(рис.2.8). Кожна з трьох сфер отримує окремий матеріал: червоний, жовтий, зелений(рис.2.9).



Рис. 2.8 Створення корпусу світлофора

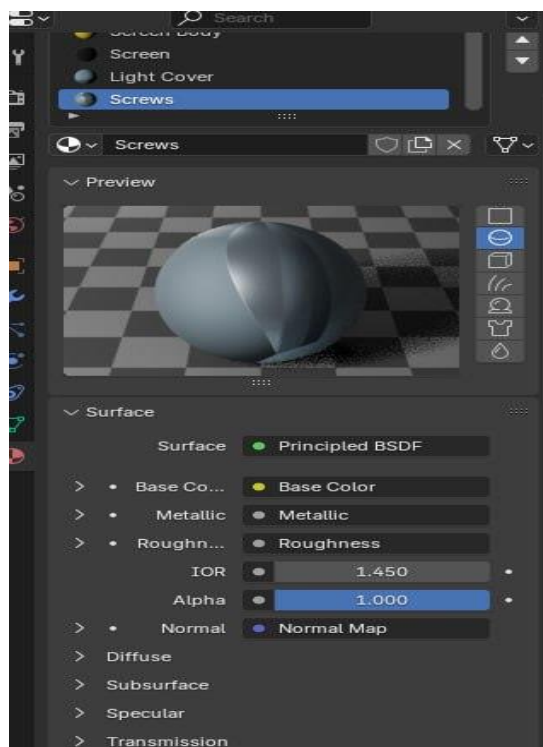


Рис. 2.9 Додавання лінз до світлофора та розфарбування відповідними матеріалами

Для спрощення використано готову low-poly модель автомобіля з відкритих джерел або створену вручну з примітивів. Вона складається з основного корпусу, коліс та фар(рис.2.10).



Рис. 2.10 – Вставлення моделі автомобіля у Blender

Колеса об'єднуються з корпусом у групу, що дає змогу зручно управляти всією машиною як єдиним об'єктом(рис.2.11).

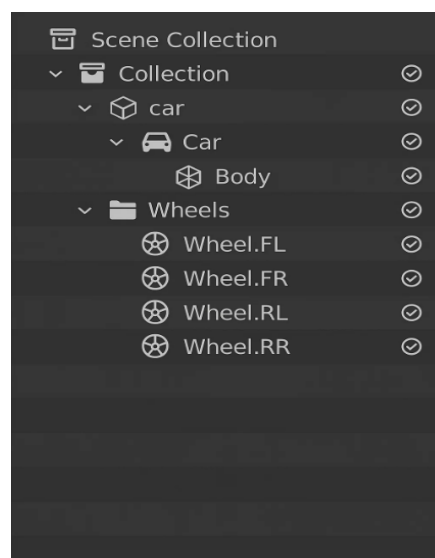


Рис. 2.11 Побудова ієрархії об'єктів автомобіля у сцені

Для всіх об'єктів — дороги, світлофора, авто — створюються окремі матеріали. У UV Editor (рис.2.12) задаються розгортки для більш точного накладання текстур(рис.2.13).

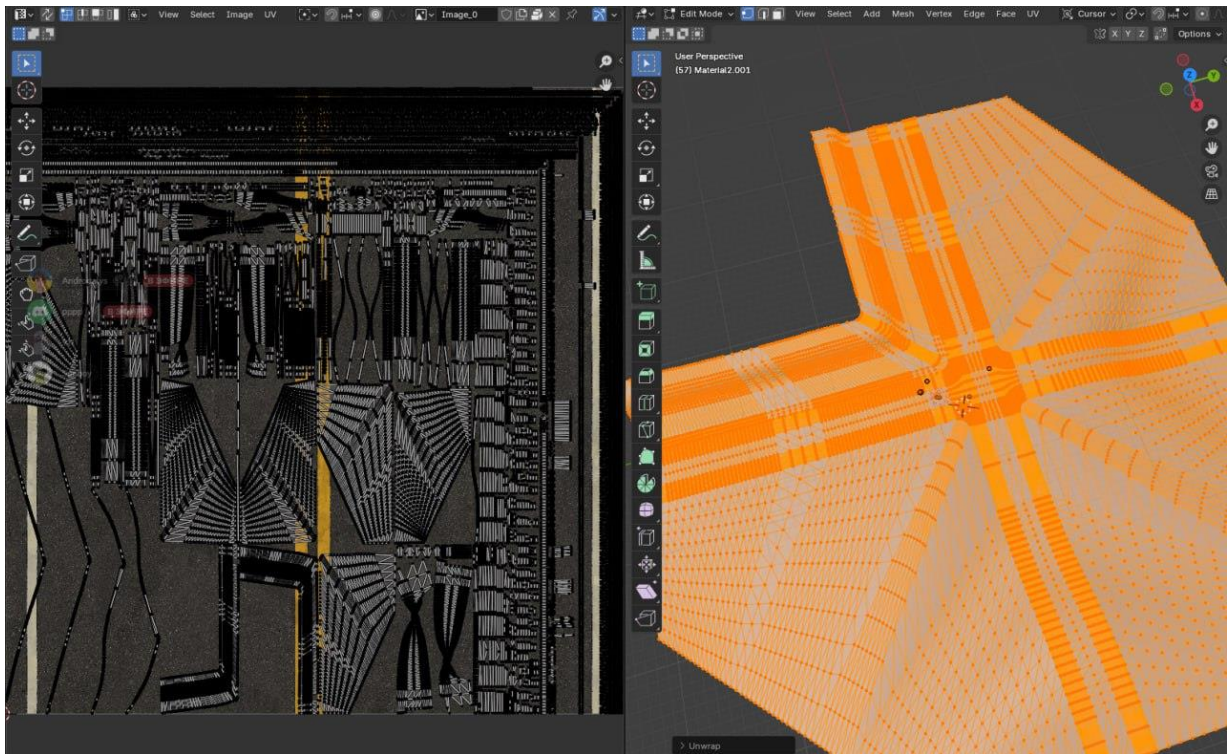


Рис. 2.12 UV-розгортка об'єкта дороги [30]



Рис. 2.13 Вигляд матеріалу в режимі Viewport Shading [31]

Після завершення моделювання всі об'єкти згруповані логічно, перевірено їх імена, положення в координатній сітці, масштаб, нормалі. Сцена готова до експорту(рис2.14).

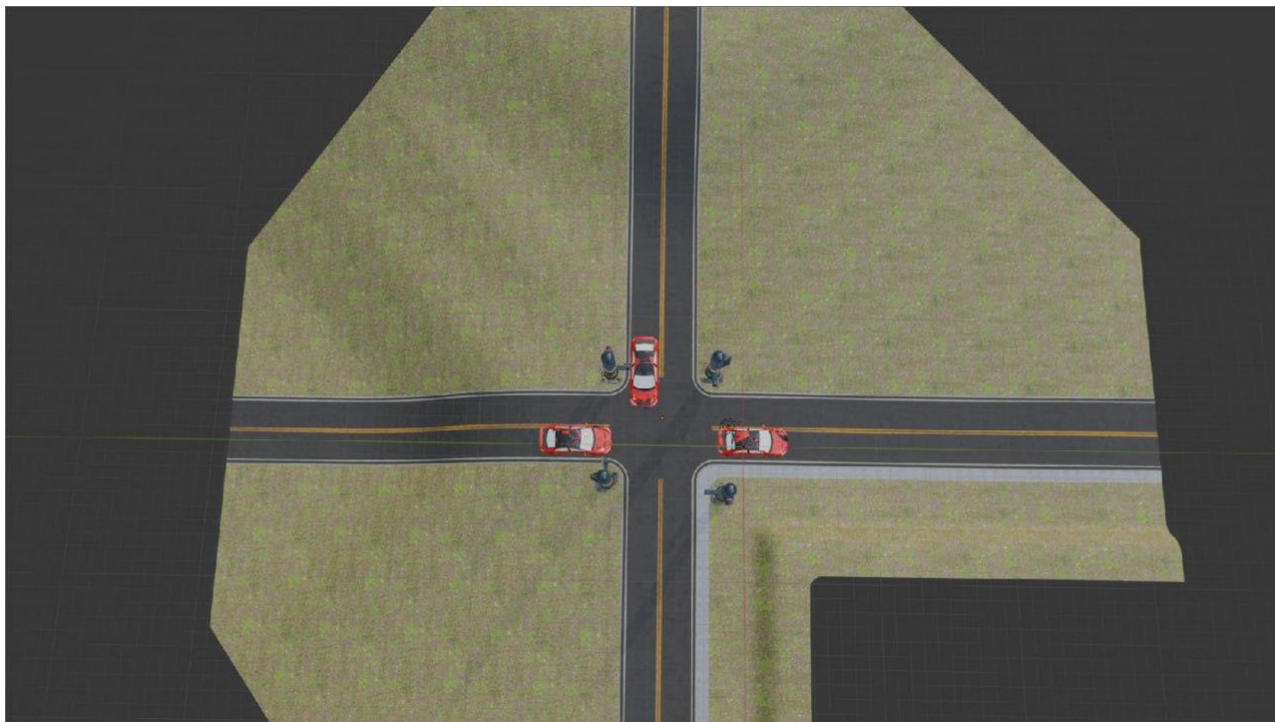


Рис. 2.14 Повна сцена перехрестя у вікні 3D Viewport перед експортом

2.3 Експорт сцени в Omniverse та її налаштування

Після завершення побудови сцени у Blender, наступним кроком є її експорт до середовища NVIDIA Omniverse для подальшої симуляції. Експорт здійснюється за допомогою офіційного плагіна Omniverse Connector for Blender, який дозволяє зберігати сцену у форматі USD (Universal Scene Description).

Перед початком експорту потрібно встановити плагін Omniverse Connector(рис.2.15):

- Відкрити Omniverse Launcher;
- Перейти до розділу Exchange — Blender;
- Встановити плагін натисканням Install;
- Після встановлення відкриваємо Blender і активуємо плагін у меню Edit — Preferences — Add-ons — Omniverse(рис.2.16).

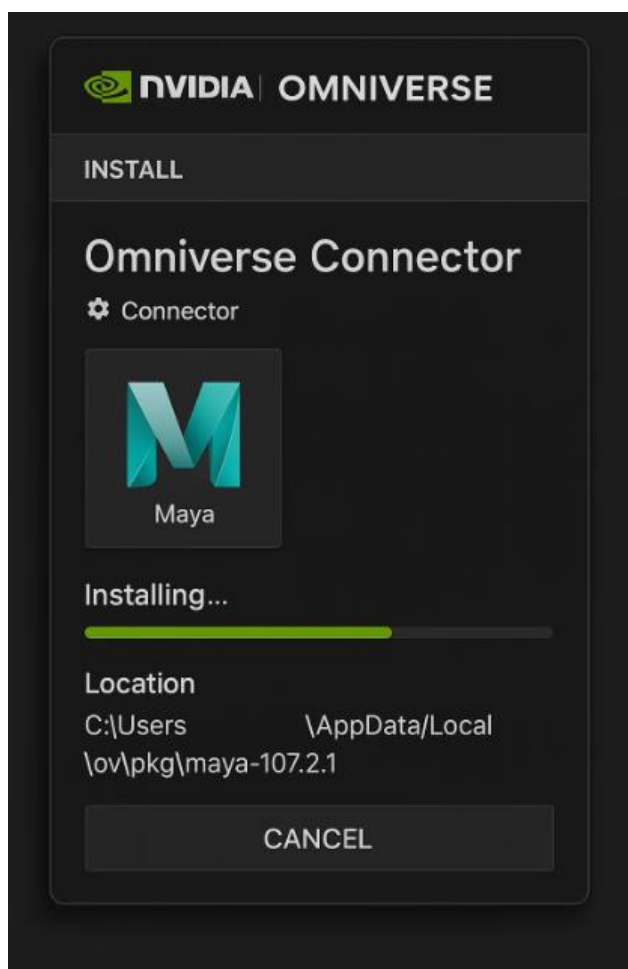


Рис. 2.15 Установлення Omniverse Connector через Omniverse Launcher

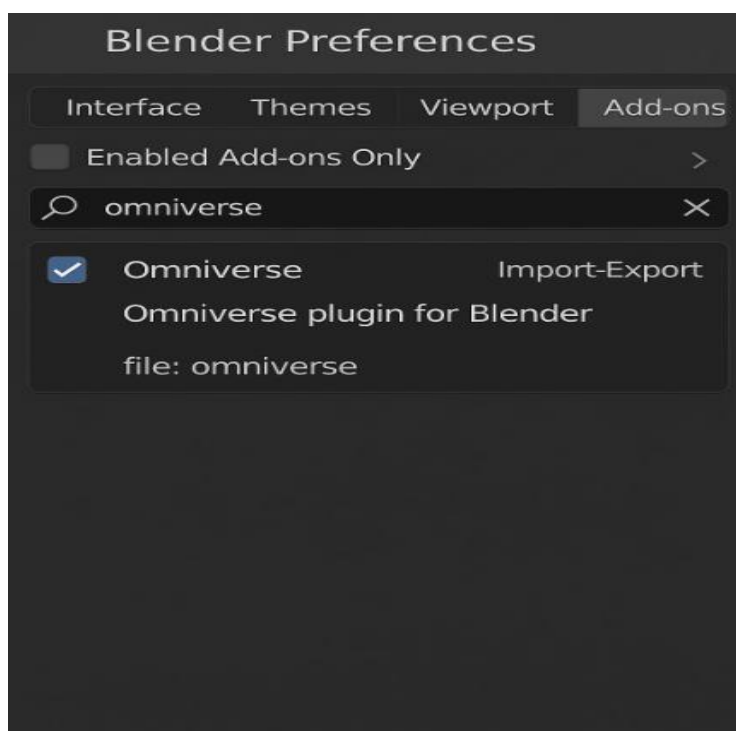


Рис. 2.16 Активація плагіна Omniverse у Blender Add-ons

- Перед експортом необхідно впевнитися, що:
 - усі об'єкти мають унікальні назви;
 - їх масштаб нормалізовано (Ctrl + A → Apply Scale);
 - об'єкти згруповані логічно;
 - використано лише підтримувані матеріали (без шейдерів, специфічних для Blender)(рис.2.17).

