

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти перший (бакалаврський)

Створення AR-моделі системи навчання з основних
правил дорожнього руху

(тема)

Виконав:

здобувач 4 року навчання,

групи КІУКІ-21-4

Анастасія КОВАЛЕНКО

(власне ім'я, прізвище)

Спеціальність

123 «Комп'ютерна інженерія»

(код і повна назва спеціальності)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма

Комп'ютерна інженерія

(повна назва освітньої програми)

Керівник: доц. Наталія БОЛОГОВА

(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ЕОМ

(підпис)

Андрій КОВАЛЕНКО

(власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління _____

Кафедра _____ електронних обчислювальних машин _____

Рівень вищої освіти _____ перший (бакалаврський) _____

Спеціальність _____ 123 «Комп'ютерна інженерія» _____
(код і повна назва)

Тип програми _____ освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Комп'ютерна інженерія _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Коваленко Анастасії Володимирівні _____
(прізвище, ім'я, по батькові)

1. Тема роботи Створення AR-моделі системи навчання з основних правил дорожнього руху

затверджена наказом по університету від “ 26 ” травня 2025 р. № 424 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 17 червня 2025 р.

3. Вхідні дані до роботи 1) платформи Qualcomm Vuforia; 2) програмний модуль Unity;
3) мова програмування: Java; 4) середовище розробки: Androidю

4. Перелік питань, що потрібно опрацювати у роботі _____

1) аналіз проблеми та огляд існуючих рішень;

2) аналіз існуючих технологій AR моделей для ПДР;

3) аналіз вимог до створення програмного модулю;

4) розробка рішення реалізації модулю;

5) програмна реалізація застосунку;

6) висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій 13 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Аналіз проблеми та огляд існуючих рішень	27.05.25-30.05.25	
2	Аналіз існуючих технологій AR моделей	31.05.25-03.06.25	
3	Розробка рішення реалізації модулю	04.06.25-07.06.25	
4	Програмна реалізація програмного модулю	08.06.25-09.06.25	
5	Оформлення матеріалів кваліфікаційної роботи	10.06.25-11.06.25	
6	Подання кваліфікаційної роботи керівникові та її попередній захист	12.06.25-13.06.25	
7	Подання кваліфікаційної роботи на рецензування	14.06.25-16.06.25	

Дата видачі завдання “ 26 ” травня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____
(підпис)

доц. Наталія БОЛОГОВА
(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 58 с., 26 рис., 2 табл., 1 дод., 14 джерел

ДОПОВНЕНА РЕАЛЬНІСТЬ, AR-ТЕХНОЛОГІЇ, МОБІЛЬНИЙ ЗАСТОСУНОК, UNITY, VUFORIA, КОМП'ЮТЕРНИЙ ЗІР, ТРЕКІНГ МАРКЕРІВ, 3D-МОДЕЛІ, ПРАВИЛА ДОРОЖНЬОГО РУХУ, ІНТЕРАКТИВНЕ НАВЧАННЯ, ОСВІТНІ ТЕХНОЛОГІЇ.

Метою кваліфікаційної роботи є вивчення та аналіз основних алгоритмів та методів для розробки застосунку доповненої реальності.

У ході виконання кваліфікаційної роботи вивчено та проаналізовано основні алгоритми та методи для розробки застосунку доповненої реальності.

У роботі бібліотека Vuforia використовується для точного розпізнавання маркерів і розміщення 3D-моделей у просторі доповненої реальності, що забезпечує стабільне трекінг-відображення в мобільному застосунку. Архітектура MVP дозволяє ефективно розділити інтерфейс, бізнес-логіку та обробку даних, що спрощує підтримку й тестування програми. Для реалізації зацикленого відтворення анімацій застосовується спеціальний клас телепортації, який автоматично повертає об'єкти до початкової позиції, забезпечуючи безперервність сценарію.

ABSTRACT

Bachelor's thesis: 58 pages, 26 figures, 2 tables, 1 appendices, 14 sources.

ARTIFICIAL INTELLIGENCE, NEURAL NETWORKS, MACHINE LEARNING, PYTHON, ANACONDA.

The major goal of this thesis is to study and analyze the core algorithms and methods for the development of an augmented reality application. In order to achieve this goal, the research involved an in-depth examination of key approaches and technologies used in AR development.

In order to the qualification work, the Vuforia library is utilized for accurate marker recognition and 3D model placement within the augmented reality space, enabling stable tracking and rendering on mobile devices. The use of the MVP architecture effectively separates the user interface, business logic, and data processing layers, which simplifies application maintenance and testing. To enable continuous animation playback, a dedicated teleportation class is implemented, allowing objects to automatically return to their initial positions, thereby ensuring seamless scenario repetition.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	7
ВСТУП	8
1 АНАЛІЗ АКТУАЛЬНОСТІ ТЕХНОЛОГІЇ ДОПОВНЕНОЇ РЕАЛЬНОСТІ ТА ПОСТАНОВКА ЗАДАЧІ	10
1.1 Огляд історії розвитку графічних інтерфейсів та технологій доповненої та віртуальної реальності	10
1.2 Аналіз актуальності використання технології доповненої реальності.....	12
1.2.1 Аналіз сучасного стану використання доповненої реальності у навчанні дітей правилам дорожнього руху в Україні	14
1.3 Вибір інструментарію для реалізації програмного модуля	16
2 ОПИС МОДЕЛІ ДОПОВНЕНОЇ РЕАЛЬНОСТІ ТА РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ	21
2.1 Методика комп'ютерного зору у доповненій реальності.....	21
3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОЄКТУ	26
3.1 Реалізація програмного модуля	26
4 ТЕСТУВАННЯ ТА КОРИГУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО МОДУЛЯ	42
4.1 Проведення обчислювальних експериментів.....	42
4.2 Оптимізація розробленої інформаційної технології	44
ВИСНОВКИ.....	49
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	50
ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	52

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

НУШ – нова українська школа

МЗ – мобільний застосунок

ОС – операційна система

ПЗ – програмне забезпечення

AR – технологія доповненої реальності (англ., augmented reality)

ВСТУП

Доповнена реальність є однією з найперспективніших технологій, що швидко розвиваються, у сфері інформаційних технологій. Область її застосування дуже широка: починаючи з ігор та розважальних додатків і закінчуючи додатками для підготовки та проведення хірургічних операцій. Головна перевага програми з технологією доповненої реальності – наочність, що сприяє кращому засвоєнню інформації. Завдяки цьому технологія набула широкого поширення у сфері освіти.

Протягом останніх двадцяти років школи та інші освітні заклади по всьому світу спішно застосовують інформаційні технології для спрощення процесу навчання, а також збільшення ступеня залучення учнів та студентів до цього процесу.

У цих умовах було вирішено поставити завдання створення моделі доповненої реальності, і навіть розробити її основі мобільний застосунок з допомогою технології доповненої реальності на навчання основам правил дорожнього руху.

Актуальність AR-моделювання обумовлена тим, що воно відкриває нові можливості для вивчення теорії та вдосконалення практичних навичок. Практичне поєднання віртуального та реального досвіду збагачує особистісно-орієнтовану діяльність учнів, дозволяє візуалізувати, переглядати та вивчати будь-які навчальні посібники, підвищуючи тим самим рівень якості освіти.

Мета роботи – створення моделі доповненої реальності та розробка на її основі програмного модуля для навчання дітей основ ПДР.

Об'єктом дослідження є вивчення та аналіз основних алгоритмів та методів для розробки застосунку доповненої реальності.

Досягнення поставленої мети у роботі вирішуються такі:

- аналіз існуючих реалізацій;

- опис моделі доповненою реальності і реалізація програмного модуля;

- тестування та коригування програмного модуля.

Кваліфікаційна робота включає вступ, три розділи, висновок і список використаних джерел.

1 АНАЛІЗ АКТУАЛЬНОСТІ ТЕХНОЛОГІЇ ДОПОВНЕНОЇ РЕАЛЬНОСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Огляд історії розвитку графічних інтерфейсів та технологій доповненої та віртуальної реальності

У розвитку інтерфейсу комп'ютерних систем виділяють три парадигми: пакетна обробка, інтерфейси командного рядка і графічний інтерфейс користувача. Пакетна обробка пов'язана з введенням певних даних у комп'ютер та очікуванням на результат. При цьому взаємодія не є інтерактивною. Так було з першими електронно-обчислювальними машинами, що з'явилися у середині ХХ століття. Програми та дані набивалися на перфокартах чи перфострічках та передавалися на обробку. Результат треба було чекати до кількох годин чи днів. Після приєднання до електронно-обчислювальних машин телетайп став можливий інтерфейс командного рядка. Інтерфейс командного рядка – це парадигма інтерфейсу, коли користувач взаємодіє з комп'ютером текстовими командами.

Починаючи з виходу графічних інтерфейсів на масовий ринок приблизно 30 років тому, взаємодія з комп'ютером не зазнала істотних змін. На цьому тлі перспективної виглядає концепція доповненої реальності. Поточні технології інтерфейсів користувача мають фундаментальні обмеження і сфокусовані в основному на взаємодії людини і комп'ютера. Технологія доповненої реальності пропонує вдосконалення інтерфейсу людини та реального навколишнього світу. Вона дозволяє природним чином впровадити комп'ютери в діяльність людини, не відволікаючи від суті завдань.

Дженніфер Уайт та Драган Ніколіч у своїй книзі «Virtual Reality and the Built Environment» згадують описують витоки технології VR: «Перші комерційні пакети віртуальної реальності стали доступні наприкінці 1980-х

років з заснуванням W Industries у Великій Британії та VPL Research у США. Виконавчому директору VPL Research Джарону Ланье приписують винахід терміна «віртуальна реальність» для опису DataGlove його компанії, що відстежується користувачем дисплея на голові, який занурює користувачів в інтерактивний» [1].

Віртуальна реальність — це термін, який використовується для 3D-середовищ, створених комп'ютером, які дозволяють користувачеві входити в синтетичні середовища та взаємодіяти з ними. Користувачі можуть по-різному «зануритися» в штучний світ комп'ютерів, який може бути або імітацією будь-якої форми реальності, або імітацією складного явища.

У телеприсутності (Telepresence, набір технологій, що дозволяє користувачеві, наприклад за допомогою спеціальних пристроїв, отримати враження того, що він знаходиться і/або впливає на місце, відмінне від його фізичного розташування.), основна мета полягає в тому, щоб розширити сенсомоторні можливості оператора і здатність вирішувати проблеми у віддаленому середовищі. У цьому сенсі телеприсутність можна визначити як систему людина/машина, в якій оператор-людина отримує достатню інформацію про телеоператора та середовище виконання завдань, що відображається досить природним чином, щоб оператор відчував фізичну присутність на віддаленому об'єкті. Дуже схоже на віртуальну реальність, у якій прагнемо створити ілюзію присутності у комп'ютерної симуляції, телеприсутність спрямовано створення ілюзії присутності у віддаленому місці.

