

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет

імені Петра Могили

Факультет комп'ютерних наук

Кафедра комп'ютерної інженерії

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри,

д-р техн. наук, проф.

_____ І. М. Жаравська

« ____ » _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Програмно-апаратний комплекс навігації

у закладах охорони здоров'я

Спеціальність 123 Комп'ютерна інженерія

123 – КБР.ПЗ.00 – 405.21910516

Студент

_____ К. А. Куденко

підпис

« ____ » _____ 2023р.

Керівник канд. техн. наук, доцент

_____ А. П. Бойко

підпис

« ____ » _____ 2023р.

Миколаїв – 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра комп'ютерної інженерії

ЗАТВЕРДЖУЮ

Зав. кафедри _____ І. М. Журавська

« _____ » _____ 2023 р.

ЗАВДАННЯ
на виконання кваліфікаційної бакалаврської роботи

Видано студенту групи 405 факультету комп'ютерних наук

_____ Куденко Кирилу Андрійовичу
(прізвище, ім'я, по-батькові студента)

1. Тема кваліфікаційної роботи

_____ Програмно-апаратний комплекс навігації у закладах охорони здоров'я

Затверджена наказом по ЧНУ від « _____ » _____ 2023 р. № _____

2. Строк представлення кваліфікаційної роботи « _____ » _____ 20 _____ р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

_____ Комплекс навігації, що показує маршрут на всій території миколаївській обласній дитячій лікарні.

4. Перелік питань, що підлягають розробці _____ Дослідити існуючі аналоги. Дослідити середовища для 3D моделювання. Розробка 3D сцени. Підібрати елементи для програмно-апаратного комплексу. Створити програмне забезпечення. Спеціальна частина з охорони праці. Висновки. Перелік _____ джерел посилання.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної бакалаврської роботи

Тема: Програмно-апаратний комплекс навігації у закладах охорони здоров'я

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КР	11.12.2022	12.12.2022	Виконав
2.	Аналіз інформації стосовно загальної інформації про програмно-апаратні комплекси навігації	17.01.2023	19.02.2023	Виконав
3.	Порівняльний аналіз існуючих програмно-апаратних комплексів навігації у закладах охорони здоров'я	19.02.2023	04.03.2023	Виконав
4.	Обґрунтування вибору складових для навігації	05.04.2023	25.04.2023	Виконав
5.	Збір інформації	30.04.2023	07.04.2023	Виконав
6.	Вибір програмного забезпечення для створення 3Dмоделей	15.04.2023	20.04.2023	Виконав
7.	Вибір програмного забезпечення для створення програмного застосунку	23.04.2023	27.04.2023	Виконав
8.	Розробка 3D моделей	02.04.2023	23.04.2023	Виконав
9.	Розробка програмного забезпечення	29.04.2023	14.05.2023	Виконав
10.	Оформлення роботи	15.05.2023	12.06.2023	Виконав
11.	Підготовка презентації	11.06.2023	12.06.2023	Виконав
12.	Попередній захист	13.06.2023	13.06.2023	Виконав
13.	Рецензування	13.06.2023	20.06.2023	Виконав
14.	Завершення оформлення КР та презентації	13.06.2023	20.06.2023	Виконав
15.	Захист кваліфікаційної роботи	27.06.2023	28.06.2023	Виконав

Розробив здобувач ВО Куденко Кирил Андрійович _____

(прізвище, ім'я, по батькові)

(підпис)

«__» _____ 20__ р.

Керівник роботи канд. техн. наук, доцент Бойко А. П. _____

(підпис)

«__» _____ 20__ р.

АНОТАЦІЯ

до кваліфікаційної роботи

«Програмно-апаратний комплекс навігації у закладах охорони здоров'я»

Студент 405 гр.: Куденко Кирил Андрійович
Керівник: канд. техн. наук, доцент Бойко А. П.

Актуальність: на сьогоднішній день програмно-апаратного комплексу навігації є актуальними та важливими засобами в закладах охорони здоров'я. Вони допомагають забезпечити точність та ефективність навігації в медичних просторах, таких як лікарні, клініки, оперційні зали.

Мета: поліпшення руху по території лікарні для пацієнтів шляхом розробки програмно-апаратного комплексу для навігації у приміщеннях складних споруд.

Об'єкт дослідження: навігація у приміщеннях зі складною конструкцією.

Предмет дослідження: програмно-апаратний комплекс для реалізації навігації у закладах охорони здоров'я.

Робота складається з фахового розділу і спеціальної частини з охорони праці. Пояснювальна записка складається зі вступу, чотирьох розділів, висновків та додатків.

У першому розділі проаналізована предметна область дослідження: види навігацій та їх практична реалізація, існуючі програмні середовища комп'ютерного проектування.

В другому розділі було розглянуто, проаналізовано та поетапно відображено складний процес створення комп'ютерної 3D моделі миколаївської обласної дитячої клінічної лікарні.

У третьому розділі надано опис технологій, що використовувалися при створенні програмної частини навігаційного сервісу. У якості рушія для створення анімації використовувалося програмне середовище Unity. Для написання всіх функцій, елементів управління, відстеження та обробки в роботі використовувалася мова програмування C#.

У четвертому розділі описані системні вимоги та порядок розгортання розробленого програмного продукту. Розглянуто сценарій роботи з мобільним додатком, порядок користування системою, її основні функції.