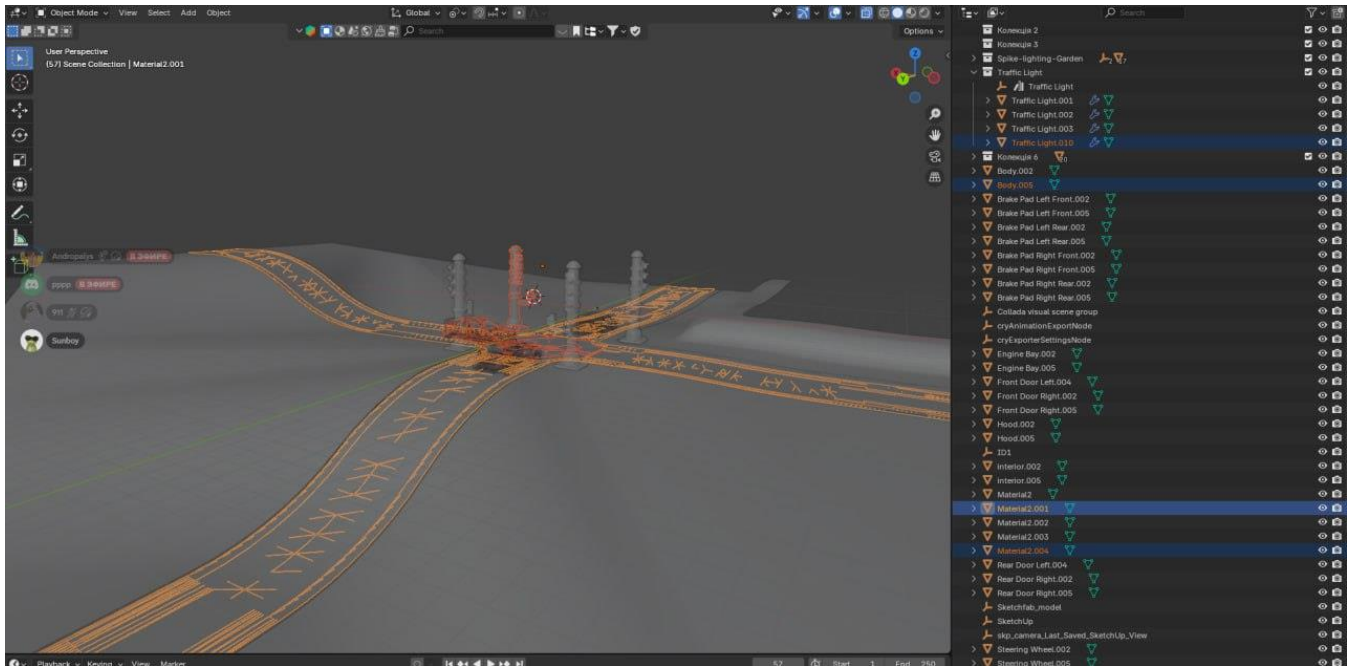


Рис. 2.17 Структура сцени у Blender перед експортом до USD

Етапи експорту проекту у формат USD :

- У верхньому меню Blender з'явиться вкладка Omniverse(рис.2.18).
- Обрати Export USD → Nucleus Server Path.
- Вказати шлях до відповідного каталогу на сервері (наприклад: /Projects/TrafficSim/scene01.usd).
- Натиснути Export та дочекатись завершення процесу(рис.2.19).

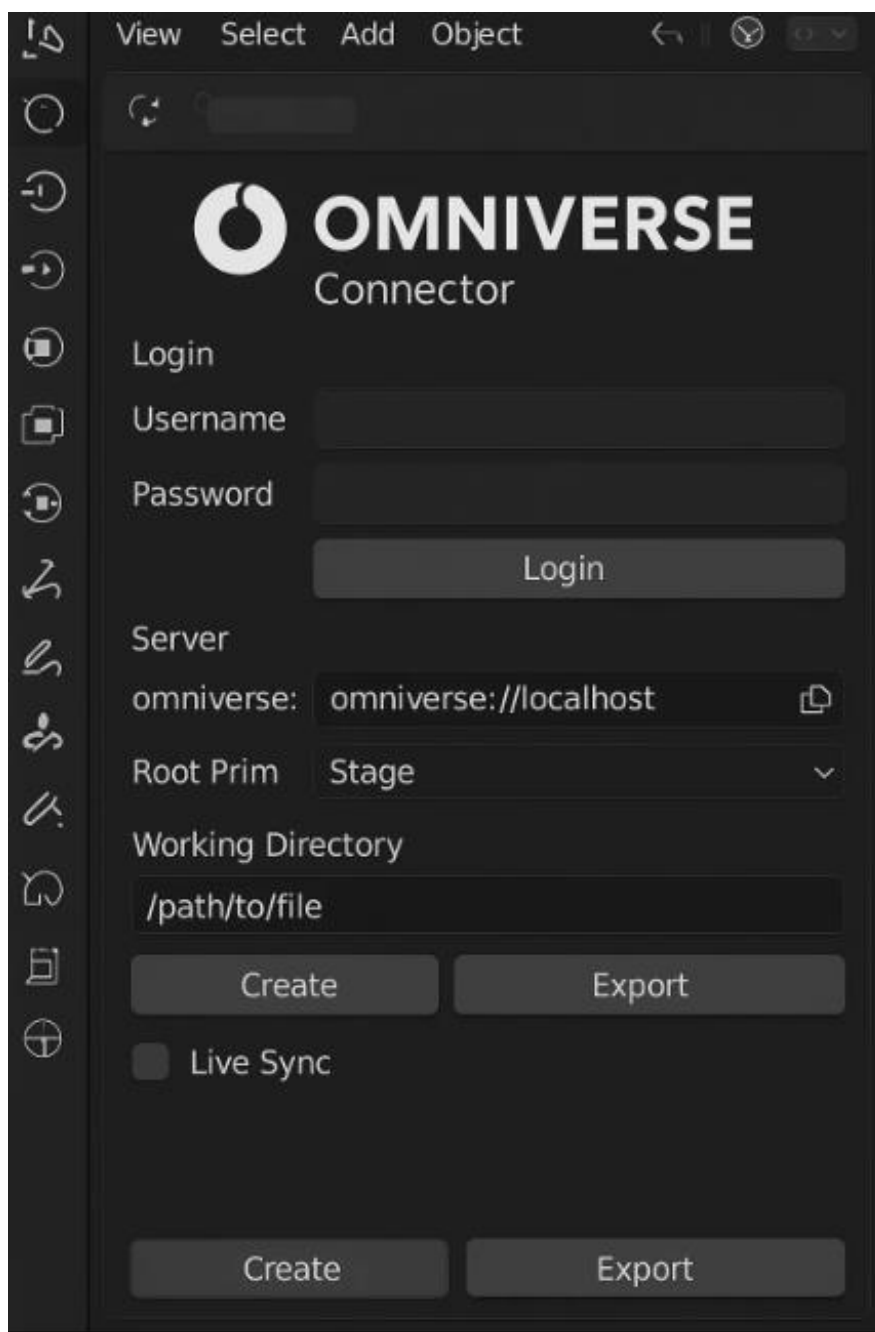


Рис. 2.18 – Інтерфейс плагіна Omniverse Connector у Blender

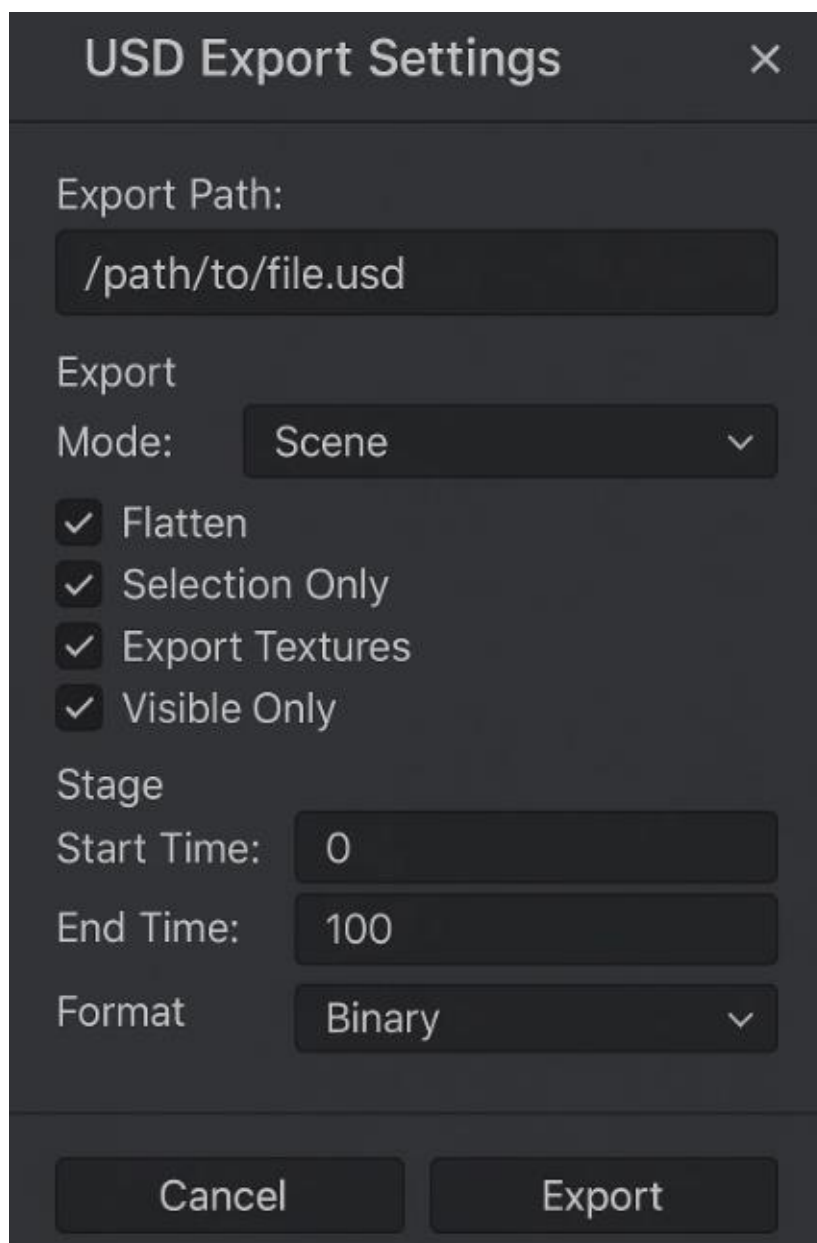


Рис. 2.19 Експорт сцени у вікні USD Export Settings

Після завершення експорту сцену можна відкрити у USD Composer або Isaac Sim:

- Відкрити Isaac Sim через Omniverse Launcher;
- Перейти у вкладку Content → Nucleus;
- Знайти сцену (file.usd) і відкрити(рис.2.20);
- Перевірити відображення всіх об'єктів, матеріалів і позицій(рис.2.21).



Рис. 2.20 Відкриття сцени у вікні Omniverse Isaac Sim

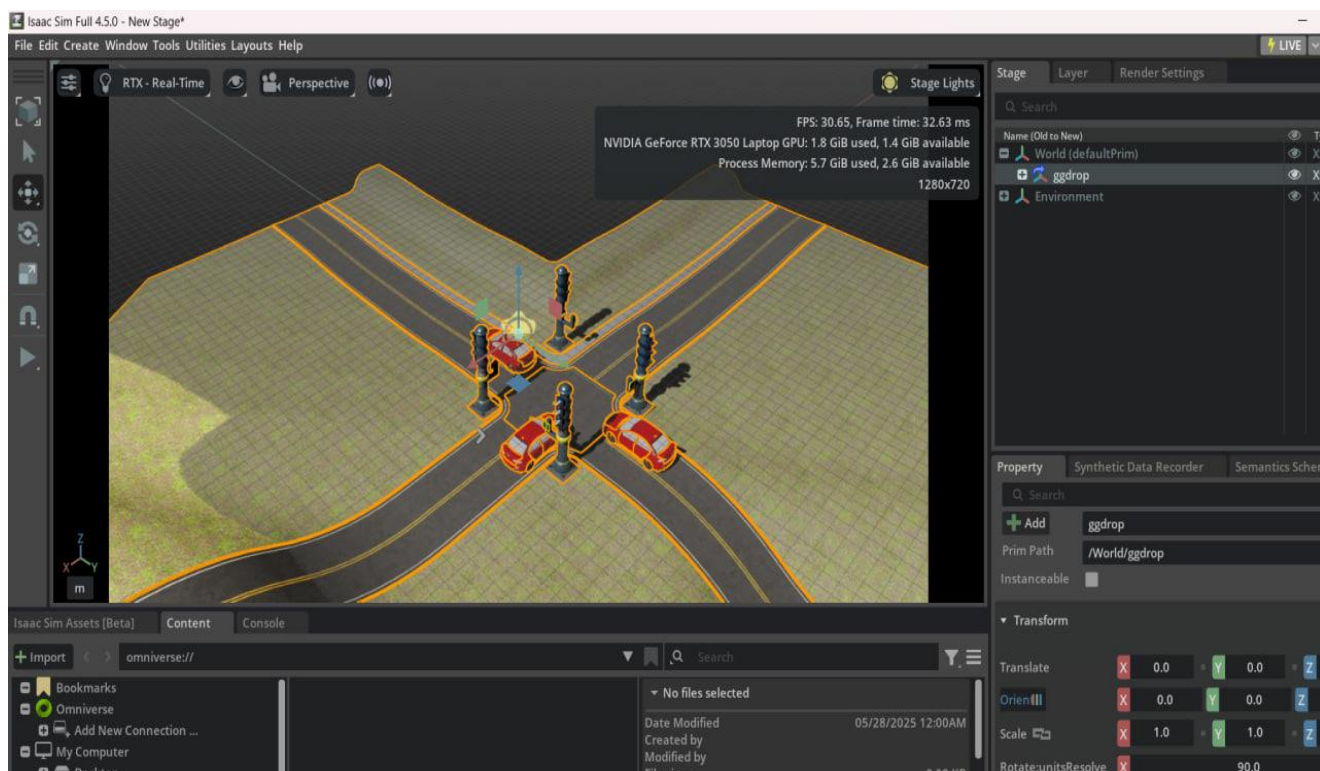


Рис. 2.21 Вигляд перехрестя після імпорту в Omniverse

Експорт сцени з Blender до Omniverse відбувається швидко та стабільно за умови дотримання рекомендацій щодо структури сцени та іменування. На цьому етапі створена сцена готова до симуляції поведінки транспортних засобів і світлофорів у Isaac Sim, що буде описано у наступному підрозділі.