«AR знаходиться у більш загальному контексті, званому змішаною реальністю (MR, Mixed reality), яка відноситься до багатовимірного спектру областей, що охоплюють віртуальну реальність (VR, Virtual reality), AR, телеприсутність та інші пов'язані технології. За допомогою передових технологій AR (наприклад, додавання комп'ютерного зору та розпізнавання об'єктів) інформація про навколишній реальний світ користувача стає інтерактивною та цифровою маніпульованою. Інформація про навколишнє

середовище та його об'єкти накладається на реальний світ» [2]. «Однією з основних завдань використання доповненої реальності є доповнення реального світу, розширюючи дійсність, що сприймається людиною» [3].

Тарек Ахрам у своїй книзі *Advances in Usability and User Experience* стверджує, що AR можна розглядати як технологію між Віртуальною реальністю і телеприсутністю. У той час як у віртуальній реальності середовище повністю синтетичне, а в телеприсутності воно цілком реальне, в AR користувач бачить реальний світ, доповнений віртуальними об'єктами. AR має унікальний потенціал впливу на користувачів, що робить його сполучною ланкою між реальною діяльністю та цифровими поліпшеннями» [3]. Хелен Папагіанніс розповідаючи про доповнену реальність у книзі «Доповнена реальність. Все, що ви хотіли дізнатися про технологію майбутнього» пише: «На технологію VR є свій специфічний попит, але AR дозволяє нам глибше занурюватися в реальний світ і бути з ним вже пов'язаними – світ, в якому ми проводимо більшу частину нашого часу» [4].

1.2 Аналіз актуальності використання технології доповненої реальності

Дослідження з тематики доповненої реальності проводяться з 80-90-х рр., але, якщо в той час потрібно відповідне обладнання (пристрої, що володіють великою обчислювальною потужністю для обробки інформації, що надходить на вхід, розрахунку тривимірної графіки з подальшим її виведенням на кольоровий дисплей), вимагала ін) і в цілому дана технологія вимагала громіздких рішень, і не була доступна масовому користувачеві, то сучасному світі ситуація дуже змінилася. Кінг Бретт у своїй книзі *Епоха доповненої реальності* стверджує: «Чим більше високотехнологічних новинок, тим вищий винахідницький потенціал – тим швидше з'являються ще більш просунуті технології. Винахід друкарського верстата призвело до різкого зростання кількості освічених людей і небувалого поширення знань»[5].

В даний час завдяки технологічному прогресу та розвитку споживчої техніки використання технології доповненої реальності стало доступно більшості людей. Зараз практично кожна людина має можливість роботи з доповненою реальністю за допомогою своїх смартфонів та персональних комп'ютерів. Сучасні смартфони оснащені камерою, потужними в порівнянні з ЕОМ минулих років процесорами, а також кольоровими дисплеями з великим розширенням.

Індустрія віртуальної та доповненої реальності пропонує інноваційні технології, які можуть бути використані у сфері розваг, освіти, професійної підготовки, медичних та промислових інновацій [6]. В освіті доповнена реальність поки що застосовується досить рідко, але все більше вчителів, дослідників та розробників починають рухатися у бік більш інтерактивних навчальних методик [7].

«Інтерактивність – ключова причина того, чому VR і AR такі привабливі для освітніх цілей» [2]. В основі комп'ютерного моделювання може бути математична модель, лабораторний експеримент або анімація, яка представляє роботу деякого механізму, перебіг процесу, будову складного предмета і т.д. У моделюючих програмах можливе широке використання інтерактивної графіки, яке дозволяє учню не тільки спостерігати за тим, що відбувається, але і впливати на параметри об'єктів, досліджуючи при цьому ефекти, які впливають на результат. Найчастіше учні віддають перевагу графіку та мультимедіа, а не тексту та паперовим носіям інформації. Замість читання розділу з підручника вони віддадуть перевагу відеоуроку [8]. «Інтерактивні навчальні методики дозволяють візуалізувати смислову складову освітнього процесу, спрямовану на мотиваційну складову учнів. Ця обставина особливо важлива для дітей молодшого шкільного віку, оскільки діти, які підходять до навчання усвідомлено, мають підвищену мотивацію та прагнення до навчання надалі» [9]. «Впровадження технології доповненої реальності дозволить підвищити якість навчання за рахунок мотивації учнів до самонавчання, підвищення інтересу аудиторії до матеріалу, що

викладається і вивчається» [6].

Саме додатки доповненої реальності дають можливість візуалізації процесів та предметів, які необхідно, але неможливо продемонструвати у навчальній аудиторії. Тому питання про використання технології доповненої реальності для підтримки освітнього процесу є одним із найактуальніших у сфері використання нових інформаційних технологій в освітньому процесі.

«Окрім переваг використання технологій доповненої реальності у різних соціально-економічних сферах існують і матеріальні переваги [7].

Після аналізу проблеми було зроблено висновок у тому, що застосування цієї технології у сфері навчання актуально. Однак у таких навчальних закладах, як школи, інформатизація освіти йде дуже повільним шляхом, а дана технологія представляється співробітникам і учням на рівні фантастики, тому застосунок, що створюється, повинен бути просто і зрозуміло у використанні.

1.2.1 Аналіз сучасного стану використання доповненої реальності у навчанні дітей правилам дорожнього руху в Україні

У сучасному освітньому середовищі України технології доповненої реальності (AR) поступово впроваджуються в навчальний процес, зокрема в початковій і середній освіті. AR дозволяє створити інтерактивне, наочне та емоційно насичене середовище, що особливо важливо при вивченні таких життєво важливих тем, як правила дорожнього руху (ПДР).

Поточний стан.

На сьогоднішній день в Україні AR-технології в освіті найчастіше використовуються у вигляді:

- мобільних додатків для смартфонів і планшетів;
- інтерактивних підручників із вбудованими QR-кодами;
- онлайн-платформ із використанням AR через веб-камеру;
- рішень у форматі STEM-лабораторій у межах проєктів Нової

української школи (НУШ).

Проте спеціалізованих AR-застосунків саме для навчання дітей правилам дорожнього руху наразі майже немає. Деякі винятки:

- «Безпечна дорога» – серія цифрових продуктів, створених за підтримки МВС та UNICEF, з елементами інтерактивності, але без повноцінної AR;

- окремі мультимедійні комплекси, які використовуються в автошколах для дорослих, не є адаптованими для дітей.

В таблиці 1.1 представлені приклади застосунків, які є на ринку.

Таблиця 1.1 – Існуючі приклади (на момент 2024–2025 років)

Назва проєкту / застосунку	Опис	Наявність AR	Орієнтація на дітей
SafeCity Kids	Навчальна гра для планшетів з віртуальними ситуаціями	Частково	Так
ПДР з поліцейським Хмарочосиком (YouTube, інтерактивні відео)	Візуальні пояснення для молодших школярів	Ні	Так
Wonderscope (іноземна)	Іноземна AR-платформа для навчання	Так	Так, але не локалізована

Таким чином, на українському ринку відсутній нішевий AR-додаток, який би інтерактивно навчав дітей ПДР у форматі доповненої реальності, що створює потенційно вільну нішу для розробки подібного продукту.

Проблеми та виклики:

- брак локалізованого AR-контенту українською мовою.
- недостатнє фінансування закладів освіти для впровадження інноваційних технологій.

- слабка інтеграція AR в освітні програми, відсутність методичних рекомендацій.

- низька цифрова грамотність у частини педагогів та відсутність ІТ-спеціалістів у школах.

Перспективи:

- з урахуванням державного курсу на цифровізацію освіти (цифрова трансформація), очікується зростання інтересу до технологій AR у початкових класах;

- AR може стати частиною національних програм безпеки дорожнього руху, що спрямовані на зменшення травматизму серед дітей;

- потенційно, такі продукти можуть отримати підтримку від МОН, МВС, UNICEF або місцевих громад, якщо буде доведено їх ефективність і безпечність.

Ринок освітніх AR-рішень в Україні наразі перебуває на початковому етапі розвитку, і в ніші навчання дітей правилам дорожнього руху практично відсутні продукти з використанням доповненої реальності. Це відкриває перспективи для створення нових освітніх рішень із використанням AR-технологій, що можуть суттєво підвищити ефективність навчання дітей правилам безпечної поведінки на дорозі.

1.3 Вибір інструментарію для реалізації програмного модуля

Було ухвалено рішення створити мобільний застосунок, який можна буде використовувати не лише в домашніх умовах, а й у рамках занять у навчальному закладі.

Таким чином, створюваний застосунок:

- використовуватиме технологію доповненої реальності;
- як пристрій відображення використовує мобільний пристрій (смартфон або планшет);

- має бути використано як допоміжний інструмент під час вирішення

стереометричних завдань зі шкільного курсу геометрії.

Користувач має роздрукований набір завдань з маркерами (мітками доповненої реальності). У нашому випадку маркери будуть розміщені на сторінках книги. Під час запуску програми на мобільному пристрої включається захоплення відеокамери. Користувач наводить камеру так, щоб маркер був у зоні видимості веб-камери, на екрані відображається прив'язаний до мітки тривимірний об'єкт, що є візуалізацією завдання, наприклад зображує якийсь об'єкт.

Мобільний застосунок, що надає доповнену реальність, може набувати різних форм, які обмежують користувачів у використанні його лише у певних операційних системах (так звані власні додатки) або, навпаки, дозволяють вільно використовувати застосунок у будь-якій операційній системі (так звані мультиплатформні додатки). Скорочення ризиків, зниження витрат, посилення терапевтичного ефекту, збереження витратних матеріалів є потенційними результатами використання технології AR у навчанні [8]. «Мобільна доповнена реальність особливо добре підходить для таких ідей, як «повсюдне навчання», в рамках якого передбачається, що кожна людина навчається постійно, де б вона не знаходилася, коли йому це потрібно. Розробка мобільного AR-проекту ґрунтується на принципах моделювання» [9].

Про ефективність застосування технологій доповненої реальності в освіті йдеться у багатьох джерелах. Наведемо кілька фрагментів із статей: «Технологія доповненої реальності дає можливість візуалізувати об'єкти, які важко собі уявити, перетворює їх на 3D-моделі, тим самим полегшуючи розуміння абстрактного та складного контенту. Додавання таких даних, як, наприклад, коротка біографія людини, цікаві факти, історичні дані про події, візуальні 3D-моделі сприяють поліпшенню сприйняття та полегшують розуміння навчального матеріалу. У багатьох випадках теоретичних знань недостатньо для отримання відповідних навичок та компетенцій у професійних галузях, тому доповнена реальність може слугувати меті

подолання розриву між теоретичними знаннями та практикою.» [10]. «Досі ситуація описувалася за допомогою тексту або, у виняткових випадках, аудіо з відео чи без нього. За допомогою технології віртуальної реальності конкретна ситуація може бути запрограмована декількома змінними та середовищами, в яких студент може діяти» [11].

М. Клаудіа том Дік і Тімоті Х. Юнг у своїй роботі «Augmented Reality and Virtual Reality: New Trends in Immersive Technology». 5,1» [12]. Незважаючи на це у AR додатку так само є свої недоліки, наприклад у статті «Про застосування технології доповненої реальності в процесі навчання математики та фізики» згадує: «До недоліків додатків доповненої реальності можна віднести їхню вузьку спрямованість на вивчення певного процесу, явища чи поняття».