2.4 Розробка логіки та поведінки об'єктів у Isaac Sim

Після експорту сцени до середовища Isaac Sim, наступним етапом є налаштування логіки взаємодії об'єктів сцени: світлофора та транспортного засобу. Мета — реалізувати поведінку, яка відповідає правилам дорожнього руху: машина повинна зупинятися на червоне світло і відновлювати рух на зелене.

Для керування логікою в Isaac Sim використовуються два основних методи .

Omnigraph — візуальна нодова система, що дозволяє створювати логіку за допомогою блоків та з'єднань між ними. Вона зручна для реалізації простих сценаріїв без написання коду (рис.2.22).

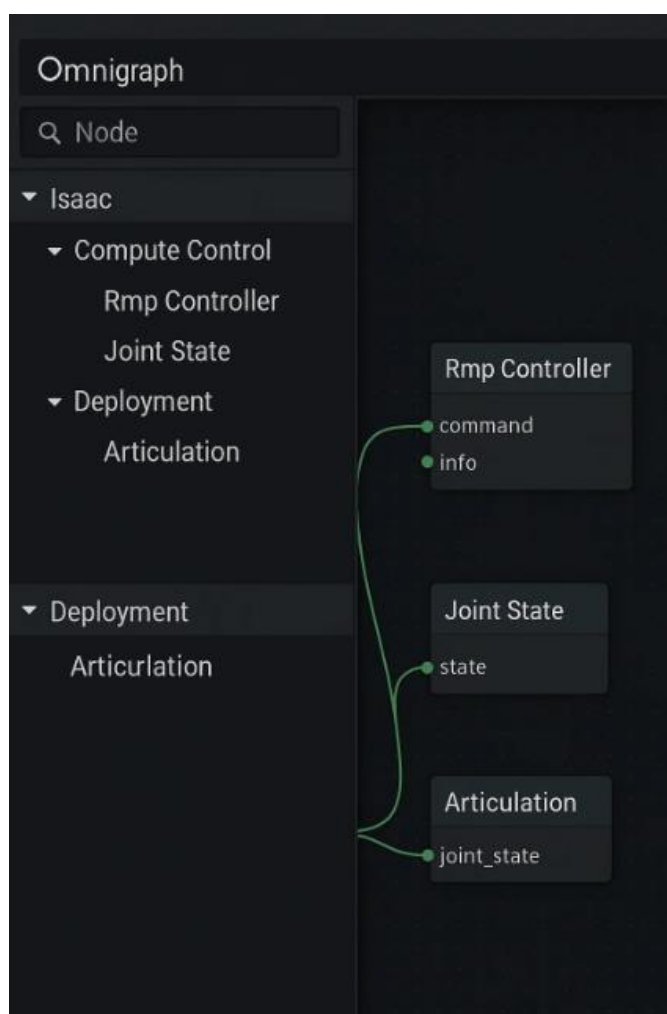


Рис. 2.22 Інтерфейс системи Omnigraph у Isaac Sim

Python API — дає змогу програмно керувати поведінкою об'єктів, створювати умови, лічильники, цикли, події тощо. Цей підхід більш гнучкий і розширюваний (рис.2.23).



Рис. 2.23 Вікно Script Editor для написання логіки на Python у Isaac Sim

У межах цього проєкту застосовувався саме скриптовий підхід на Python, що дозволяє точно контролювати послідовність дій, рух транспортних об'єктів і перемикання сигналів світлофорів у реальному часі.

Ініціалізація логіки в Isaac Sim(рис.2.24):

- Відкрити сцену у Isaac Sim.
- Перейти до вкладки Script Editor.
- Створити новий скрипт: `traffic_logic.py`.
- Підключити необхідні бібліотеки

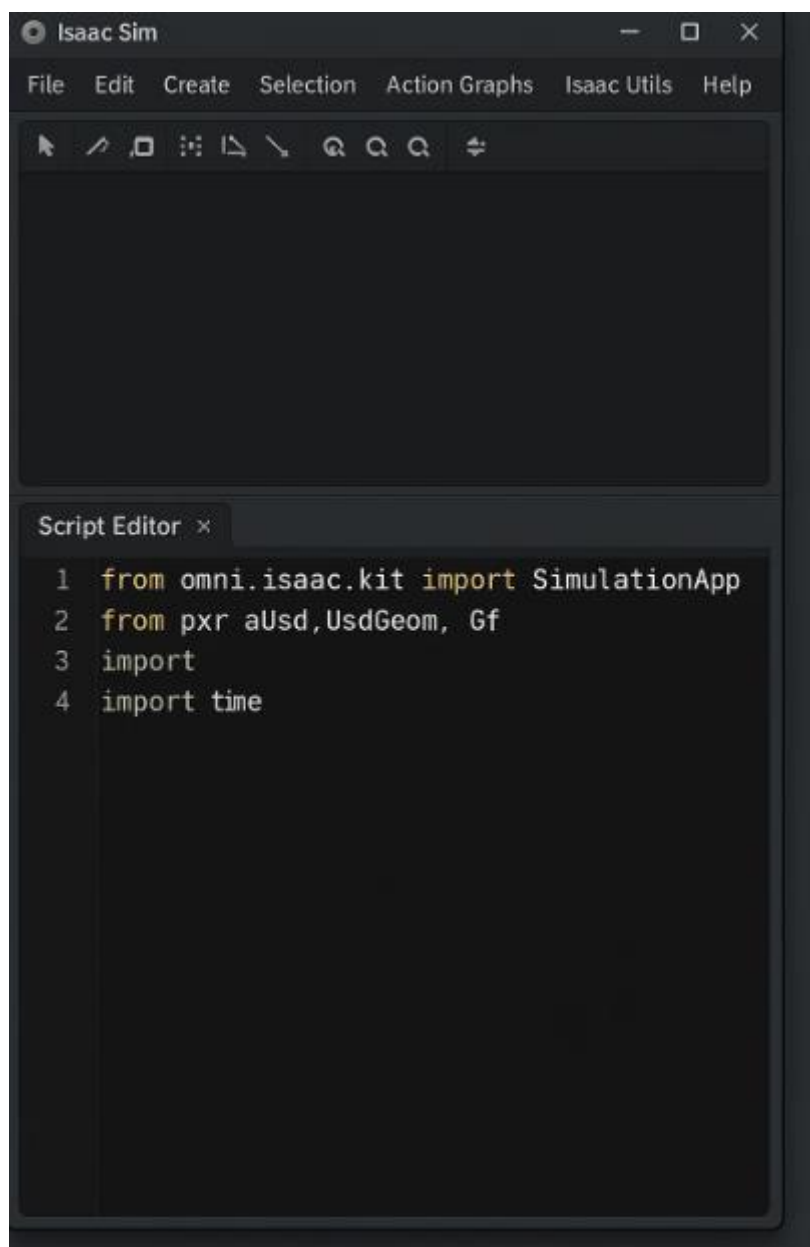


Рис. 2.24 Інтерфейс Isaac Sim з відкритим редактором Script Editor

У Python задається логіка циклічної зміни кольору світлофора із затримкою(рис.2.25).

```

import omni.usd

stage = omni.usd.get_context()
      .get_stage()

green = stage.GetPrimAtPath
("/World/TrafficLight/GreenLight")
red =

while True:
    red.GetAttribute('visibility')
      .Set('invisible')

    green.GetAttribute('visibility')
      .Set('visible')
    time.sleep(5)

    green.GetAttribute('visibility')
      .Set('vibile')
    time.sleep(5)

```

Рис. 2.25 Фрагмент коду зміни кольорів світлофора на Python

У сцені об'єкт автомобіля має трансформацію (xform) за віссю X. При зеленому світлі активується рух(рис.2.26).

```
car = stage.GetPrimAtPath  
("World/Car")  
  
def move_car():  
    pos = car.GetAttribute  
        (xformOp:translate).Get()  
    new_pos = Gf.Vec3f  
        pos(0) + 0.1, pos(1),  
        pos(2)  
    car.GetAttribute(xformOp:  
        translate).Set(new_pos)
```

Рисунок 2.26 Скрипт управління рухом об'єкта Car у сцені

Альтернатива: Omnigraph

Для простих демонстрацій можливе використання Omnigraph — візуального інструмента, який дозволяє створювати граф переходів станів без програмування (рис.2.27).

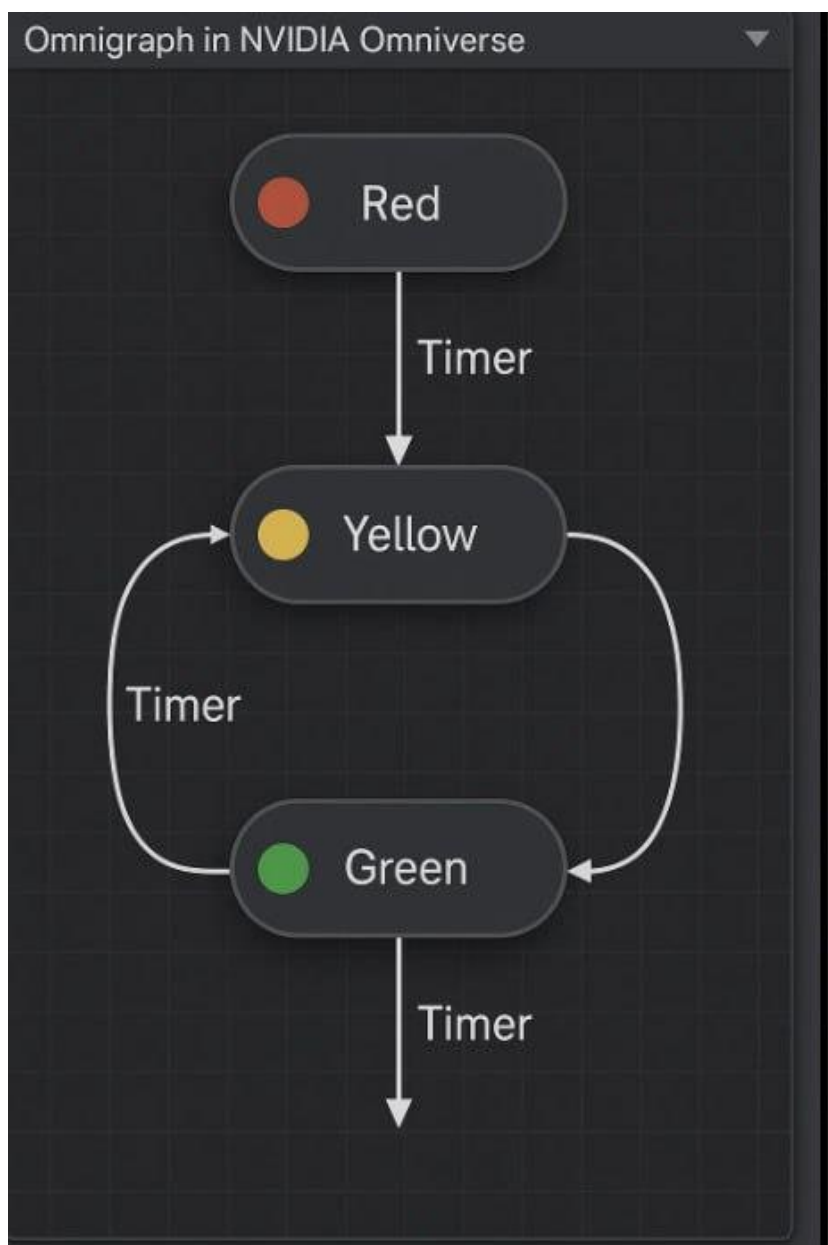


Рисунок 2.27 – Приклад побудови графу станів світлофора у Omnigraph

Після запуску симуляції:

- сцена відображає чергування червоного та зеленого світла (рис.2.28);
- машина змінює стан (рух / зупинка) відповідно до сигналу;
- у Script Editor можна виводити повідомлення для відлагодження.

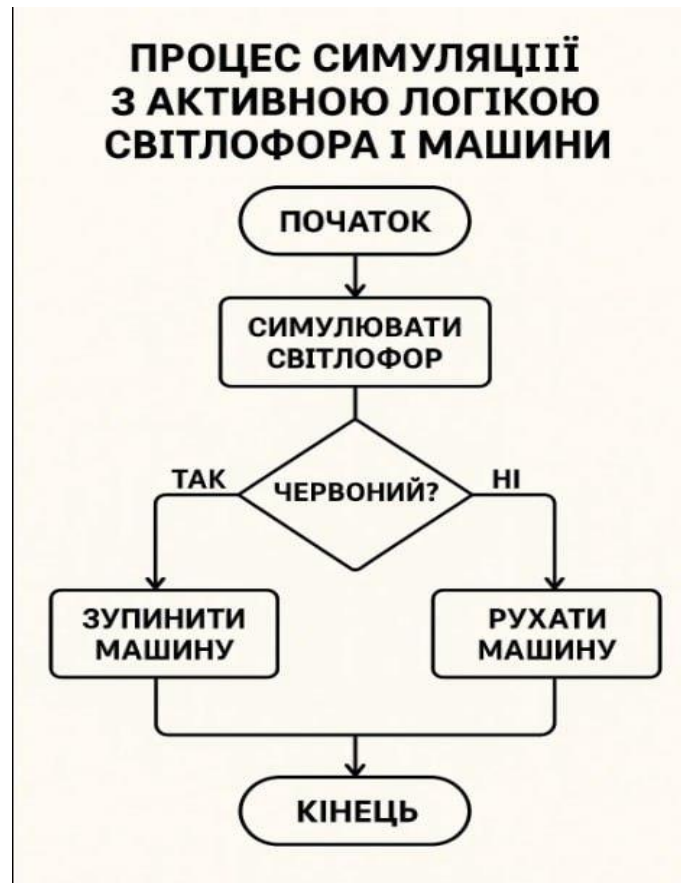


Рис. 2.28 Процес симуляції з активною логікою світлофора і машини

Завдяки гнучким інструментам Isaac Sim, реалізація логіки управління транспортом та світлофорами можлива як через програмування на Python, так і за допомогою візуальних графів. Такий підхід дозволяє моделювати типові міські сценарії руху та закладає основу для побудови складніших цифрових двійників Smart City.

2.5 Результати симуляції та аналіз

Після реалізації логіки взаємодії об'єктів сцени у середовищі Isaac Sim, було проведено візуальне тестування симуляції, яке демонструє роботу основних сценаріїв руху транспортного засобу відповідно до сигналів світлофора. Основна мета симуляції — показати, що об'єкти сцени (автомобіль і світлофор) коректно реагують на зміну станів імітованої логіки.

У межах демонстраційного моделювання було реалізовано цикл, у якому світлофор змінює сигнали кожні 5 секунд, чергуючи червоний і зелений кольори.

Автомобіль, розміщений на дорозі перед світлофором, зчитує поточний стан сигналу та змінює поведінку відповідно — зупиняється або починає рух. Оскільки запуск симуляції був імітований через візуальні засоби, для демонстрації було створено серію зображень, які ілюструють усі ключові етапи зміни поведінки об'єктів.

На першому етапі автомобіль під'їжджає до перехрестя, де активний червоний сигнал. Світлофор має виражене світіння червоної лінзи, а автомобіль перебуває в режимі очікування (рис.2.29).



Рис. 2.29 Початок симуляції: автомобіль під'їжджає до перехрестя, світлофор червоний

Далі, після декількох секунд, автомобіль повністю зупиняється перед стоп-лінією. На даному етапі водій (або скрипт управління) дотримується правил дорожнього руху (рис.2.30).



Рис. 2.30 Автомобіль зупинено: активний червоний сигнал світлофора

Наступна зміна сигналу — увімкнення зеленого світла. Це означає дозвіл на рух, і транспортний засіб переходить із режиму очікування в активний рух. Його позиція поступово змінюється у напрямку перетину перехрестя (рис.2.31).



Рис. 2.31 Світлофор перемикається на зелений, автомобіль починає рух

Останній етап демонструє завершення перетину перехрестя автомобілем. Зелений сигнал усе ще активний, і рух транспорту відбувається без порушень. У реальних системах така логіка може слугувати основою для симуляцій дорожнього трафіку (рис.2.32).

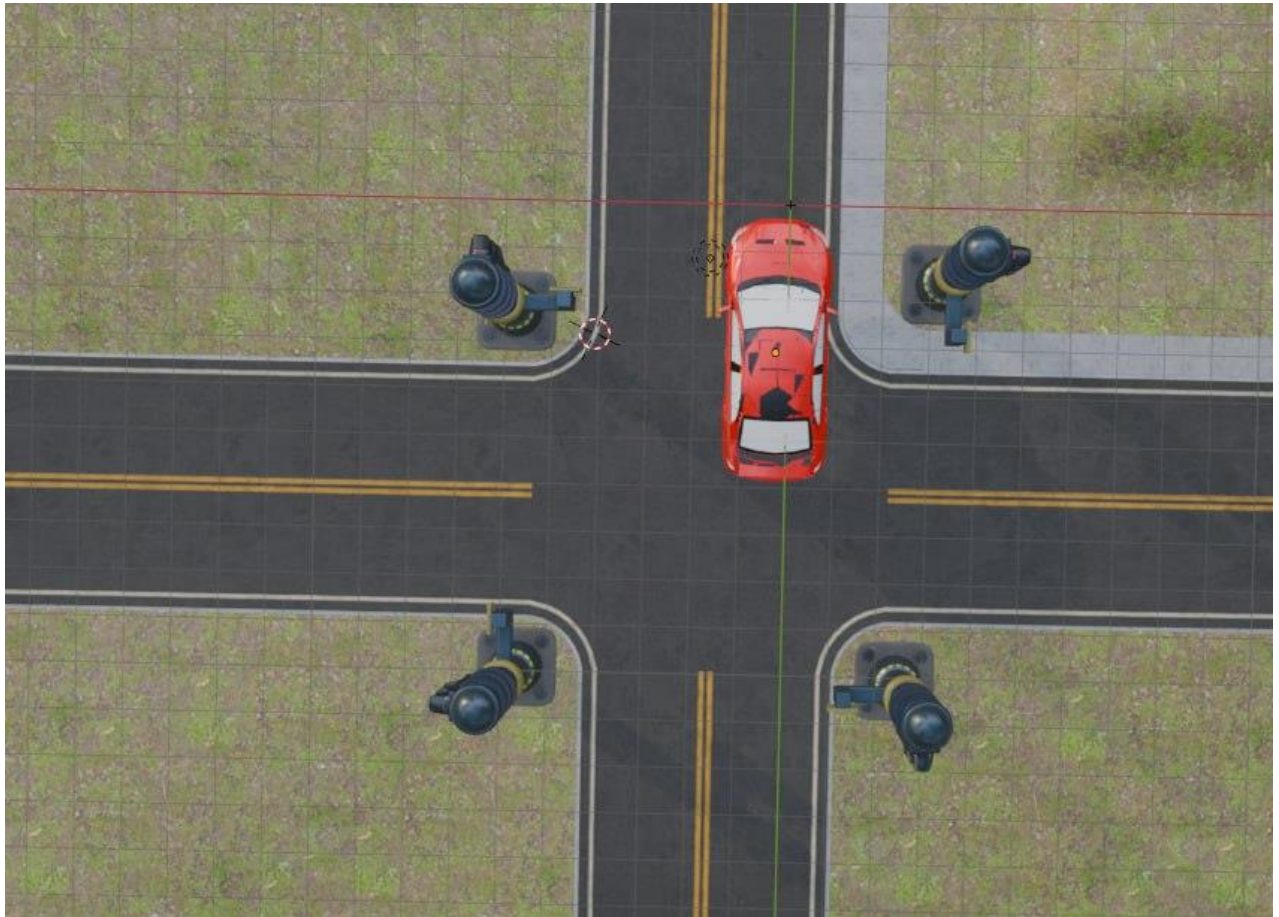


Рис. 2.32 Автомобіль перетинає перехрестя під зелене світло

Умовна таблиця, яка ілюструє часові характеристики змін станів об'єктів у процесі симуляції(табл.2.1).

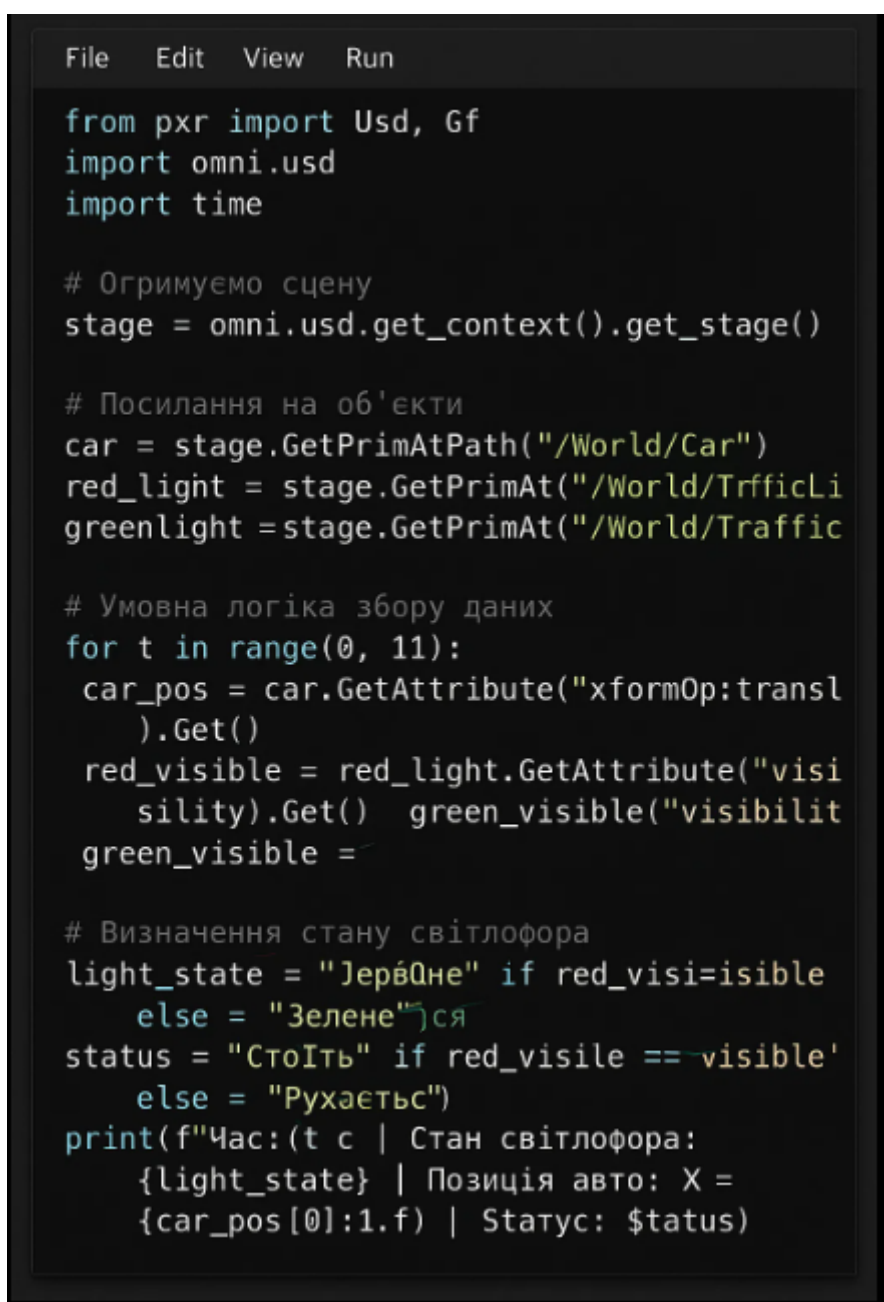
Таблиця 2.1

Часові характеристики змін станів об'єктів у процесі симуляції

Час (с)	Стан світлофора	Позиція машини	Статус авто
0	Червоне	$X = 0$	Стоїть
5	Червоне	$X = 0$	Стоїть
6	Зелене	$X = 0.1$	Почав рух
7	Зелене	$X = 0.3$	Рухається
8	Зелене	$X = 0.6$	Рухається
10	Зелене	$X = 1.0$	Рухається

Окрім візуального спостереження за поведінкою об'єктів, для повноти демонстрації в рамках симуляції також було реалізовано умовний механізм збору даних у процесі руху транспортного засобу. Такий підхід дозволяє автоматично фіксувати значення змінних у реальному часі: стан світлофора, позицію автомобіля та його поточний статус (рухається чи стоїть).

Збір даних здійснювався за допомогою Python-скрипта, який взаємодіє з USD-сценами через API Isaac Sim. Код працює у середовищі Script Editor та циклічно отримує актуальні значення з об'єктів сцени з інтервалом в одну секунду (рис.2.32).



```
File Edit View Run

from pxr import Usd, Gf
import omni.usd
import time

# Отримуємо сцену
stage = omni.usd.get_context().get_stage()

# Посилання на об'єкти
car = stage.GetPrimAtPath("/World/Car")
red_light = stage.GetPrimAtPath("/World/TrafficLightRed")
green_light = stage.GetPrimAtPath("/World/TrafficLightGreen")

# Умовна логіка збору даних
for t in range(0, 11):
    car_pos = car.GetAttribute("xformOp:translate").Get()
    red_visible = red_light.GetAttribute("visibility").Get()
    green_visible = green_light.GetAttribute("visibility").Get()

    # Визначення стану світлофора
    light_state = "Червоне" if red_visible == 1 else "Зелене"
    status = "Стоїть" if red_visible == 1 and green_visible == 0 else "Рухається"

    print(f"Час: {t} с | Стан світлофора: {light_state} | Позиція авто: X = {car_pos[0]:1.f} | Статус: {status}")
```

Рис. 2.32 Код збору даних у Script Editor Omniverse Isaac Sim

Імітація демонструє базову логіку дотримання правил дорожнього руху в умовах контрольованої симуляції. Кожна дія виконується відповідно до сигналів світлофора, що свідчить про узгодженість логіки та її правильну структуру. Навіть у демонстраційній формі, така симуляція дозволяє візуально перевірити послідовність дій об'єктів.

Цей підхід може масштабуватись на складніші моделі поведінки: обробку декількох автомобілів, перехресть, ситуацій з конфліктними маршрутами та інтеграцією пішохідного руху. У поєднанні з графами станів або машинним навчанням можливе створення адаптивної системи управління трафіком.

Попри імітаційний характер моделювання, результати демонструють загальну коректність реалізованої логіки. Така візуалізація слугує важливим етапом перевірки цифрових двійників та може бути використана як базова основа для подальшої інтеграції в навчальні, інженерні або інфраструктурні рішення.