Створення програми доповненої реальності означає вибір платформи доповненої реальності. Ці платформи є набором API та інструментів, що дозволяють розробнику створювати контент, що взаємодіє з реальним світом. Для реалізації технології доповненої реальності необхідні два основні програмні компоненти: трекінг та візуалізація.

Користувач має роздрукований набір завдань з маркерами (мітками доповненої реальності). У нашому випадку маркери будуть розміщені на сторінках книги. Під час запуску програми на мобільному пристрої включається захоплення відеокамери. Користувач наводить камеру так, щоб маркер був у зоні видимості веб-камери, на екрані відображається прив'язаний до мітки тривимірний об'єкт, що є візуалізацією завдання, наприклад зображує якийсь об'єкт.

Мобільний застосунок, що надає доповнену реальність, може набувати різних форм, які обмежують користувачів у використанні його лише у певних операційних системах (так звані власні додатки).

Щоб визначитися з вибором засобу розробки доповненої реальності, виберемо найбільш популярні SDK, і розглянемо їх переваги та недоліки (таблиця 1.2).

Таблиця 1.2 – Порівняння популярних засобів розробки AR

ПЗ	Опис	Переваги	Недоліки
DART	Набір інструментів для прототипування та створення програм з доповненою реальністю. Виконаний як застосунок до Macromedia Director	Підтримка різних сенсорів для трекінгу	Підтримує тільки ARToolkit і ARTag як двигуни для трекінгу
D'Fusion	Комерційний пакет програм для створення програм з доповненою реальністю	Хороший трекінг ключових точок	Погана документація, відсутність спільноти розробників, ціна
Metaio SDK	Комерційний пакет програм розробки доповненої реальності	Різноманітність функцій.	Висока ціна
flare	Набір бібліотек комп'ютерного зору для Adobe Flash/Flex для виявлення, відстеження та розрахунку позиції двовимірних штрих-кодів або зображень у відеопотоці веб-камери	Відмінний трекінг: стійкість до спотворення маркера та відблисків	Найвища ціна.
Qualcomm Vuforia (до 2011 QCAR)	Інструментарій для розробки доповненої реальності у вигляді розширення до Unity	Гарний трекінг. Безкоштовний	Закритий вихідний код

На даний момент дослідниками та фахівцями було напрацьовано велику теоретичну та алгоритмічну базу для їх реалізації як у вигляді різних

окремих компонентів, так і у вигляді інтегрованих програм та наборів засобів розробки доповненої реальності. До окремих компонентів відносяться бібліотеки та фреймворки комп'ютерного зору, двигуни тривимірної графіки, інші рішення.

«За допомогою Vuforia можна писати код доповненої реальності різними мовами програмування, її підтримує також і Unity, що дозволяє створювати мультиплатформні рішення. Vuforia добре і стабільно працює на великій кількості пристроїв» [4]. З перелічених переваг і недоліків мною був обраний інструментарій Qualcomm Vuforia.

2 ОПИС МОДЕЛІ ДОПОВНЕНОЇ РЕАЛЬНОСТІ ТА РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ

2.1 Методика комп'ютерного зору у доповненій реальності

Для реалізації технології доповненої реальності необхідне забезпечення стабільної роботи трекінгу на популярних споживчих пристроях. Однак ці пристрої, як правило, мають обмежені обчислювальні ресурси, працюють з урахуванням енергозбереження та здебільшого оснащені лише однією RGB-камерою. Це суттєво звужує спектр можливих методів реалізації трекінгу. У зв'язку з цим особливу увагу привертає оптичний трекінг, який активно досліджується та розвивається.

Оптичний трекінг є перспективним для масового впровадження, оскільки він може функціонувати з використанням лише однієї камери, без необхідності застосування спеціалізованого додаткового обладнання. Зазвичай під оптичним трекінгом мається на увазі така система доповненої реальності, в якій камера використовується одночасно як джерело інформації про реальний світ і як сенсор для обробки вхідного зображення з метою відстеження. При цьому відбувається визначення положення самої камери в просторі як точки спостереження.

Оскільки головне завдання доповненої реальності полягає в інтеграції віртуальних об'єктів у реальне середовище, виникає потреба у попередньому аналізі просторових даних. Часто маркерне розпізнавання реалізується за допомогою простого послідовного алгоритму, що включає шість основних кроків: перетворення кольорового зображення у відтінки сірого, подальшу бінаризацію, виявлення замкнених областей, виділення контурів, знаходження кутів маркера та перетворення координат [10].

Маркерна схема суттєво зменшує складність, характерну для традиційних методів комп'ютерного зору. Типові алгоритми комп'ютерного

зору здебільшого ефективні лише в контрольованих умовах — наприклад, при стабільному освітленні або специфічних формах об'єктів. Для подолання цих обмежень та зниження складності обробки, кожен об'єкт, який підлягає відстеженню, може бути позначений унікальним кодом, що його ідентифікує. Головне завдання такої кодувальної схеми — однозначно ідентифікувати об'єкти трекінгу та точно визначити їх положення без двозначності.

Обрані коди являють собою набори точок, розташованих у певному геометричному шаблоні з рівними інтервалами. Обов'язковою умовою є наявність яскраво виражених великих і малих осей, а також відсутність симетрії у візерунку. Крім того, передбачається, що всі точки на маркері мають однаковий колір, а фон — однорідний.

У разі використання чотирьох точок на головній осі та трьох на малій, кількість можливих комбінацій становить чотири. Загальна кількість варіантів у цьому випадку дорівнює шести. Якщо ж головна вісь має довжину 10, а мала — 9, тоді максимальна кількість об'єктів, які можна унікально ідентифікувати за допомогою маркерів, становить 530. Цей обсяг є більш ніж достатнім для більшості застосувань, у яких необхідно відстежувати декілька об'єктів одночасно. У разі використання декількох кольорів візерунки можуть повторюватися, що дозволяє ще більше розширити діапазон ідентифікації.

На першому етапі зображення обробляється шляхом згортки з ядром Гауса з метою зниження деталізації високих частот. Далі проводиться сегментація областей, інтенсивність яких подібна до інтенсивності опорних точок. Для цього застосовується один із двох підходів, вибір яких залежить від типу вхідного відеопотоку (кольорове або напівтонове): порогова обробка або колірна сегментація.

У першому випадку застосовується бінаризація на основі порогового значення, де виділяються яскраві точки на темному тлі або навпаки. Основна перевага цього методу — знижена чутливість до змін освітлення. Проте головним недоліком є висока ймовірність появи шумів, зумовлених

випадковими яскравими або темними плямами в кадрі.

Другий підхід – колірна сегментація – дозволяє використовувати маркери з різними кольорами, як для точок, так і для фону. Такий підхід зменшує рівень шуму, але водночас чутливий до змін умов зйомки: освітлення, кута огляду тощо, що може ускладнити точну ідентифікацію точок у складному середовищі.

У результаті обробки формується чорно-біле зображення, де білі області відповідають кольоровим точкам, які вдалося виявити. На наступному етапі зменшена версія бінаризованого зображення (із кроком вибірки 8, 4 або 2 пікселі) сканується на наявність білих областей. При виявленні такої області запускається алгоритм вирощування області, який обчислює центр кожної яскравої плями, фіксуючи координати відповідних пікселів у повному зображенні.

Яскраві області можуть виникати внаслідок бликів, кольорових поверхонь, текстур або фонового шуму. Для підвищення точності та продуктивності надто великі області, які не відповідають критеріям, відкидаються. Після цього здійснюється фінальна оцінка форм виявлених областей, що дозволяє відокремити релевантні маркери від випадкових об'єктів.

Обчислюються їх співвідношення сторін, евклідова відстань між його центром мас і центром його огорожувальної коробки, а також інші співвідношення. Накладаючи хрест, розташований над центром ваги плями, ми вважаємо пляму дійсною, якщо є повне перекриття у всіх чотирьох сторонах цього хреста. Також розраховуються інші співвідношення, пов'язані з формою, такі як горизонтальні та вертикальні розміри. Виправлення, які виходять за межі заданих діапазонів цих коефіцієнтів, видаляються.

Наступна мета – згрупувати дійсні точки. Розмір кластерів повинен бути в межах кількості точок, знайдених у наборі маркерів. Алгоритм з'єднує центроїди всіх обраних ділянок, розглядаючи їх як вершини у графі та генеруючи мінімальне сполучне дерево (МСД) між ними. МСД групує

ділянки в кластери відповідно до локальної близькості. Обчислюються статистичні дані, такі як середня довжина та стандартне відхилення всіх гілок МСД, а довжина ребер зберігається у списку суміжності або поданні матриці суміжності. Другим етапом проблеми зондування в галузі збирання є трекінг. Трекінг це динамічний визначення/обчислення просторових властивостей під час роботи програми Щоб відображати віртуальні об'єкти в тривимірному просторі, ми повинні знати їх відносне становище по порівнянні з відносним становищем розташування та об'єктів реального світу. Оскільки AR часто є процесом у реальному часі, ці вимірювання повинні постійно оновлюватися, щоб вони залишалися точними. Алгоритм відстеження складається з кількох модулів. До них відносяться: попередня обробка зображення, виявлення виправлень, генерація мінімального сполучного дерева, усунення гілок викидів, вилучення осей, ідентифікація деталей та оцінка положення у 3D.

Алгоритм визначає ймовірні головні та другорядні осі в кожному кластері шляхом вимірювання орієнтації кожної гілки та знаходження двох можливих орієнтацій осей. Обчислюється довжина кожної осі, що визначається як кількість точок, і визначається рамка, що обмежує навколо маркера. Підпрограма визначення осі виводить три неколінеарні точки, елемент периметра прямокутника. Правила, що використовуються для ідентифікації об'єктів, можуть ґрунтуватися на:

- загальна кількість точок;
- довжина кожної осі;
- відносне становище осей щодо один одного.

Ці правила ще більше скоротять кількість допустимих геометричних кластерів, створюваних шумом. На рисунку 2.1 показані всі виявлені точки, а також МСД, кластеризація точок та розпізнавання маркерів

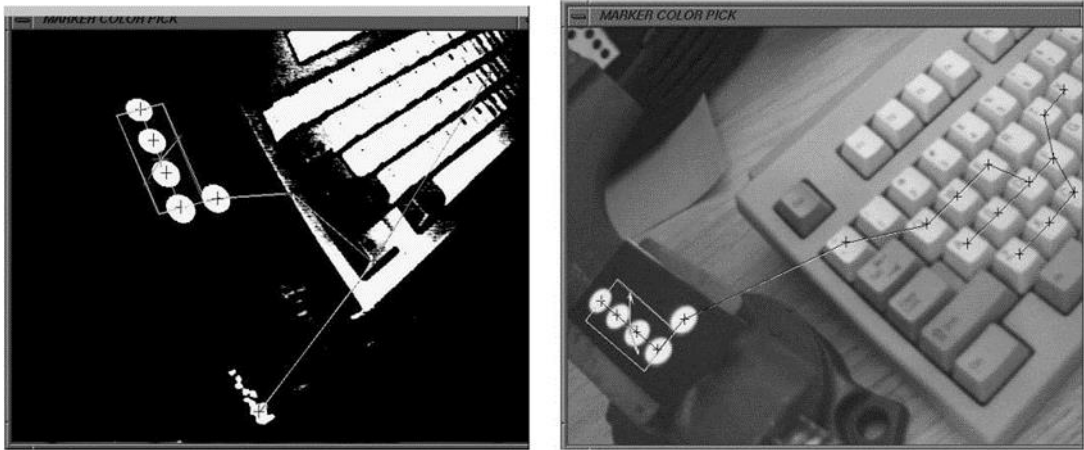


Рисунок 2.1 – Процес виявлення точок

Крапки, що відповідають маркерам, були згруповані у підмножини, і єдині кластери, розпізнані як маркери, отримують заключну рамку. Згодом оцінюється тривимірне становище маркера. Три неколінеарні точки, вершини рамки, що обмежує, передаються в процедуру оцінки тривимірної пози. Відносне становище цих точок відомо, і може бути збережено в масиві з індексом кожного маркера.