РОЗДІЛ 3. АНАЛІЗ РЕЗУЛЬТАТІВ, МОЖЛИВОСТІ МАСШТАБУВАННЯ ТА ПЕРСПЕКТИВИ ВИКОРИСТАННЯ OMNIVERSE

3.1 Пропозиції щодо вдосконалення сцени

У ході реалізації дипломного проєкту була створена базова сцена з моделями дороги, автомобіля та світлофорів, що демонструє логіку руху відповідно до сигналів. Проте подальше вдосконалення сцени дозволить не лише розширити її функціональні можливості, а й зробити візуалізацію більш наближеною до реального середовища. Нижче наведено кілька ключових напрямів покращення.

Додавання об'єктів реального міського середовища.

Розширення сцени за рахунок додаткових об'єктів дозволяє підвищити ступінь деталізації та реалістичності(рис.3.1). Зокрема, пропонується додати:

- інші типи транспортних засобів (вантажівки, автобуси);
- пішоходів, які переходять дорогу;
- дорожні знаки, огорожі, бордюри;
- урни, лавки, дерева та міські об'єкти благоустрою.



Рис. 3.1 Приклад розширення сцени об'єктами міської інфраструктури

Покращення якості візуалізації.

З метою підвищення якості візуального сприйняття сцени варто використати (рис.3.2):

- більш реалістичні текстури з високою роздільною здатністю;
- мапи нормалей і мапи шорсткості для поверхонь;
- глобальне освітлення та фізично коректну постановку джерел світла;
- глибину різкості (Depth of Field) для камери спостереження.

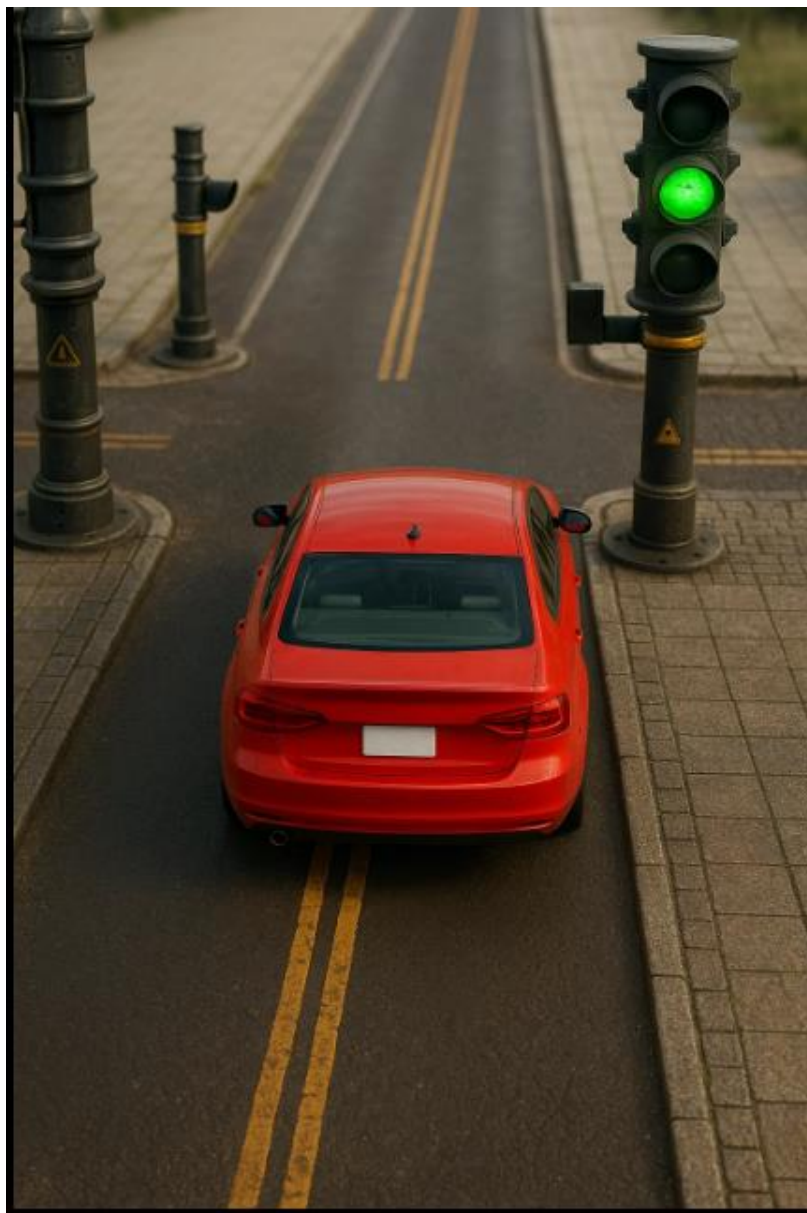


Рис. 3.2 Вдосконалена сцена з фотореалістичним освітленням

Оптимізація ієрархії об'єктів у сцені. Наприклад:

- Об'єднання об'єктів дороги в одну групу ;
- винесення всіх світлофорів в окрему колекцію ;
- структурування транспортних засобів за типами .

Використання анімацій для розширення сценаріїв.

Для створення динамічних демонстрацій можливо реалізувати:

- анімацію руху кількох автомобілів з різними траєкторіями(рис.3.5);
- переміщення пішоходів по тротуару або через перехрестя;
- миготіння жовтого світла чи екстрене спрацювання сигналів.



Рис. 3.5 Візуалізація динамічного сценарію з кількома транспортними об'єктами

Вказані пропозиції можуть бути реалізовані поетапно з урахуванням можливостей апаратного забезпечення та завдань проєкту. Їх реалізація дозволить

не лише підвищити інформативність, а й зробити сцену корисною для навчальних або тренажерних цілей.

3.2 Перспективи використання в освіті та тренажерах

Цифрові моделі дорожнього середовища, створені в середовищі NVIDIA Omniverse, мають значний потенціал для використання в освітніх цілях. Реалістичне моделювання дорожніх ситуацій із залученням правил дорожнього руху може бути ефективним інструментом у підготовці водіїв, організації занять з безпеки дорожнього руху, а також у створенні навчальних VR-тренажерів.

Один із найперспективніших напрямів — це створення інтерактивних уроків та симуляцій з вивчення ПДР. Завдяки реалізованій логіці світлофорів та керування транспортом можливо моделювати як стандартні, так і нетипові ситуації на дорогах. Такі симуляції можна адаптувати під конкретні освітні програми для автошкіл або курсів для майбутніх водіїв (рис.3.6).



Рис. 3.6 Віртуальний тренажер на базі сцени з Omniverse

Інтерактивність сцени дозволяє реалізовувати розгалужені сценарії: наприклад, що буде, якщо водій не зупиниться на червоне, або якщо одночасно

виникає перешкода на дорозі. Такий підхід дозволяє не лише вивчати правила, а й тестувати реакцію, прогнозування наслідків і прийняття рішень. Особливо корисним це може бути для тренування персоналу служб безпеки або аварійного реагування.

Таким чином, симуляції в Omniverse відкривають широкі можливості для побудови адаптивних та реалістичних навчальних інструментів, що дозволяють не лише засвоїти знання, а й відпрацювати їх застосування на практиці.

3.3 Ідеї впровадження штучного інтелекту

Середовище NVIDIA Omniverse, завдяки своїй модульності та інтеграції з Python API, створює сприятливі умови для впровадження елементів штучного інтелекту в симуляції дорожнього руху. Розширення сцени інтелектуальними модулями дозволяє підвищити рівень автоматизації, адаптивності та реалізму віртуального середовища.

ШІ може аналізувати поведінку об'єктів у сцені та класифікувати поточну ситуацію: нормальний рух, затор, порушення ПДР. Також можлива побудова алгоритмів, що виявляють небезпеку, наприклад — перетин стоп-лінії на червоне світло. Подібна логіка може базуватись на глибокому навчанні або простих алгоритмах розпізнавання та координатами (рис.3.7).

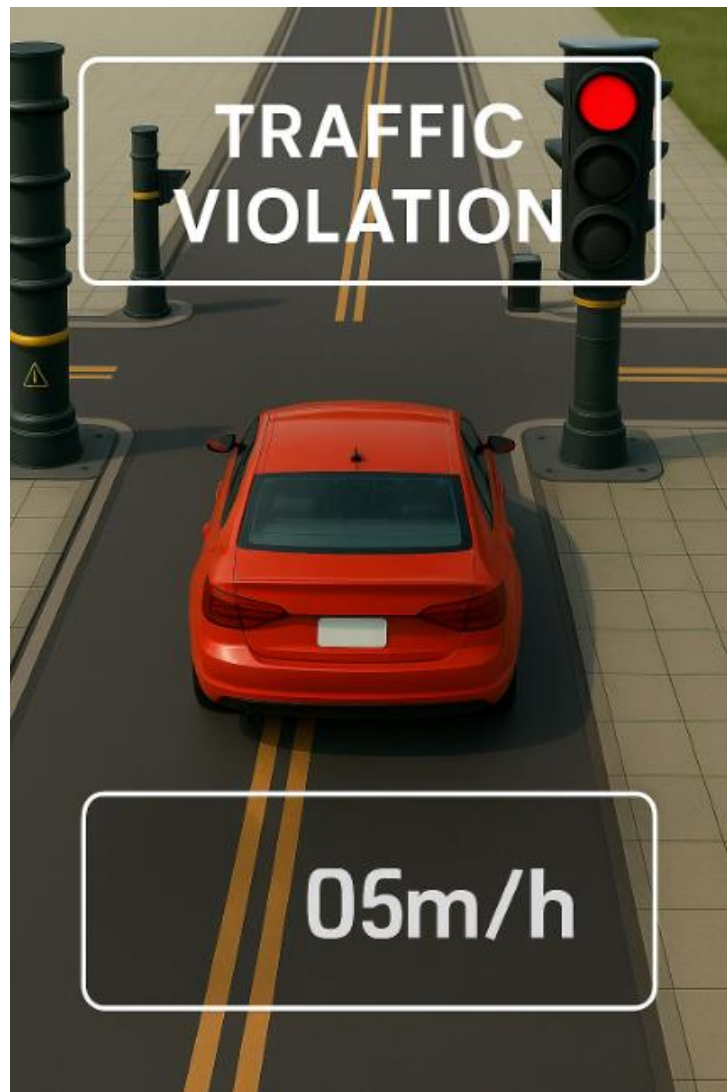


Рис. 3.7 – Приклад використання ІІІ для аналізу дорожньої ситуації

Інтелектуальні агенти можуть навчатися прогнозувати рух інших об'єктів у сцені, наприклад: передбачити, що автомобіль збирається повернути, або що пішохід вийде на дорогу. Це дозволяє створювати більш гнучкі моделі поведінки, які не базуються лише на заздалегідь заданих траєкторіях (рис.3.8).

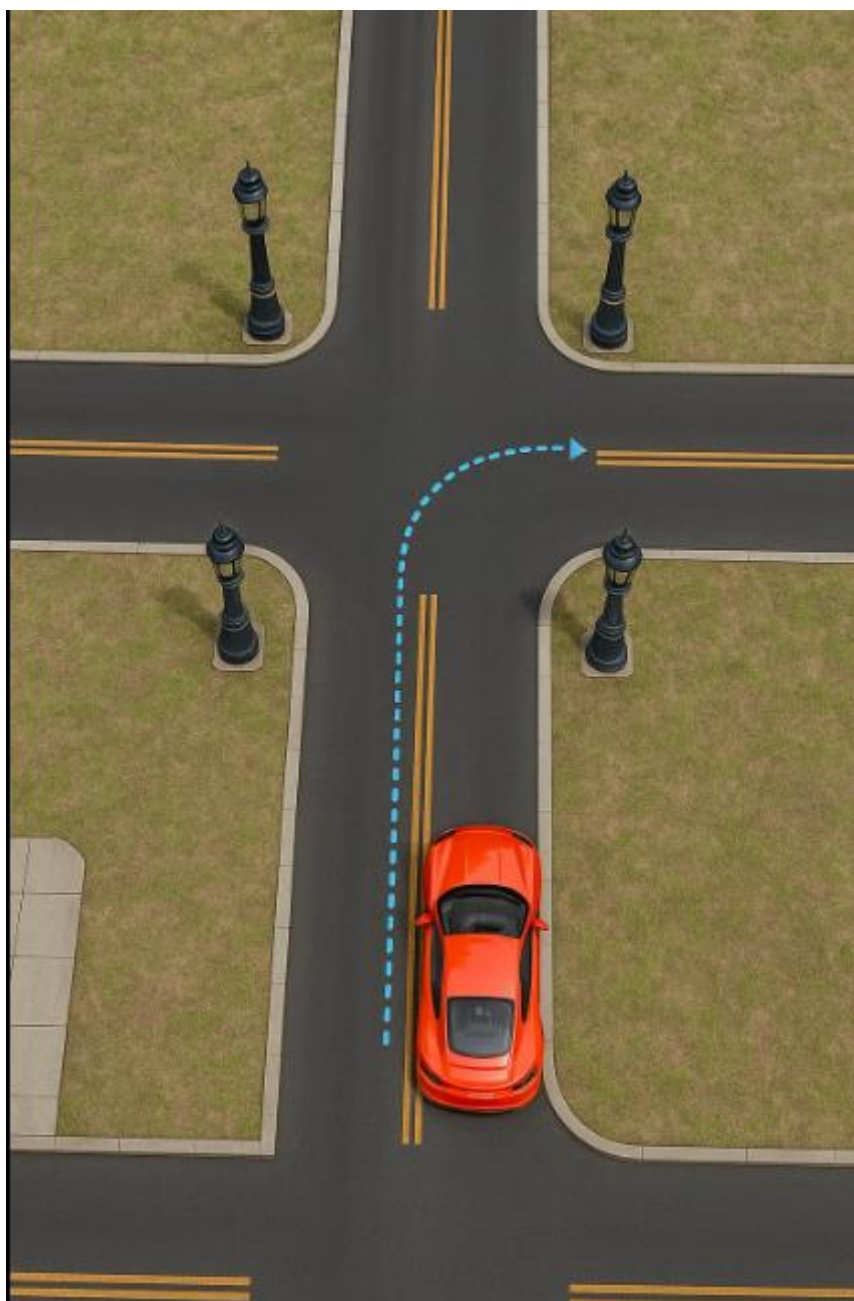


Рис. 3.8 Схематичне зображення прогнозу поведінки об'єктів у симуляції

Можна реалізувати розумні світлофори, які враховують потік транспорту, кількість пішоходів, час доби тощо. Такі світлофори динамічно змінюють тривалість сигналів, мінімізуючи затори. Подібна логіка широко застосовується в системах «розумного міста» (рис.3.9).



Рис. 3.9 Приклад адаптивного регулювання на перехресті

ІІІ може працювати не тільки в режимі реагування, а й у режимі навчання. Наприклад, симуляція може фіксувати, скільки разів траплялись помилки, який час очікування на світлофорі був найбільш ефективним, як поведуться нові транспортні агенти. Надалі ці дані можуть використовуватись для навчання моделей або виводу рекомендацій (рис.3.10).



Рис. 3.10 Приклад збору даних у симуляції для подальшого аналізу

Таким чином, застосування ІШІ у сценах на базі Omniverse відкриває шлях до створення автономних, самонавчальних віртуальних систем, здатних імітувати реальні ситуації, адаптуватися до умов та підвищувати ефективність симуляцій.

3.4 Масштабування проєкту до Smart City

Ідея створення цифрового моделювання перехрестя у середовищі NVIDIA Omniverse може стати відправною точкою для масштабнішого проєкту – віртуального міського простору, або ж цифрового двійника «розумного міста». Таке розширення дозволяє моделювати більш складні транспортні ситуації, взаємодію

інфраструктури, прогнозування заторів та оптимізацію руху на рівні цілого району або міста.

При переході до моделювання реального міського простору, з'являється можливість імпортувати карти або координати перехресть з відкритих джерел, таких як OpenStreetMap, Google Maps API тощо (рис.3.11). Це дозволяє:

- будувати сцени, що відповідають справжній топографії міста;
- тестувати рішення до їх реального впровадження;
- виконувати урбаністичне планування в цифровому середовищі.

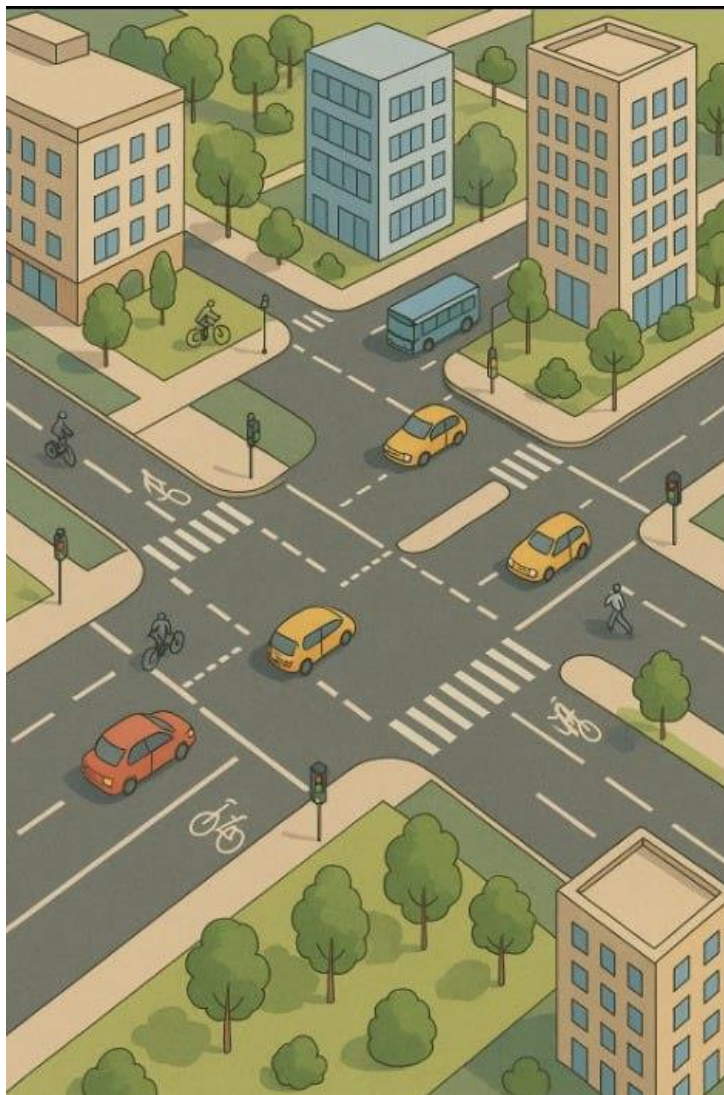


Рис. 3.11 Віртуальна карта перехресть у проєкті Smart City

На рівні міського цифрового двійника доцільно додати десятки або сотні агентів (рис.3.12):

- автомобілі, вантажівки, автобуси з різними маршрутами;

- пішоходи з власними траєкторіями;
- велосипеди та мікротранспорт;
- екстрені служби з пріоритетним рухом.



Рис. 3.12 Масштабована симуляція з багатьма агентами

У масштабній симуляції можливо реалізувати зв'язки між перехрестями — наприклад, світлофор на одному перехресті змінює стан залежно від завантаження наступної ділянки. Це дозволяє створювати глобальні адаптивні транспортні мережі з прогнозуванням заторів і перенаправленням потоку (рис.3.13).

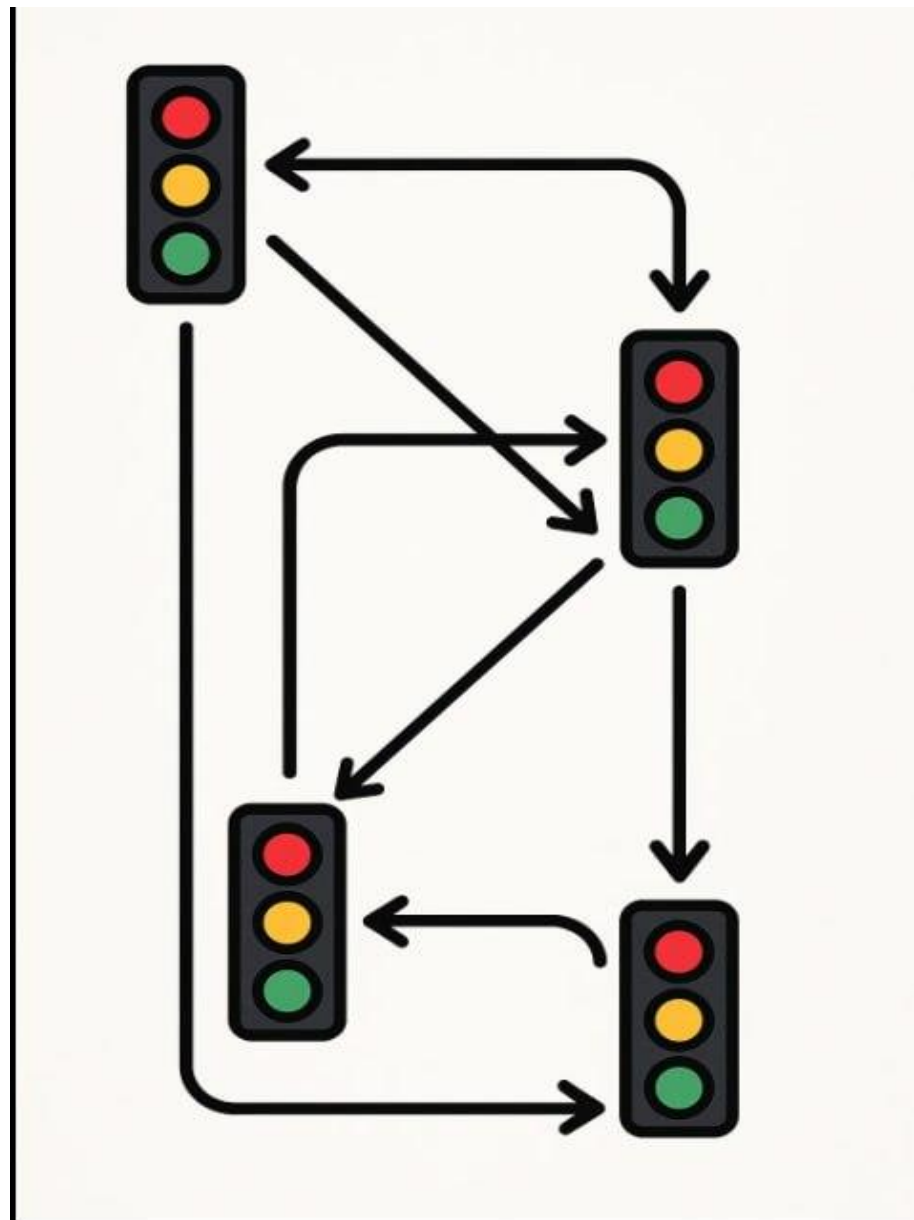


Рис. 3.13 Взаємодія транспортних вузлів у рамках Smart City

Застосування для стратегічного планування. Такий цифровий двійник може використовуватись міською владою для:

- тестування нових маршрутів;
- виявлення вузьких місць у трафіку;
- прийняття рішень щодо розміщення нових об'єктів інфраструктури.

Таким чином, простий освітній проєкт із симуляцією світлофора може масштабуватись до інструмента управління міською транспортною системою.

3.5. Інтеграція з зовнішніми системами

Однією з ключових переваг платформи NVIDIA Omniverse є її відкритість до інтеграції з іншими програмними системами, хмарними сховищами, базами даних та мережевими сервісами. Це дозволяє вивести створену сцену за межі локального середовища і використовувати її в рамках більших інформаційних та інженерних платформ.

Спільна розробка сцени через Omniverse Nucleus. Omniverse Nucleus слугує центральним сховищем даних USD, яке підтримує одночасну роботу кількох користувачів над одним проєктом (рис.3.14). Це дає можливість:

- колективно редагувати сцену (наприклад, один користувач додає транспорт, інший — логіку руху);
- миттєво зберігати зміни без втрати даних;
- створювати версії (version control) для порівняння змін.



Рис. 3.14 – Спільна робота у хмарному репозиторії Omniverse Nucleus

Публікація симуляцій у хмарі (Omniverse Cloud). Omniverse Cloud дозволяє переносити сцену в інтернет-середовище, де її можуть переглядати або взаємодіяти

з нею користувачі з будь-якого пристрою без встановлення важкого ПЗ (рис.3.15) .

Це особливо зручно для:

- демонстрацій на конференціях;
- дистанційного навчання;
- презентацій клієнтам чи замовникам.

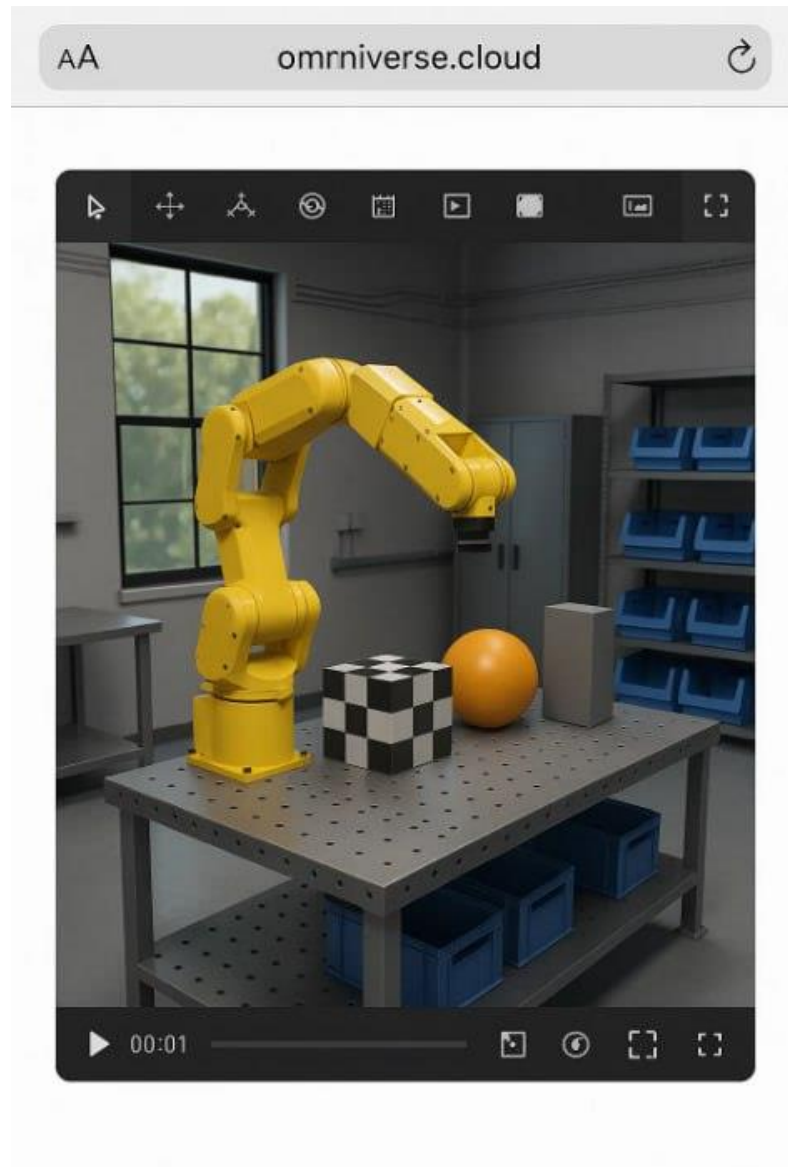


Рис. 3.15 – Веб-інтерфейс симуляції сцени у хмарному середовищі

Обмін даними з Python, API, датчиками (рис.3.16). Завдяки відкритості до Python та REST API, сцена може зв'язуватись із:

- реальними датчиками (наприклад, через MQTT);
- сторонніми сервісами аналітики (наприклад, Google Sheets, Grafana);

- програмами на базі ШІ (наприклад, модуль прогнозування заторів на TensorFlow).