Правила для ідентифікації об'єктів ґрунтуються на особливостях коду та можуть бути організовані ієрархічним чином. Приклад правила: якщо код містить n точок, то деталь належить класу об'єктів A . Якщо на ньому $n + 1$ точок, то деталь відноситься до класу об'єктів B . Якщо головна вісь має m точок, а деталь є елементом множини A , деталь належить підкласу $A1$, в іншому випадку вона належить підкласу. Застосування ієрархічних правил забезпечує швидке розпізнавання та природну організацію типів маркерів. Правила для розпізнавання маркерів, що використовуються в даний час, пов'язані з розміром осей та їх відносним положенням щодо один одного.

3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОЄКТУ

Обґрунтування вибору інструментарію для розробки базується на застосуванні платформи Qualcomm Vuforia, яка спеціалізується на реалізації технологій доповненої реальності. В якості середовища розробки обрана Unity, оскільки вона забезпечує кросплатформену розробку мобільних додатків з оптимізацією за розміром та продуктивністю.

Враховуючи орієнтацію проєкту на мобільні пристрої, було прийнято рішення зосередитись на платформі Android, що забезпечує широке охоплення користувачів і сумісність із популярними апаратними засобами.

Для інтеграції AR-контенту у застосунок необхідно отримати ліцензійний ключ на офіційному ресурсі Vuforia [11], який є обов'язковим елементом налаштувань проєкту в Unity.

Використання Unity у поєднанні з бібліотекою Vuforia дозволяє реалізувати високопродуктивний AR-застосунок з компактним розміром інсталяційного пакету, ефективно використовувати ресурси мобільних пристроїв, а також забезпечити функціональність, необхідну для точного розпізнавання маркерів і відображення 3D-моделей у реальному часі.

3.1 Реалізація програмного модуля

Для розробки програми в Unity для початку необхідно визначитися з платформою. Так як ми розробляємо мобільний додаток найкращим варіантом буде Android. Заходимо в налаштування проєкту (рисунок 3.1), та виставляємо бажану платформу [12].

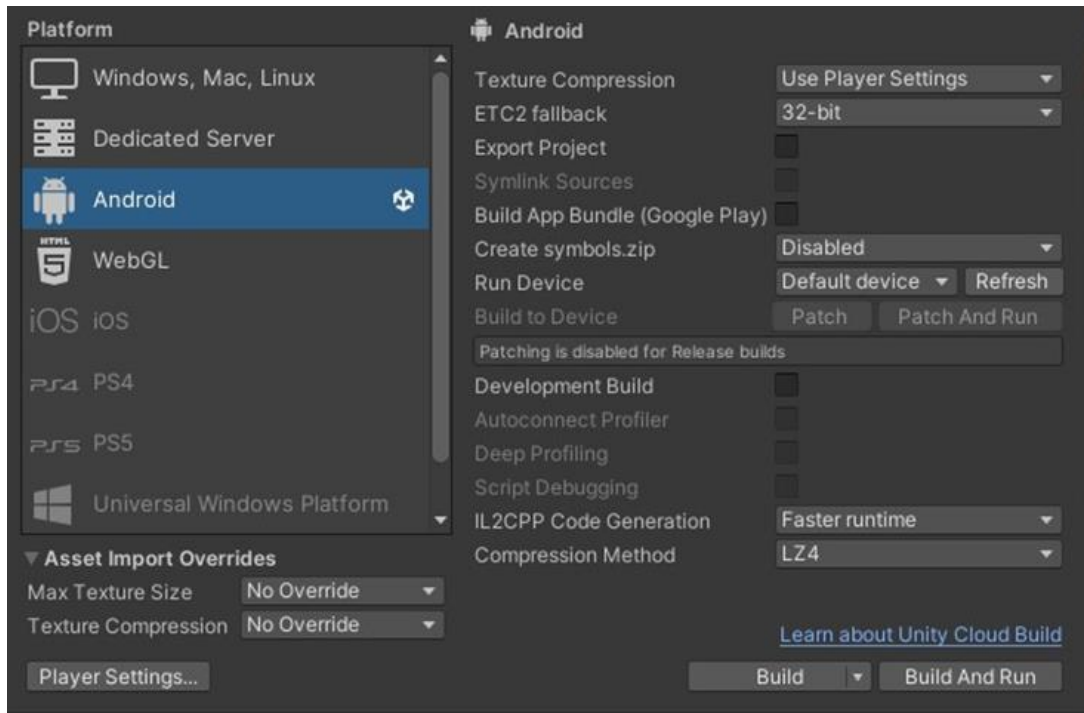


Рисунок 3.1 – Вибір платформи

Наступним етапом є створення сцени та додавання до неї AR-камери як ключового компонента для реалізації функціоналу доповненої реальності. Для забезпечення коректної роботи програми, а також правильного відображення 3D-моделей та віртуального контенту, усі об'єкти сцени (маркери, моделі тощо) повинні бути організовані як дочірні елементи AR-камери [13].

Після цього необхідно сформувати набір маркерів і імпортувати його до поточного проєкту. З урахуванням структури інтерактивної книги було прийнято рішення реалізувати п'ять окремих сцен, що відповідають п'яти унікальним маркерам – по одному для кожної сторінки книги, де відтворюватиметься конкретна дорожньо-транспортна ситуація за участю дитини.

Для наповнення сцен було обрано п'ять типових ситуацій із попередньо підготовленого текстового матеріалу, а саме:

- перехід дитиною дороги по пішохідному переходу;
- пересування дитини поблизу проїжджої частини;
- очікування дитиною громадського транспорту на зупинці;

- перебування дитини всередині транспортного засобу;
- рух дитини на велосипеді.

На рисунку 3.2 представлено лістинг ієрархічної структури об'єктів сцени, що ілюструє логіку їх розміщення та зв'язків у межах проєкту.



Рисунок 3.2 – Ієрархія об'єктів

Щоб почати додавати до програми моделі та маркери необхідно створити ліцензію та отримати ліцензійний ключ на сайті Vuforia [13]. Даний ключ (рисунок 3.3) необхідний для подальшої розробки в Unity.

License Manager > For TSU Diploma

For TSU Diploma

Edit Name Delete License Key

License Key Usage

Please copy the license key below into your app

```
AXhFoJ3/////AAABmT/IRF/1iEj0jLVpu6Qv5xkx/F8+K26jie/ldn0JrDFp1E1VAeBKwoYifALwvhQ7kYIq1+X01PtcCaNKj968Bhr
WVeSwfbDXotXB422Wgp1/6dH9chxJYWAzyYnICYLf/vzTFmkFwizGnmgiE2DoK2zDxYYYi7aD5xgXZdc/phJI79t6cuAIYbw0K02E5t
SqnLKYaata07MSraECFq1KbjuhVEecVfD9dB4iC1Q1r+p+2kjJtqC6bfedETC2cOh7x56Y9LN8cWUVjPAzWfnO7XNdBZGKf4HlyU2+T
RR5z7D1hJ1cW7ke/xG02RwoeGoffFeQUqZMoF4mKa0ZL8nwXGkOoV8tatXurDVCF1Jxxv9Z
```

Plan Type: Basic
Status: Active
Created: May 05, 2025 20:24
License UUID: 1db33712607547ff95a70dc41879b6aa

Permissions:

- Advanced Camera
- External Camera
- Model & Area Targets
- Watermark

Рисунок 3.3 – Ключ ліцензії

Після створення ліцензії та отримання ліцензійного ключа необхідним наступним кроком є створення маркера, на поверхні якого буде здійснюватися проєкція 3D-моделі. При розробці маркера доцільно орієнтуватися на низку базових принципів: рівномірне розташування опорних точок, максимальна довжина головних осей та оптимальне покриття зображення.

З метою підвищення точності визначення просторового положення маркера додатково рекомендовано розмістити контрольні точки в кутах зображення. Це забезпечує більш стабільне розпізнавання з різних ракурсів та за умов часткової оклюзії. На рисунку 3.4 продемонстровано етапи створення першого маркера відповідно до зазначених критеріїв.

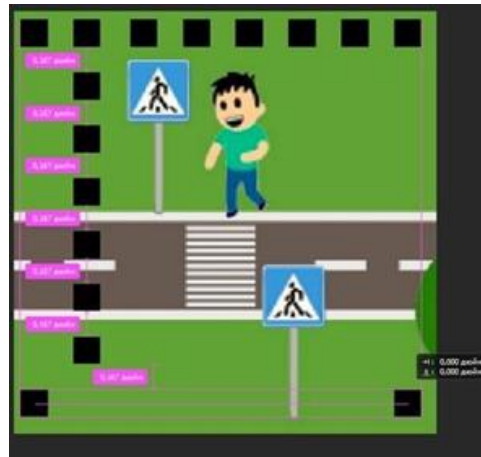


Рисунок 3.4 – Створення першого маркера

Аналогічним чином виконується розміщення контрольних точок для решти маркерів, при цьому важливо щоразу використовувати унікальну конфігурацію їх розташування, щоб забезпечити однозначне розпізнавання кожного маркера.

Після завантаження зображень-маркерів у вебпортал Vuforia, вони проходять попередню обробку спеціалізованим алгоритмом, після чого експортуються у вигляді спеціального пакета, сумісного з Unity. Після імпорту цього пакета до проєкту, у компоненті ImageTarget можна обрати відповідний набір маркерів для використання в сцені [6].

Система автоматично оцінює якість розпізнавання кожного маркера за п'ятибальною шкалою, що дозволяє розробнику обрати оптимальні варіанти для подальшого використання в додатку. На рисунку 3.5 представлено процес імпорту маркерів у середовище розробки та відповідні індикатори оцінки їх розпізнаваності.

AR_PDD_TEST [Edit Name](#)
Type: Device

Targets (5)

Add Target Download Database (All)






<input type="checkbox"/> Target Name	Type	Rating [ⓘ]	Status [▼]	Date Modified
<input type="checkbox"/>  exmpl5	Single Image	★★★★★	Active	March 13, 2025 22:03
<input type="checkbox"/>  exmpl4	Single Image	★★★★★	Active	March 13, 2025 22:03
<input type="checkbox"/>  exmpl3	Single Image	★★★★★	Active	March 13, 2025 22:03
<input type="checkbox"/>  exmpl2	Single Image	★★★★★	Active	March 13, 2025 22:03
<input type="checkbox"/>  exmpl1	Single Image	★★★★★	Active	March 13, 2025 22:03

Рисунок 3.5 – База даних маркерів

Як видно з результатів, наші маркери отримали оцінку системи в діапазоні від 3 до 4 зірок. Це свідчить про задовільний рівень розпізнавання та відстеження маркерів у мобільному додатку, хоча в окремих випадках можливі труднощі з трекінгом, зокрема за складних умов освітлення або при частковій оклюзії.

На наступному етапі необхідно завантажити створену базу даних маркерів із порталу Vuforia та імпортувати її до середовища розробки (рисунок 3.6).

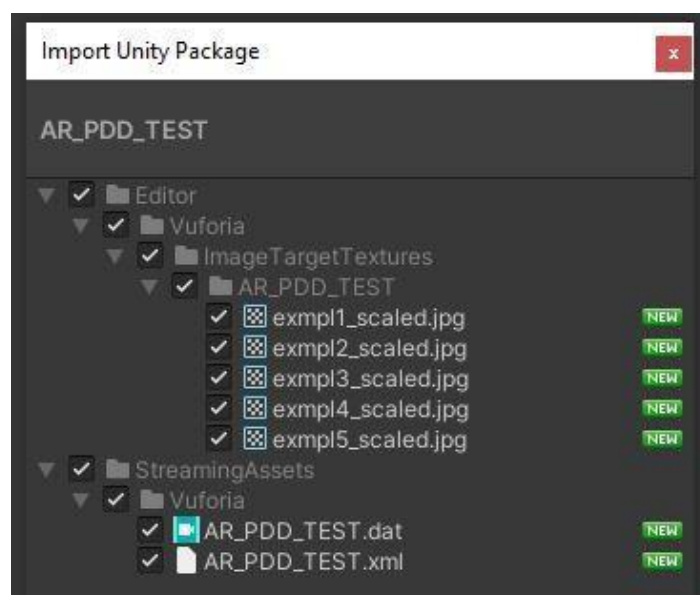


Рисунок 3.6 – Імпорт бази даних маркерів у проект

Після цього в ієрархії сцени до AR-камери додається дочірній об'єкт

типу Image Target. У його властивостях необхідно вказати відповідну базу даних маркерів, після чого обрати конкретний маркер, який використовуватиметься в поточній сцені (рисунок 3.7). Це забезпечить відображення відповідної 3D-моделі при розпізнаванні маркера камерою пристрою.

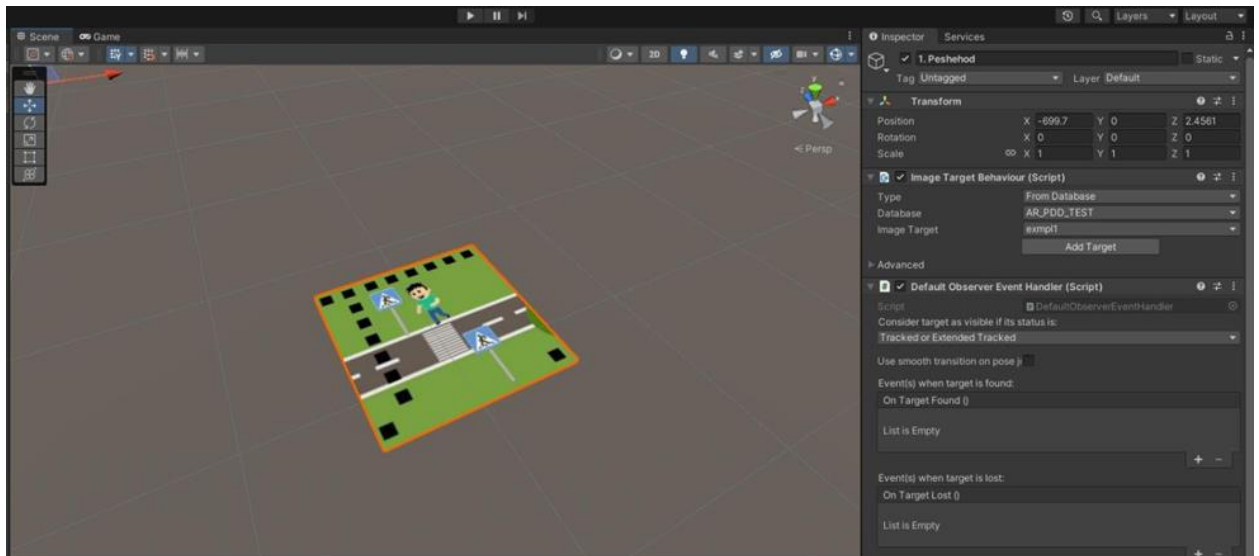


Рисунок 3.7 – Маркер поміщений у віртуальний простір

На наступному етапі виконується створення тривимірної моделі для відповідної сцени. Для цього використовується бібліотека безкоштовних ігрових ресурсів (асетів), що включає 3D-моделі, текстури та інші графічні об'єкти, необхідні для реалізації сцени.

Враховуючи, що сюжетна дія кожної сцени розгортатиметься безпосередньо на сторінці друкованої книги, базову модель сцени необхідно масштабувати та розмістити на маркері відповідно до фактичних геометричних параметрів сторінки. Це забезпечить коректне позиціонування віртуального об'єкта в доповненій реальності при розпізнаванні маркера.

На рисунку 3.8 представлено приклад розташування основи моделі сцени на поверхні маркера.

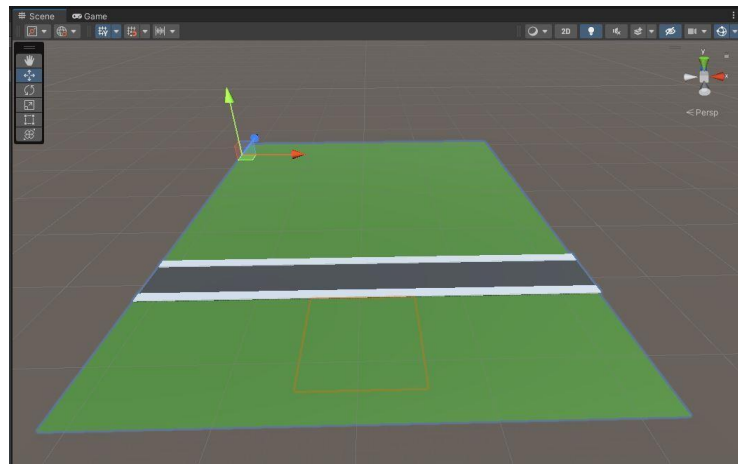


Рисунок 3.8 – Основа моделі на маркері

Як показано на рисунку 3.9, червоними лініями окреслено межі розташування маркера, який слугує основою для відображення віртуальної сцени в доповненій реальності. Наступним етапом є побудова макета міського середовища шляхом використання заздалегідь підготовлених тривимірних моделей (будівель, доріжок, інфраструктурних об'єктів) та відповідних текстур.

Початково здійснюється розміщення основних об'єктів – будівель та пішохідних стежок – з урахуванням масштабування сцени відносно розмірів маркера та пропорційності всієї композиції. Це формує базову структуру середовища, в якому відбуватимуться події навчального сценарію.

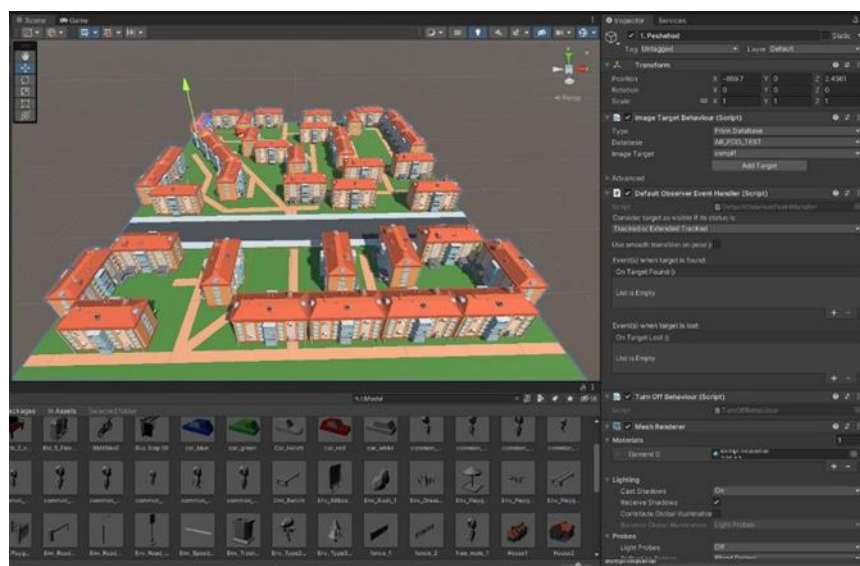
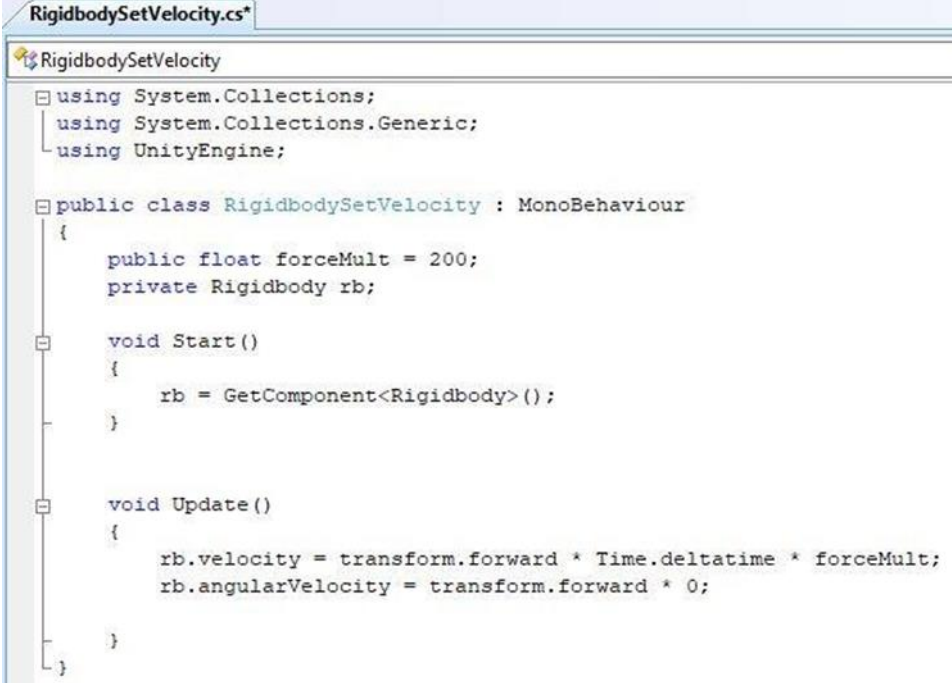


Рисунок 3.9 – Готова модель міста у віртуальному просторі

Тепер, коли макет міста, що відображається на мітці, готовий необхідно задати рух персонажу, а також машинам. Для цього створимо клас `RigidbodySetVelocity`, лістинг якого представлений на рисунку 3.10.



```

RigidbodySetVelocity.cs*
RigidbodySetVelocity
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class RigidbodySetVelocity : MonoBehaviour
{
    public float forceMult = 200;
    private Rigidbody rb;

    void Start()
    {
        rb = GetComponent<Rigidbody>();
    }

    void Update()
    {
        rb.velocity = transform.forward * Time.deltaTime * forceMult;
        rb.angularVelocity = transform.forward * 0;
    }
}

```

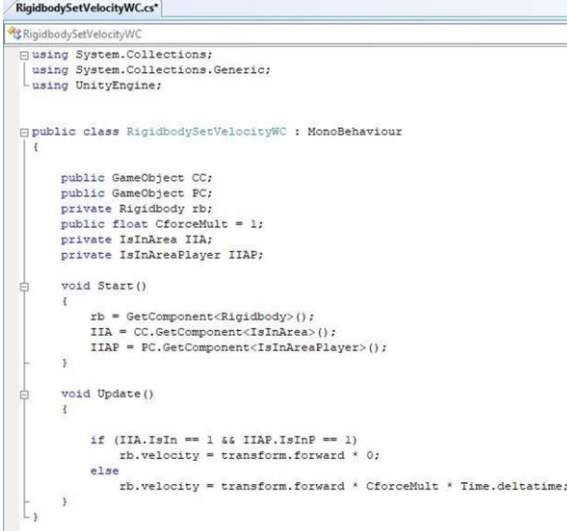
Рисунок 3.10 – Лістинг класу `RigidbodySetVelocity`

У цьому класі реалізовано метод `Update`, який виконується кожного кадру – це характерно для додатків на базі Unity, де оновлення сцени відбувається в режимі реального часу. Такий підхід дозволяє динамічно керувати поведінкою об’єктів у просторі доповненої реальності.