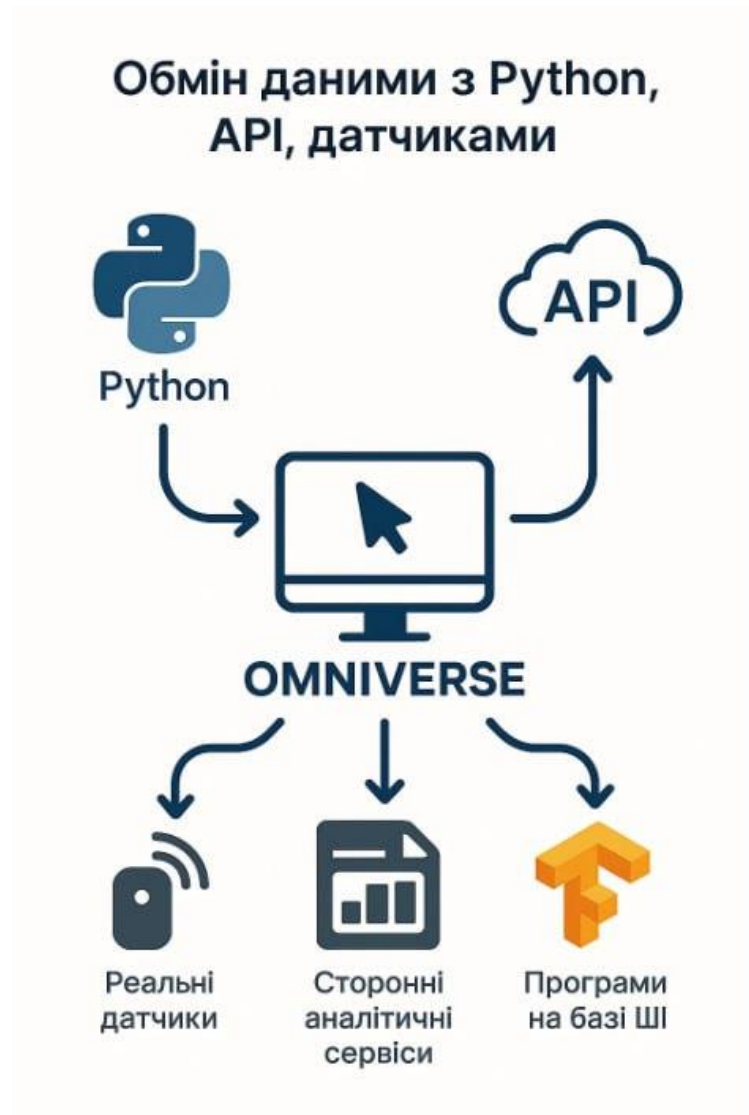


Рис. 3.16 – Схема зв'язку Omniverse з зовнішніми системами через API

Потенціал для VR/AR інтеграції . Оскільки сцени в Omniverse будуються на основі Universal Scene Description (USD), вони можуть бути легко перенесені в інші платформи — Unreal Engine, Unity, WebXR. Це відкриває можливості для:

- доповненої реальності з використанням планшетів/смартфонів;
- повного занурення у VR (наприклад, під час тренувань);
- змішаної реальності для міського планування.

Усе це підтверджує, що сцени, створені в Omniverse, можуть бути не просто навчальними моделями, а інтегрованими частинами більшої інформаційної екосистеми, яка охоплює як цифрове, так і фізичне середовище.

3.6. Прогноз подальшого розвитку платформи Omniverse

Платформа NVIDIA Omniverse розвивається стрімкими темпами, і її потенціал виходить далеко за межі 3D-моделювання. Завдяки своїй гнучкій архітектурі, модульності та орієнтації на відкриті стандарти (зокрема USD), Omniverse перетворюється на інструмент, що стає центральним елементом індустрії цифрових двійників, метавсесвіту та індустрії 5.0.

Поглиблення інтеграції зі штучним інтелектом. Omniverse вже сьогодні тісно взаємодіє з модулями NVIDIA AI. Очікується:

- подальший розвиток бібліотек для розпізнавання об'єктів у реальному часі;
- навчання агентів на базі поведінки в симульованих середовищах;
- інтеграція з моделями генеративного ШІ (наприклад, GPT, Diffusion AI).

Це дозволить створювати абсолютно автономні системи з адаптивною поведінкою в реальному та цифровому світі.

Розвиток Omniverse Cloud та віртуальних колаборацій. Omniverse Cloud стане центральною платформою для спільної роботи дизайнерів, інженерів, архітекторів і дослідників з усього світу (рис.3.17). Очікується:

- покращення веб-інтерфейсів перегляду сцен без встановлення програм;
- підтримка спільного моделювання в реальному часі;
- інтеграція з популярними корпоративними інструментами (Slack, GitHub, Jira).



Рис.3.17 Перспектива спільної хмарної роботи у віртуальному середовищі

Підтримка розширених реальностей (XR,MR,VR). Завдяки USD як стандарту обміну 3D-сценами, Omniverse отримує нативну підтримку:

- для VR-платформ (HTC Vive, Oculus, Varjo);
- для AR (Apple Vision Pro, WebXR);
- для MR-рішень на базі Unity або Unreal Engine.

Це означає, що сцена, змодельована в Omniverse, може бути адаптована як для шолома віртуальної реальності, так і для доповненої реальності на планшеті.

Автоматизація та Low-code рішення. Ще один вектор розвитку — створення інтерфейсів для користувачів без технічної підготовки (рис.3.18):

- вузлові редактори логіки (Omnigraph);
- конструктори симуляцій за шаблоном;
- сценарії для викладання в школах та університетах.

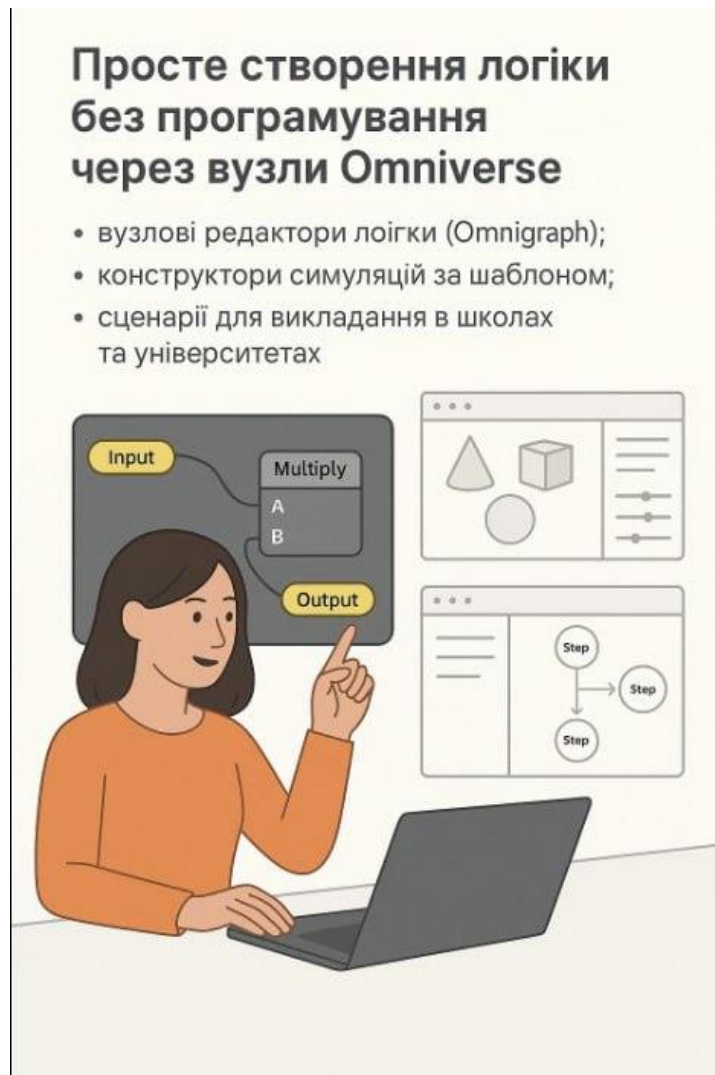


Рис. 3.18 Просте створення логіки без програмування через вузли Omniverse

Вихід за межі віртуального світу. Omniverse стає ядром екосистеми цифрових двійників. У найближчому майбутньому очікується:

- тісна взаємодія з IoT-пристроями (сенсори, камери, дрони);
- симуляція заводів, міст, аеропортів у реальному часі;
- впровадження у системи управління енергією, безпекою, логістикою.

Це означає, що навіть прості сцени, створені сьогодні студентами, можуть стати базою для реального виробничого або міського середовища в майбутньому.

3.7. Узагальнення конструктивних пропозицій

У цьому підпункті підсумовано основні напрями вдосконалення цифрової сцени, її розширення, інтеграції та подальшого застосування. Пропозиції згруповані за ключовими напрямками та представлені у вигляді таблиці для зручного сприйняття (табл.3.1).

Таблиця 3.1

Проблеми та відповідні конструктивні рішення

Проблема або обмеження	Запропоноване рішення
Обмежена деталізація сцени	Додавання пішоходів, знаків, текстур, об'єктів благоустрою
Статична логіка	Реалізація анімацій, станів, реакцій на події
Обмежене візуальне сприйняття	Покращення освітлення, рендерингу, текстуровання
Відсутність адаптації до зміни умов	Використання ІІІ для адаптивної логіки та аналізу ситуацій
Локальна сцена, без зв'язку із системами	Інтеграція з API, IoT, хмарними сервісами
Вузький сценарій використання	Розширення на освіту, Smart City, AR/VR, інженерію

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було реалізовано повноцінну цифрову сцену дорожнього перехрестя з дотриманням правил дорожнього руху. Застосування програмних засобів моделювання, таких як Blender та NVIDIA Omniverse, дозволило створити реалістичне середовище, в якому об'єкти взаємодіють відповідно до заданої логіки. Особливу увагу було приділено реалізації логіки світлофорів та поведінки транспортних засобів у симульованих умовах.

На першому етапі була проведена ґрунтовна теоретико-методична підготовка: розглянуто сучасні підходи до цифрового моделювання, порівняно популярні програмні середовища, такі як Unity, Blender, Autodesk 3ds Max та інші. Найбільш перспективним для завдання цієї роботи виявився саме Omniverse, зокрема його модуль Isaac Sim, що забезпечує високий рівень фізичної точності та підтримку Python API.

У практичній частині сцена була побудована поетапно: спочатку змодельовані об'єкти в Blender, потім імпортовані в Omniverse та згруповані в ієрархії. Після цього було реалізовано симуляцію логіки світлофорів, що керує рухом автомобіля відповідно до кольору сигналу. Для цього застосовувався скриптовий підхід на Python, що дозволив створити гнучке керування поведінкою об'єктів.

Оскільки апаратні ресурси були обмежені, частина результатів симуляції була згенерована у вигляді візуалізацій, що імітують поведінку об'єктів у реальному часі. Це дало змогу повноцінно представити функціонування сцени, зберігаючи достовірність роботи.

У конструктивному розділі було висунуто низку пропозицій для подальшого розвитку проекту. Серед них — додавання пішоходів, розширення сцени до рівня Smart City, впровадження модулів штучного інтелекту для аналізу ситуацій, а також інтеграція з зовнішніми API та хмарними сервісами. Усі ці напрямки

доводять, що навіть на базі навчального проєкту можна створити фундамент для сучасного цифрового двійника транспортного середовища.

Загалом, виконання цієї роботи дозволило не лише закріпити знання в галузі 3D-моделювання та програмування, а й отримати цінний досвід роботи з новітніми інструментами симуляції та візуалізації, які все частіше застосовуються в інженерії, освіті та міському плануванні.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

NVIDIA Omniverse Documentation [Електронний ресурс]. – Режим доступу до ресурсу: <https://docs.nvidia.com/omniverse>

Isaac Sim Application Template [Електронний ресурс]. – Режим доступу до ресурсу: <https://github.com/isaac-sim/isaacsim-app-template>

Nucleus Server Documentation [Електронний ресурс]. – Режим доступу до ресурсу: <https://docs.omniverse.nvidia.com/nucleus>

NVIDIA Omniverse GitHub [Електронний ресурс]. – Режим доступу до ресурсу: <https://github.com/NVIDIA-Omniverse>

Blender Foundation. Blender – Open Source 3D Creation Software [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.blender.org/>

Unity Technologies. Unity Manual and Documentation [Електронний ресурс]. – Режим доступу до ресурсу: <https://docs.unity3d.com/>

Khronos Group. Universal Scene Description (USD) [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.khronos.org/usd/>

Autodesk. What is 3D Modeling? [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.autodesk.com/solutions/3d-modeling>

OpenUSD – Pixar USD Overview [Електронний ресурс]. – Режим доступу до ресурсу: <https://openusd.org>

NVIDIA RTX Renderer Overview [Електронний ресурс]. – Режим доступу до ресурсу: <https://developer.nvidia.com/rtx>

NVIDIA PhysX SDK Documentation [Электронный ресурс]. – Режим доступа до ресурсу: <https://developer.nvidia.com/physx-sdk>

NVIDIA Omniverse Isaac Sim User Guide [Электронный ресурс]. – Режим доступа до ресурсу: <https://docs.omniverse.nvidia.com/isaacsim/latest>

NVIDIA Omnigraph Node-Based Logic [Электронный ресурс]. – Режим доступа до ресурсу: <https://docs.omniverse.nvidia.com/kit/docs/omni.graph/latest>

GitHub – Omniverse Samples [Электронный ресурс]. – Режим доступа до ресурсу: <https://github.com/NVIDIA-Omniverse/OmniSamples>

Digital Twin Consortium. Definition & Use Cases [Электронный ресурс]. – Режим доступа до ресурсу: <https://www.digitaltwinconsortium.org>

Grieves M., Vickers J. Digital Twin: Mitigating Emergent Behavior in Complex Systems. Springer, 2017 [Электронный ресурс]. – Режим доступа до ресурсу: https://link.springer.com/chapter/10.1007/978-3-319-38756-7_4

Microsoft. Introduction to Digital Twins [Электронный ресурс]. – Режим доступа до ресурсу: <https://learn.microsoft.com/en-us/azure/digital-twins>

Siemens Digital Twin Explained [Электронный ресурс]. – Режим доступа до ресурсу: <https://new.siemens.com/global/en/products/software/digital-twin.html>

OpenStreetMap [Электронный ресурс]. – Режим доступа до ресурсу: <https://www.openstreetmap.org>

Unreal Engine Docs – USD Import [Электронный ресурс]. – Режим доступа до ресурсу: <https://docs.unrealengine.com/en-US/WorkingWithContent/USD>

Python.org – Official Python Documentation [Електронний ресурс]. – Режим доступу до ресурсу: <https://docs.python.org>

NVIDIA AI Enterprise [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.nvidia.com/en-us/ai-data-science/ai-enterprise/>

NVIDIA Isaac ROS [Електронний ресурс]. – Режим доступу до ресурсу: <https://developer.nvidia.com/isaac/ros>

Omniverse Extensions & Scripting [Електронний ресурс]. – Режим доступу до ресурсу: <https://docs.omniverse.nvidia.com/kit/docs/omni.kit.scripting/latest>

Реутов А.І., Пилипенко І.М. Технології віртуальної і доповненої реальності. – Київ: НАУ, 2021. – 204 с.