У нашому випадку перед кожним рендерингом кадру об’єкту задається швидкість, яка розраховується як добуток напрямку руху, сили впливу та проміжку часу (`deltaTime`). Це забезпечує плавний та контрольований рух елементів сцени.

Застосувавши цей клас до моделей транспортного засобу та персонажа, під час запуску додатку спостерігається рух об’єктів у заданому напрямку. Однак просте лінійне переміщення є недостатнім, оскільки це може призвести до зіткнень з іншими об’єктами сцени або до виходу за межі міської моделі.

Для реалізації логіки, за якої персонаж зупиняється у разі присутності автомобіля на шляху (наприклад, між «лежачими поліцейськими»), було створено новий клас `RigidbodySetVelocityWC`. У цьому класі реалізовано поведінкові обмеження, що дозволяють динамічно реагувати на зміну ситуації в середовищі. Лістинг даного класу представлено на рисунку 3.11.



```

RigidbodySetVelocityWC.cs
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class RigidbodySetVelocityWC : MonoBehaviour
{
    public GameObject CC;
    public GameObject PC;
    private Rigidbody rb;
    public float CforceMult = 1;
    private IsInArea IIA;
    private IsInAreaPlayer IIAPlayer;

    void Start()
    {
        rb = GetComponent<Rigidbody>();
        IIA = CC.GetComponent<IsInArea>();
        IIAPlayer = PC.GetComponent<IsInAreaPlayer>();
    }

    void Update()
    {
        if (IIA.IsIn == 1 && IIAPlayer.IsInP == 1)
            rb.velocity = transform.forward * 0;
        else
            rb.velocity = transform.forward * CforceMult * Time.deltaTime;
    }
}

```

Рисунок 3.11 – Лістинг класу `RigidbodySetVelocityWC`

У модифікованій версії попереднього класу реалізовано розширену логіку взаємодії об'єктів сцени шляхом введення додаткових умов. Зокрема, було оголошено дві логічні змінні:

- `IsInArea` (IIA) – індикатор присутності транспортного засобу в зоні пішохідного переходу;
- `IsInAreaPlayer` (IIAPlayer) – індикатор розташування персонажа безпосередньо перед пішохідним переходом.

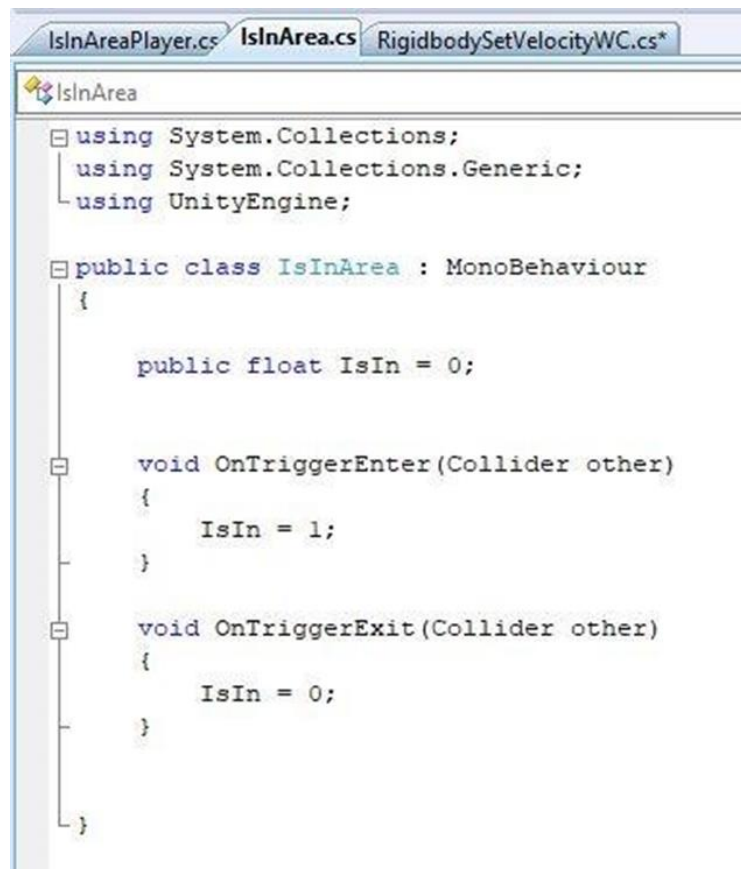
Ці змінні оголошені з модифікатором доступу `public`, що дозволяє звертатися до них із інших класів. Це забезпечує взаємодію між компонентами програми та дозволяє динамічно реагувати на зміну просторового розташування об'єктів у середовищі Unity.

У методі `Update` реалізована логіка, яка контролює рух персонажа: якщо персонаж перебуває перед пішохідним переходом (`IIAPlayer == true`) і в цей

момент автомобіль присутній у межах переходу (ПА == true), то швидкість персонажа встановлюється рівною нулю. У протилежному випадку – коли умова не виконується – персонаж продовжує рух із розрахованою швидкістю (як добуток напрямку, сили і проміжку часу).

Таким чином, у моделі реалізовано базову поведінкову модель безпечного переходу дороги: персонаж чекає, поки транспортний засіб покине небезпечну зону, що імітує реальну поведінку пішохода на переході.

Для виявлення перебування об'єктів у відповідних зонах було створено окремі класи: `IsInArea` — для транспортних засобів і `IsInAreaPlayer` – для персонажа. У класах використовується система тригерів (`OnTriggerEnter`, `OnTriggerExit`) Unity для визначення входу та виходу об'єкта з контрольованої області. Код цих класів наведено у лістингах на рисунках 3.12 та 3.13.



```

IsInAreaPlayer.cs  IsInArea.cs  RigidbodySetVelocityWC.cs*
IsInArea
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

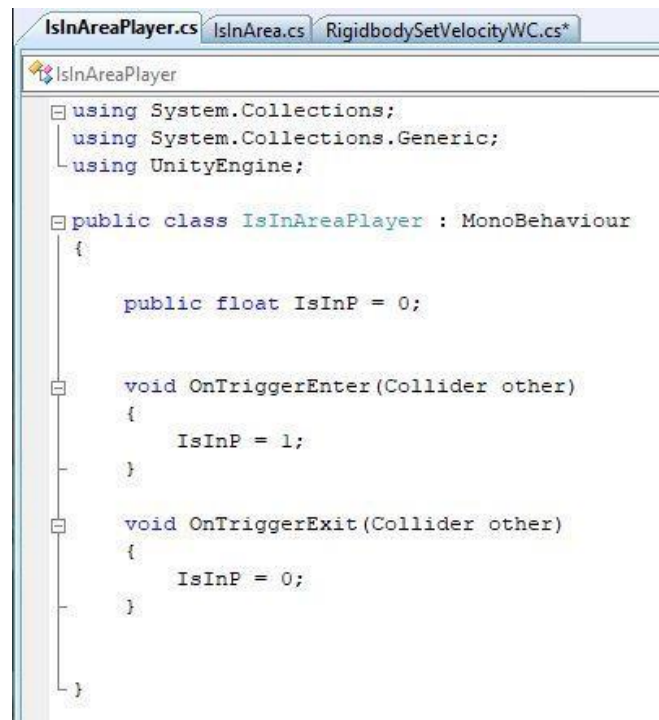
public class IsInArea : MonoBehaviour
{
    public float IsIn = 0;

    void OnTriggerEnter(Collider other)
    {
        IsIn = 1;
    }

    void OnTriggerExit(Collider other)
    {
        IsIn = 0;
    }
}

```

Рисунок 3.12 – Лістинг класу `IsInArea`



```

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

public class IsInAreaPlayer : MonoBehaviour
{
    public float IsInP = 0;

    void OnTriggerEnter(Collider other)
    {
        IsInP = 1;
    }

    void OnTriggerExit(Collider other)
    {
        IsInP = 0;
    }
}

```

Рисунок 3.13 – Лістинг класу IsInAreaPlayer

Дані класи реалізують функціональність відслідковування перебування об'єктів (персонажа або транспортного засобу) у визначених зонах за допомогою вбудованих подій OnTriggerEnter та OnTriggerExit:

- метод OnTriggerEnter викликається при вході об'єкта у межі тригер-зони, змінна ПА (для автомобіля) або ПАР (для персонажа) набуває значення true (1), сигналізуючи про присутність об'єкта у відповідній області;
- метод OnTriggerExit активується, коли об'єкт покидає межі тригер-зони, і відповідна змінна повертається у стан false (0).

Ця логіка дозволяє точно відстежувати перебування об'єктів у критичних зонах, таких як пішохідний перехід, і динамічно керувати поведінкою персонажа залежно від дорожньої ситуації.

Наступним етапом є створення віртуальних об'єктів-середовища, які виконуватимуть функцію граничних зон. Ці об'єкти мають бути помічені як тригери (IsTrigger), і до них необхідно приєднати відповідні скрипти IsInArea або IsInAreaPlayer, залежно від їх призначення. На рисунку 3.14 наведено приклад створеної тригер-зони.

Як зазначають фахівці, діти у процесі навчання часто діють за

повторюваним шаблоном [10]. У контексті даного застосунку, усі об'єкти здійснюють рух у прямому напрямку без зупинки, а анімаційна сцена, за замовчуванням, відтворюється лише один раз під час запуску. Така поведінка може бути доопрацьована для підтримки повторюваності сценаріїв, що є важливим аспектом для навчальних систем із залученням дітей.