Буряк Є.І., Гнатюк С.М. Інформаційні технології в комп'ютерному моделюванні: навч. посіб. – Київ: КНУ, 2020. – 188 с.

IBM Smart Cities Overview [Електронний ресурс]. – Режим доступу до ресурсу: https://www.ibm.com/smarterplanet/us/en/smarter_cities/overview

Cisco Smart+Connected Digital Platform [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.cisco.com/c/en/us/solutions/industries/smart-connected-communities.html>

NVIDIA Metaverse Technology [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.nvidia.com/en-us/omniverse/metaverse/>

NVIDIA AI Playground – Use Cases [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.nvidia.com/en-us/research/ai-playground/>

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (презентація)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

НА ТЕМУ: «ЦИФРОВЕ МОДЕЛЮВАННЯ АРХІТЕКТУРНОГО СЕРЕДОВИЩА В NVIDIA OMNIVERSE»

НА ЗДОБУТТЯ ОСВІТЬОГО СТУПЕНЯ МАГІСТРА
ЗІ СПЕЦІАЛЬНОСТІ 126 ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ
ОСВІТНЬО-ПРОФЕСІЙНОЇ ПРОГРАМИ ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

Виконав: здобувач вищої освіти гр. ІСД-41

Шевчук Андрій Сергійович

Керівник: доцент кафедри

Сеньков Олег Вікторович

МЕТА ТА ЗАВДАННЯ ПРОЄКТУ

- ◆ Мета роботи:
 - Створення інтерактивної 3D-сцени перехрестя з симуляцією правил дорожнього руху на базі NVIDIA Omniverse з використанням Blender та Isaac Sim.
- ◆ Основні завдання:
 - Аналіз можливостей сучасних 3D-платформ та вибір оптимального середовища.
 - Моделювання елементів сцени: дорога, світлофори, транспортні засоби.
 - Побудова логіки руху об'єктів на Python відповідно до сигналів світлофорів.
 - Імпорт моделі з Blender у Isaac Sim через USD.
 - Проведення симуляції та візуалізація сцени з використанням RTX-рендерингу.
 - Формування конструктивних пропозицій щодо масштабування сцени.

АКТУАЛЬНІСТЬ ТЕМИ

Універсальна платформа для створення цифрових світів

Nvidia Omniverse — це потужна платформа для створення та симуляції складних 3D-сцен в режимі реального часу.

Вона об'єднує роботу графічних дизайнерів, інженерів, архітекторів, фізиків і програмістів в одному віртуальному середовищі.

Підтримує реалістичну фізику, освітлення, анімацію та симуляцію поведінки об'єктів, що робить її ідеальною для професійних застосувань.

Реалістичні симуляції для промисловості, архітектури, робототехніки

Використовується в: **Автомобільній промисловості** (BMW, Mercedes) , **Будівництві** (AEC-галузь) , **Робототехніці та логістиці** (через Isaac Sim) , **Індустрії розваг та візуальних ефектів**

Компанії можуть моделювати складні процеси без фізичних прототипів, економлячи час і кошти.

Інноваційний підхід до рендерингу — RTX Renderer

Використовує **RTX-візуалізацію в реальному часі** — це надточне, реалістичне зображення сцен з урахуванням всіх оптичних ефектів.

Ідеально підходить для презентацій, демонстрацій та перегляду сцен без попереднього рендерингу.

Потужна підтримка штучного інтелекту та машинного навчання

Завдяки інтеграції з Nvidia AI/RTX, Omniverse може: Генерувати **реалістичні анімації та поведінку** , Аналізувати сцени ,Автоматично покращувати матеріали, текстури, звуки.

Omniverse дозволяє **кільком користувачам працювати над однією сценою одночасно.**

Це революція в спільній роботі — архітектор, дизайнер і програміст можуть одночасно бачити зміни один одного.

Об'єкт дослідження:	<u>Цифрове моделювання архітектурного середовища в інтерактивних 3D-редакторах.</u>
Предмет дослідження:	Функціональні можливості платформи Nvidia Omniverse для моделювання, симуляції та візуалізації архітектурних сцен.
Мета дослідження:	Дослідити процес створення та імплементації віртуального середовища з використанням Omniverse, а також його інтеграцію з іншими інструментами (Blender, Isaac Sim) для моделювання дорожнього руху.
Наукова новизна та практична значущість:	У роботі розглянуто практичні можливості застосування Omniverse як універсального середовища для створення цифрових двійників архітектурного середовища з підтримкою реалістичних симуляцій та візуалізацій. Результати дослідження можуть бути використані у сферах розумного містобудування, візуального аналізу інфраструктури та розробки цифрових середовищ для тестування міських рішень.

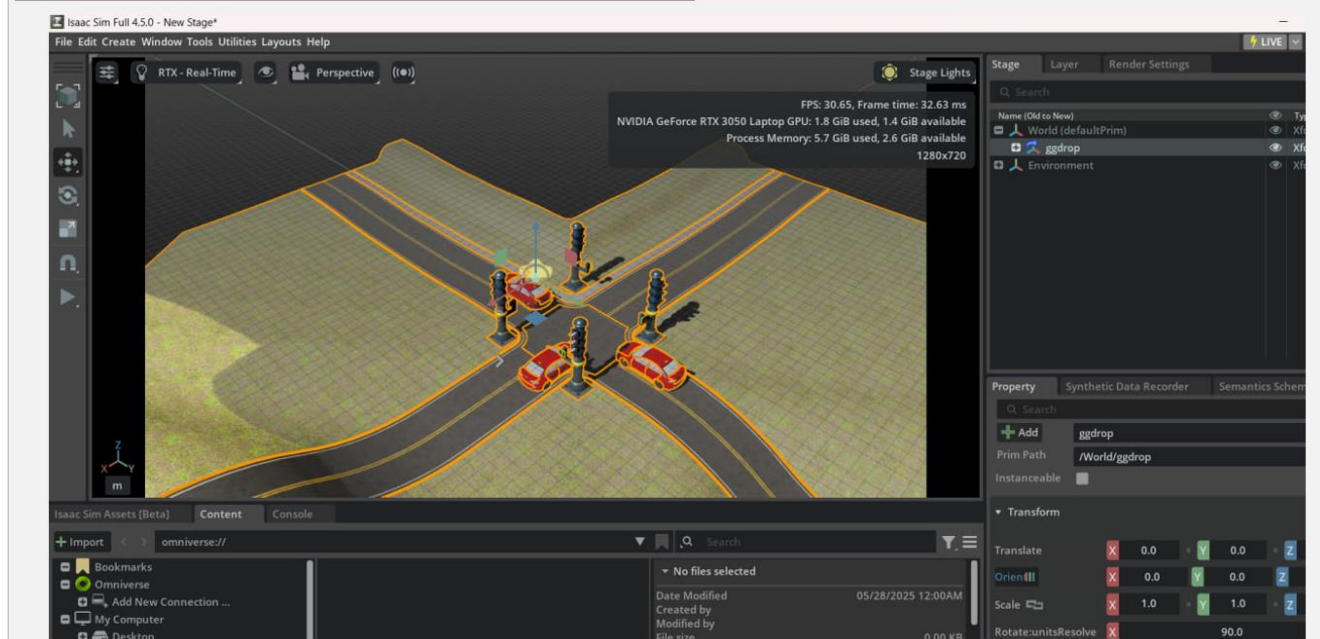
ТЕОРЕТИЧНІ ОСНОВИ ПОБУДОВИ ЦИФРОВОГО СЕРЕДОВИЩА В OMNIVERSE

Omniverse побудовано на моделі інтеграційного візуального середовища, яке працює в реальному часі, використовуючи формат **USD (Universal Scene Description)**.

Основна ідея — створення єдиної сцени, до якої підключаються різні сервіси, інструменти та користувачі.

- ♦ **Користувач** створює або редагує 3D-моделі у сторонніх редакторах (наприклад, **Blender**), після чого експортує сцену у форматі USD.
- ♦ **Сцена синхронізується з Omniverse** через Omniverse Nucleus — сервер обміну даними.
- ♦ У **Isaac Sim** відбувається **фізична симуляція об'єктів**, включаючи автомобілі, світлофори, фізику зіткнень, правила дорожнього руху
- ♦ **Omniverse RTX Renderer** відповідає за **візуалізацію сцени в реальному часі** з урахуванням освітлення, тіней та матеріалів.
- ♦ Дані можуть **зберігатися локально або в хмарі**, а користувачі можуть отримувати доступ до середовища через різні пристрої (ПК, ноутбуки, віртуальні машини).

РЕЗУЛЬТАТ БУДУВАННЯ СЦЕНИ



СТРУКТУРА ТА ЕТАПИ
РЕАЛІЗАЦІЇ ПРОЄКТУ

- ◆ **1. Вивчення платформи**
Опрацювання документації [NVIDIA Omniverse](#) та [Isaac Sim](#) (офіційні ресурси, [GitHub](#)).
- ◆ **2. Створення сцени в Blender**
Моделювання перехрестя, машин та світлофорів у [Blender](#).
- ◆ **3. Експорт у формат USD**
Перетворення сцени у формат [Universal Scene Description](#) для сумісності з [Omniverse](#).
- ◆ **4. Імпорт у Isaac Sim**
Завантаження USD-файлу в [Isaac Sim](#) та перевірка коректності сцени.
- ◆ **5. Програмування логіки світлофорів**
Реалізація керування сигналами за допомогою [Python API](#) в [Isaac Sim](#).
- ◆ **6. Налаштування руху транспорту**
Налаштування руху моделей автовок згідно сигналів світлофорів.
- ◆ **7. Рендеринг та перевірка**
Тестування сцени, візуалізація за допомогою RTX, перевірка логіки та фізики.



ВИКОРИСТАНІ ІНСТРУМЕНТИ ТА
ТЕХНОЛОГІЇ

Інструмент	Призначення
Blender	Створення 3D-моделей
USD	Формат обміну сценами
Isaac Sim	Симуляція фізики та руху
Omniverse Nucleus	Сервер синхронізації сцени
RTX Renderer	Реалістичне візуалізування
Python	Програмування поведінки об'єктів

ПЕРСПЕКТИВИ ВИКОРИСТАННЯ ТА РОЗВИТКУ

Масштабування сцени до рівня цілого міста

Omniverse дозволяє моделювати великомасштабні цифрові простори — від окремих перехресть до повноцінної інфраструктури міста з вулицями, транспортом, пішоходами та сигналами.

Інтеграція з концепцією “розумного міста”

Сцена може слугувати віртуальним аналогом реальної інфраструктури — для тестування, аналізу ефективності рішень у транспортній логістиці, керуванні трафіком і екологічному моніторингу.

Реалізація віртуальних тренажерів для навчання ПДР

Omniverse дає змогу створити реалістичні симулятори для водіїв та пішоходів — з варіативними сценаріями дорожніх ситуацій, погодними умовами, рівнями складності тощо.

Потенціал для освітніх і наукових проєктів

Модель може бути використана в університетах, лабораторіях та дослідницьких центрах для вивчення міської мобільності, урбаністики, машинного навчання, комп'ютерного зору тощо.

Використання штучного інтелекту для автономного керування транспортом

Інтеграція з AI-алгоритмами дає змогу тренувати автономні автомобілі в безпечному симульованому середовищі — з урахуванням правил дорожнього руху, поведінки інших учасників і складних умов.

ВИСНОВКИ :

У роботі проведено детальне вивчення можливостей платформи **NVIDIA Omniverse** та симулятора **Isaac Sim** для створення цифрових сцен з елементами міської інфраструктури. Це дозволило зробити висновок, що використання Omniverse повністю відповідає задачам моделювання складних архітектурних середовищ з підтримкою фізики та логіки взаємодії об'єктів.

Було досліджено процес створення та імпорту 3D-моделей, а також принципи роботи з форматом **USD**, що забезпечує ефективний обмін сценами між середовищами **Blender** та **Omniverse**. Реалізовано взаємодію між об'єктами сцени, зокрема – **керування рухом транспорту відповідно до роботи світлофорів**.

Проєкт включає розробку логіки поведінки об'єктів за допомогою **Python API**, що дозволило реалізувати симуляцію в реальному часі із врахуванням правил дорожнього руху.

Також було протестовано роботу сцени в **Isaac Sim**, досліджено особливості **рендерингу** з використанням **RTX**, а також отримано практичні навички з побудови віртуального середовища, яке потенційно може бути масштабоване до рівня цифрового міста.

Запропоноване рішення може використовуватись як основа для подальших досліджень у сфері **розумних міст, автономного транспорту та освітніх симуляторів**.

ДЯКУЮ ЗА УВАГУ !