Рисунок 3.14 – Область пішохідного переходу

Щоб вирішити цю проблему, створимо новий клас Teleporting, за допомогою якого ми зможемо зациклити анімацію об'єктів, завдяки чому користувач програми зможе без перезапуску що.разу переглядати анімацію з різних сторін. Лістинг класу представлений на рисунку 3.15.

```

Teleporting.cs | IsInAreaPlayer.cs | IsInArea.cs | RigidbodySetVelocityWC.cs*
Teleporting
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Teleporting : MonoBehaviour
{
    public Transform teleportTarget;
    public GameObject thePlayer;

    void OnTriggerEnter(Collider other)
    {
        thePlayer.transform.position = teleportTarget.transform.position;
    }
}

```

Рисунок 3.15 – Лістинг класу Teleporting

Цей клас також використовує функцію `OnTriggerEnter`, завдяки якій в момент зіткнення персонажа з першим об'єктом [13] (назвемо його входом) персонажу присвоюватимуться координати другого об'єкта (виходу) (рисунок 3.16). Залишається створити вхід і вихід для кожного об'єкта, і прив'язати до них клас `Teleporting`. Щоб отримати ефект реалістичності точки входу та виходу для персонажа встановимо в під'їздах будинків, таким чином персонаж не просто телепортуватиметься з однієї точки в іншу, а ніби виходитиме і входить в під'їзд.

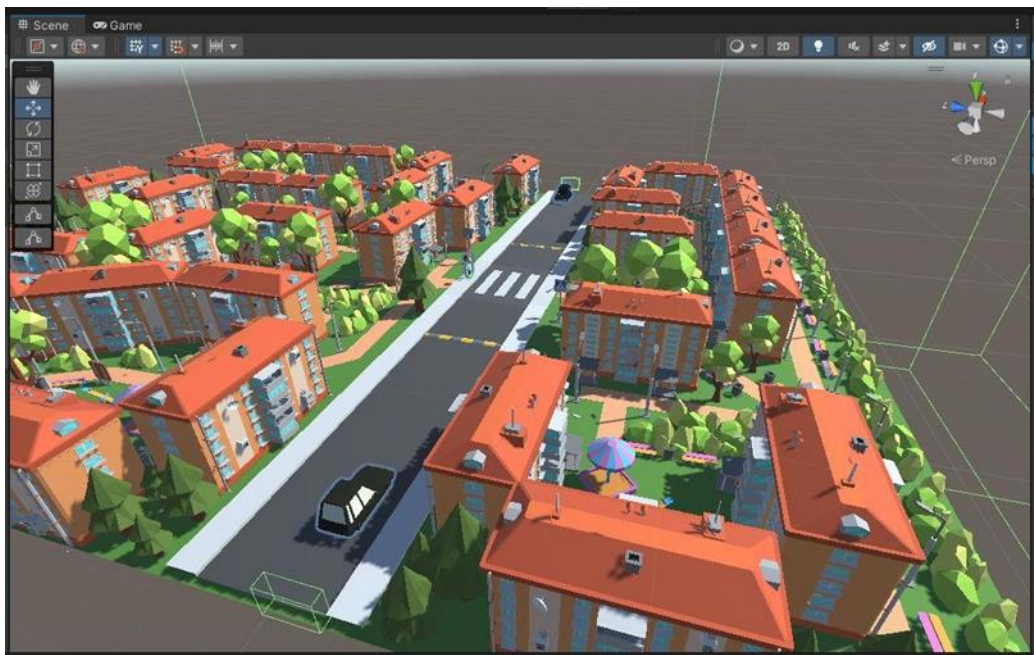


Рисунок 3.16 – Точки входу та виходу для об'єктів, що рухаються

Після запуску додатку та наведення камери на відповідний маркер, система розпізнає зображення, активує AR-сцену та відображає створений віртуальний макет міста. Як показано на рисунку 3.17, віртуальне середовище включає дорожню інфраструктуру з анімованими транспортними засобами та персонажем, який здійснює перехід через проїжджу частину.

На основі вже реалізованої першої сцени, з використанням імпортованих 3D-моделей, налаштованих скриптів і базового міського макету, здійснюється розробка решти чотирьох сценаріїв. Для кожної нової сцени адаптуються візуальні елементи та параметри поведінки об'єктів

відповідно до змодельованої дорожньої ситуації.

Зокрема, для другої, третьої та п'ятої сцен модифікується існуючий ландшафт, змінюється розміщення головного персонажа, а також координати точок входу та виходу об'єктів. У другій сцені додатково інтегрується 3D-модель автобусної зупинки. Четверта сцена, яка відображає ситуацію всередині транспортного засобу, розробляється окремо, оскільки її середовище суттєво відрізняється – дія відбувається у лісовій зоні, що вимагає створення нової сцени з відповідним антуражем і об'єктами. Такий підхід дозволяє гнучко налаштовувати AR-сценарії відповідно до контексту навчальних матеріалів, підтримуючи інтерактивність і реалізм у візуалізації правил дорожнього руху.



Рисунок 3.17 – Персонаж переходить дорогу

Відтворити правдоподібний ліс досить швидко можна шляхом копіювання груп дерев різних розмірів, розставлених хаотично. На рисунку

3.18 продемонстровано всі готові сцени.

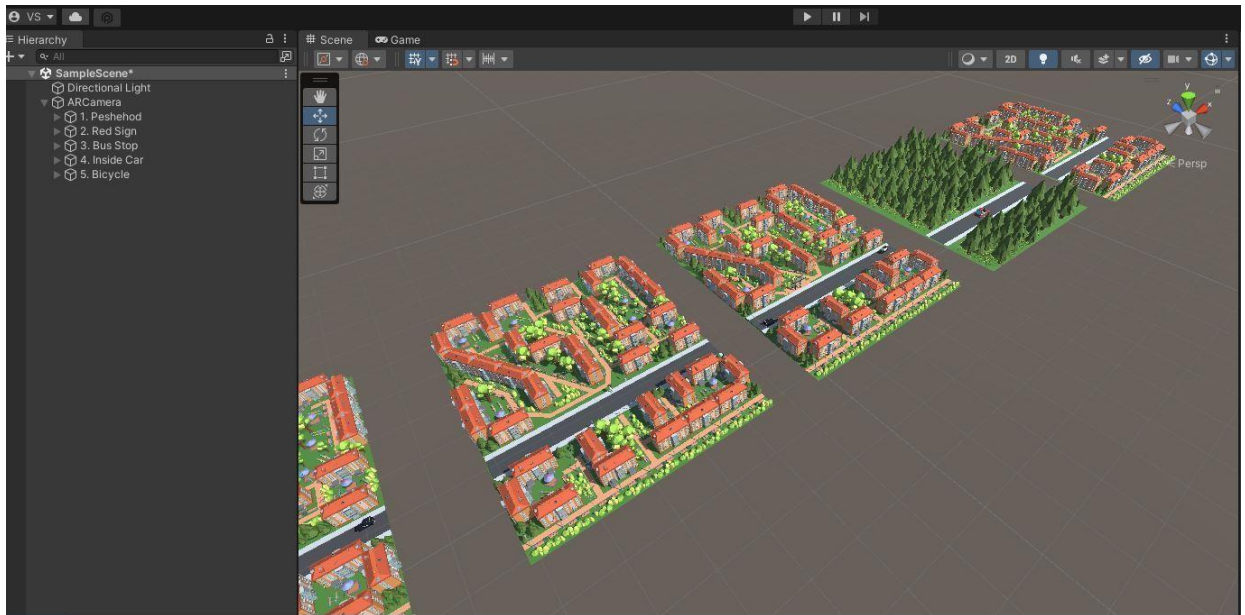


Рисунок 3.18 – Усі макети

Останнім етапом є збірка проєкту. У Unity відкриваємо Build Settings, обираємо платформу Android та натискаємо Build. Після завершення отримаємо .apk файл, який переносимо на смартфон для встановлення та тестування роботи застосунку.

4 ТЕСТУВАННЯ ТА КОРИГУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО МОДУЛЯ

4.1 Проведення обчислювальних експериментів

Після встановлення програми на мобільний пристрій запускаємо її для перевірки працездатності. Навівши камеру на перший маркер, переконуємось у правильному відображенні 3D-моделі (рисунок 4.1). Фіксуємо час, необхідний додатку для розпізнавання мітки та виведення моделі на екран, щоб оцінити швидкість реакції системи.



Рисунок 4.1 – Модель на маркері у реальному світі

Вимірюючи час розпізнавання позначки, отримуємо наступний результат:

- сцена з пішохідним переходом: 6,3 секунди;
- сцена біля проїжджої частини: 7,6 секунд;
- сцена з автобусною зупинкою: 4,8 секунд;
- сцена за містом: 3,2 секунди;
- сцена з велосипедом: 7,8 секунд.

Розрахуємо середній час розпізнавання позначок – 5,94 секунди. Результат не найкращий, тому що в ідеалі для надання користувачеві кращого досвіду, додаток повинен практично миттєво розпізнавати маркери. Далі необхідно перевірити як добре додаток відстежує мітку та розміщує поверх неї модель. На рисунку 4.2 відображено роботу програми при наведенні камери на мітку під кутом.



Рисунок 4.2 – Некоректне відображення моделі

Під час тестування було виявлено, що модель відображається некоректно – зокрема, її положення зміщується відносно маркера. Це пов'язано з тим, що система погано відстежує положення маркера під кутом, коли деякі контрольні точки виходять із поля зору камери. Якщо користувач

спробує оглянути сцену зверху, модель зникає через повне зникнення маркера з кадру.

Ще однією проблемою є залежність швидкості анімації від технічних характеристик пристрою. Оскільки методи класів орієнтовані на кількість кадрів у секунду (FPS), на малопродуктивних пристроях анімація відтворюється з затримкою, тоді як на потужніших – коректно. Це може спричинити відмінності у поведінці додатку на різних пристроях.

4.2 Оптимізація розробленої інформаційної технології

З метою забезпечення стабільного та коректного візуального відображення 3D-моделі на площині друкованої сторінки незалежно від кута огляду, було прийнято рішення масштабувати область маркера до розмірів повної сторінки книги. Такий підхід дозволяє суттєво спростити процедуру трекінгу – навіть у випадку часткового виходу маркера за межі поля зору камери, система продовжує успішно визначати його положення у просторі. Крім того, користувач отримує можливість вільного перегляду будь-якої частини сцени без ризику втрати візуалізації об'єктів через втрату маркером трекінгової площини.

На рисунку 4.3 представлено оновлений варіант маркера, в якому традиційне монохромне (чорне) заповнення замінено на комбінацію чорно-білих візерунків. Така модифікація дозволяє збільшити кількість розпізнаваних ключових точок із 4 до 12 на кожну область, що суттєво підвищує точність і стабільність трекінгу [14].

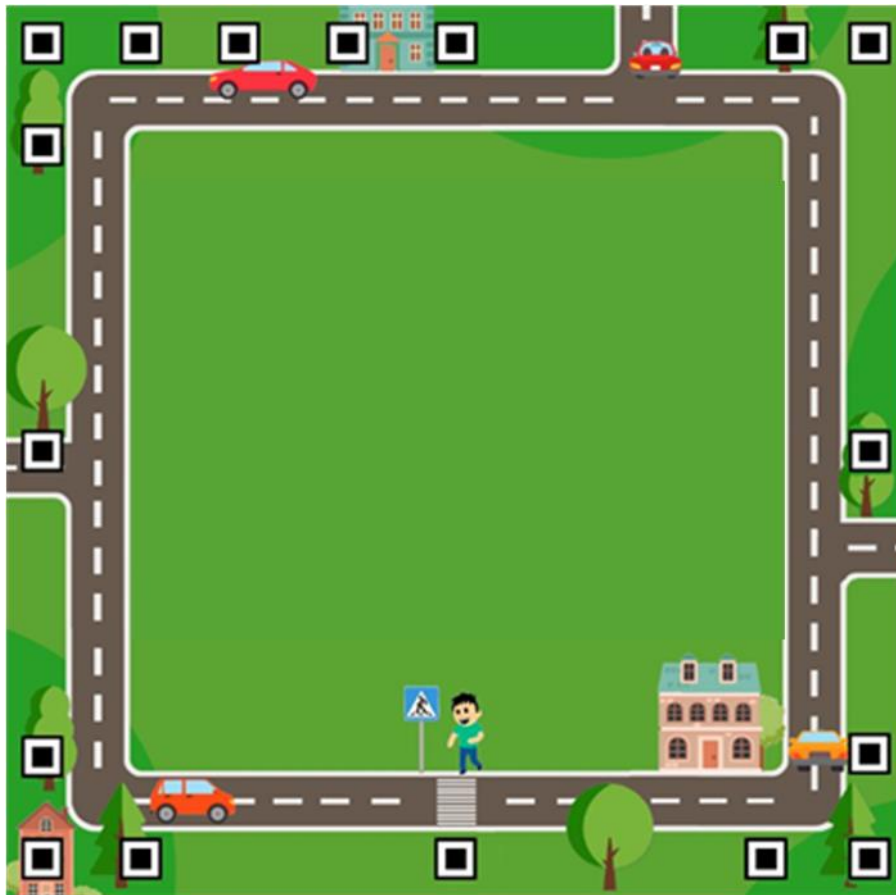


Рисунок 4.3 – Нова версія маркера

Відповідні зміни було впроваджено для кожної з п'яти сторінок. Оновлені маркери були завантажені на платформу Vuforia Developer Portal, де автоматизована система надала кожному з них найвищий рівень якості – 5 зірок (рисунок 4.4). Це свідчить про високий потенціал стабільного розпізнавання та ефективного використання у доповненій реальності (AR) на мобільних пристроях.

ARTR_BOOK_updated Edit Name
Type: Device

Targets (5)

Add Target Download Database (All)






<input type="checkbox"/> Target Name	Type	Rating ⓘ	Status ▾	Date Modified
<input type="checkbox"/>  NewPageMark5	Single Image	★★★★★	Active	March 15, 2025 13:57
<input type="checkbox"/>  NewPageMark4	Single Image	★★★★★	Active	March 15, 2025 13:57
<input type="checkbox"/>  NewPageMark3	Single Image	★★★★★	Active	March 15, 2025 13:56
<input type="checkbox"/>  NewPageMark2	Single Image	★★★★★	Active	March 15, 2025 13:56
<input type="checkbox"/>  NewPageMark1	Single Image	★★★★★	Active	March 15, 2025 13:56

Рисунок 4.4 – Модель на оновленому маркері

За аналогією з минулим разом імпортуємо в проект новий набір міток та розміщуємо модель на мітці щодо розміру сторінки (рисунок 4.5).

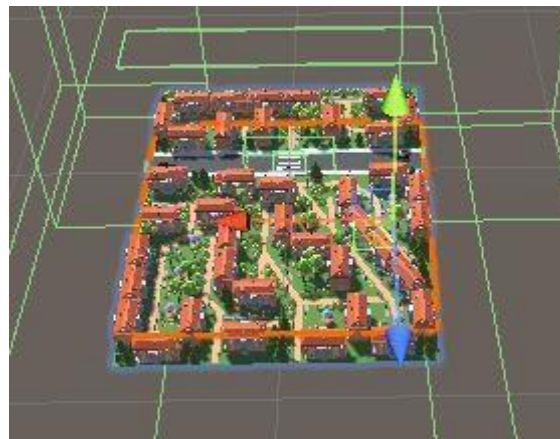


Рисунок 4.5 – Модель поверх нового маркера

Ми можемо використовувати FixedUpdate замість Update, (рисунок 4.6) що дозволить нам незалежно від частоти кадрів викликати функцію 50 разів на секунду [14], завдяки чому пропаде залежність швидкості об'єкта від частоти кадрів.

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

public class RigidbodySetVelocity : MonoBehaviour
{
    public float forceMult = 2;
    private Rigidbody rb;

    void Start()
    {
        rb = GetComponent<Rigidbody>();
    }

    void FixedUpdate()
    {
        rb.velocity = transform.forward * forceMult;
        rb.angularVelocity = transform.forward * 0;
    }
}
```

Рисунок 4.6 – Оновлений лістинг класу RigidbodySetVelocity

Зберігши проект і зібравши застосунок, відправимо його на мобільний пристрій (рисунок 4.7).

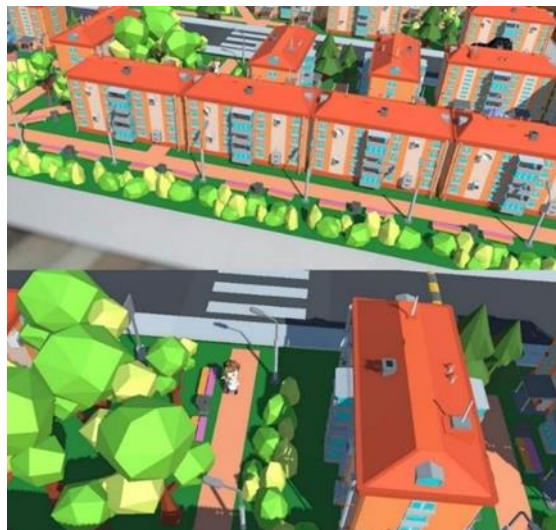


Рисунок 4.7 – Демонстрація відображення моделі на мітці

Також необхідно перевірити трекінг та накладення моделі на маркер під час наведення камери під кутом (рисунок 4.8).



Рисунок 4.8 – Модель на мітці під кутом

Як видно з рисунка 4.8, розширення області маркера до повного розміру сторінки суттєво покращило якість трекінгу. Завдяки цьому рішення система відстеження оперативно і стабільно визначає положення мітки у просторі. При наведенні камери мобільного пристрою на сторінку сцена завантажується майже миттєво, що свідчить про високу ефективність реалізованого підходу.

Крім того, покращено плавність трекінгу: модель фіксується на мітці без помітного зсуву, зберігаючи стабільне положення навіть при зміні кута огляду або частковому перекритті маркера. Це дозволяє користувачу взаємодіяти зі сценою у більш природний спосіб, забезпечуючи комфортну візуалізацію в умовах змінного освітлення та положення камери.

ВИСНОВКИ

У межах кваліфікаційної роботи було досліджено методи комп'ютерного зору та трекінгу, побудовано математичну модель доповненої реальності, а також реалізовано програмний модуль для її практичного застосування. На основі моделі створено набір маркерів, оптимізованих для надійного розпізнавання у системі доповненої реальності.

Результатом стала розробка кросплатформного мобільного застосунку на базі Unity з використанням бібліотеки Vuforia. Основною метою застосунку є візуалізація правил дорожнього руху для дітей у форматі інтерактивної AR-сцени.

Під час тестування виявлено низку недоліків, пов'язаних із точністю трекінгу та часом розпізнавання маркерів. Після доопрацювання структури маркерів було досягнуто значного покращення стабільності відображення 3D-моделей та загальної продуктивності застосунку. Отримані результати свідчать про ефективність запропонованого підходу до побудови AR-систем із освітньою метою.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Whyte J., Dragana N. *Virtual Reality and the Built Environment* / J. Whyte, N. Dragana. – CRC Press, 2018.
2. Бологова Н., Рубан І. Дослідження моделей та методів обробки зображень та шляхи вдосконалення технологій розпізнавання маркерів в системах доповненої реальності / Н. Бологова, І. Рубан // *Innovative Technologies and Scientific Solutions for Industries*. – 2019. – № 7. – DOI: 10.30837/2522-9818.2019.7.025.
3. Ahram T., Falcão C. *Advances in Usability and User Experience* / T. Ahram, C. Falcão. – 1-е вид. – Springer, 2019. – [Електронний ресурс]. – Режим доступу: <https://link.springer.com/book/10.1007/978-3-030-19135-1> (дата звернення: 10.05.2025).
4. Papagiannis H. *Augmented Human: How Technology Is Shaping the New Reality* / H. Papagiannis. – Sebastopol, CA : O'Reilly Media, Inc., 2017. – 156 с. – ISBN 978-1-4919-2832-5.
5. Сухіх А. Потенціал віртуальної та доповненої реальності для покращення змішаного навчання в закладах загальної середньої освіти / А. Сухіх // Інститут цифровізації освіти НАПН України. – [Електронний ресурс]. – Режим доступу: <http://digital-academy.org.ua> (дата звернення: 12.05.2025).
6. Кінг Б. *Augmented: життя в розумному світі* / Б. Кінг ; пер. з англ. – Сінгапур : Marshall Cavendish Business, 2016. – 320 с. – ISBN 978-981-4634-03-5.
7. Tyagi A. K. *Multimedia and Sensory Input for Augmented, Mixed, and Virtual Reality* / A. K. Tyagi. – 1-е вид. – IGI Global, 2021. – [Електронний ресурс]. – Режим доступу: <https://www.igi-global.com/book/multimedia-sensory-input-augmented-mixed/244546> (дата звернення: 19.05.2025).
8. Rukzio E., Park S., Böhmer M. *Mobile augmented reality: Design*

principles for mobile augmented reality applications / E. Rukzio, S. Park, M. Böhmer // Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services. – 2009. – P. 137.

9. Schmalstieg D., Hollerer T. Augmented reality: Principles and practice / D. Schmalstieg, T. Hollerer. – Addison-Wesley, 2016. – 576 c.

10. Specht M., Klemke R., Tochtermann K. Mobile learning: Challenges, content and technologies / M. Specht, R. Klemke, K. Tochtermann // International Journal of Technology Enhanced Learning. – 2011. – Vol. 3, № 6. – P. 474–488.

11. Vavoula G. N., Sharples M. Challenges in evaluating mobile learning: A 3-year case study / G. N. Vavoula, M. Sharples // Proceedings of the 4th IEEE International Workshop on Wireless and Mobile Technologies in Education. – 2006. – P. 195–199.

12. Wang F., Hannafin M. J. Design-based research and technology-enhanced learning environments / F. Wang, M. J. Hannafin // Educational Technology Research and Development. – 2005. – Vol. 53, № 4. – P. 5–23.

13. Wu H., Lee S., Chang H., Liang J. Current status, opportunities and challenges of augmented reality in education / H. Wu, S. Lee, H. Chang, J. Liang // Computers & Education. – 2013. – Vol. 62. – P. 41–49.

14. Preece J., Rogers Y., Sharp H. Interaction Design: Beyond Human-Computer Interaction / J. Preece, Y. Rogers, H. Sharp. – 5th ed. – Wiley, 2015. – 600 c.