Результатом роботи став програмно-апаратний комплекс навігації у закладах охорони здоров'я.

Бакалаврська кваліфікаційна робота викладена на 59 сторінок, містить 4 розділи, 39 ілюстрацій, 4 таблиці та 24 джерел у переліку посилань.

Ключові слова: *програмно-апаратний комплекс, навігація, 3D моделювання, заклад охорони здоров'я, програмний застосунок.*

ABSTRACT

of the Bachelor's Thesis

«Software and hardware complex of navigation in healthcare institutions»

Student: Kudenko Kiril Andreyovych

Supervisor: Candidate of technical sciences, Docent Boyko A. P.

Relevance: today software and hardware navigation complexes are relevant and important tools in health care institutions. They help ensure the accuracy and efficiency of navigation in medical spaces such as hospitals, clinics, and operating rooms.

Purpose: improvement of movement around the hospital territory for patients through the development of a software and hardware complex for navigation in the premises of complex buildings.

Research object: navigation in rooms with a complex structure.

Subject of research: software and hardware complex for implementation of navigation in health care institutions.

The work consists of a professional section and a special part on labor protection. The explanatory note consists of an introduction, four chapters, conclusions and appendices.

The first chapter analyzes the subject area of research: types of navigation and their practical implementation, existing computer design software environments.

In the second chapter, the complex process of creating a computer 3D model of the Mykolaiv Regional Children's Clinical Hospital was considered, analyzed, and step-by-step displayed.

The third section provides a description of the technologies used in the creation of the software part of the navigation service. The Unity software environment was used as the engine for creating the animation. The C# programming language was used to write all functions, control elements, tracking and processing in the work.

The fourth chapter describes the system requirements and the procedure for deploying the developed software product. The scenario of working with the mobile

application, the procedure for using the system, and its main functions are considered.

The result of the work was a hardware and software complex for navigation in healthcare facilities.

The bachelor's thesis is laid out on 59 pages, contains 4 chapters, 39 illustrations, 4 tables and 24 sources in the list of references.

Keywords: *hardware and software complex, navigation, 3D modeling, health care facility, software application*

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	3
ВСТУП	4
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	6
1.1 Поняття та використання 2D та 3D навігації	6
1.2 Огляд сучасного програмного забезпечення для розробки 3D графіки. 16	
Висновки до розділу 1	23
2 РОЗРОБКА ТЕХНОЛОГІЇ 3D МОДЕЛЮВАННЯ ТА АПАРАТНОЇ ЧАСТИНИ НАВІГАЦІЇ.....	24
2.1 Проектування 3D моделі комплексу у закладах охорони здоров'я	24
2.2 Використання застосунку Blender.....	38
2.3 Підбір та опис апаратної частини сервісу навігації	40
Висновки до розділу 2	44
3 РОЗРОБКА ПРОГРАМНОГО ЗАСТОСУНКУ.....	45
3.1 Компіляція програми	45
3.2 Реалізація серверу	52
Висновки до розділу 3	55
4 РОБОТА КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ.....	56
4.1 Системні вимоги та інсталяція	56
4.2 Сценарій роботи користувача з системою	57
Висновки до розділу 4	59
ВИСНОВОК.....	60
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	61
ДОДАТОК А Довідка	64
ДОДАТОК Б Код застосунку.....	65

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

3DS	-	результат генерації тривимірного графічного редактора
AR	-	augmented reality
CAD	-	computer-aided design
FBX	-	технологія та формат файлів
GPS	-	global positioning system
GUI	-	graphical user interface
HCI	-	human-computer interaction
IDE	-	integrated development environment
OBJ	-	формат файлів опису геометрії
UI	-	user interface
VR	-	virtual reality

ВСТУП

Програмно-апаратний комплекс навігації стає все більш популярним у закладах охорони здоров'я. Його основною метою є полегшення навігації по медичних закладах для пацієнтів та працівників, що дозволяє ефективніше використовувати час і засоби, знижувати ризик помилок та покращувати якість медичних послуг.

Програмно-апаратний комплекс навігації може значно полегшити організацію медичного процесу в закладах охорони здоров'я. Основною функцією такого комплексу є надання точної та швидкої інформації щодо місцезнаходження кабінетів лікарів та адміністрації, що дозволяє ефективніше розподіляти ресурси та забезпечувати оптимальний режим роботи медичного закладу.

Для пацієнтів програмно-апаратний комплекс навігації може бути корисним, оскільки він дозволяє швидко та легко знайти необхідний кабінет, лікаря чи відділення в закладі охорони здоров'я. Це допомагає зменшити стрес та дискомфорт у пацієнтів та забезпечує більш точну та своєчасну надання медичних послуг.

Загалом, програмно-апаратний комплекс навігації є важливим інструментом для поліпшення організації медичного процесу в закладах охорони здоров'я. Він допомагає знижувати ризик помилок, забезпечувати ефективне використання ресурсів та покращувати якість медичних послуг для пацієнтів.

Для досягнення поставленої **мети** необхідно виконати такі завдання:

- аналізувати наявні сервіси для навігації у внутрішніх приміщеннях;
- дослідити сучасні технології тривимірного моделювання;
- вивчити актуальні засоби розробки додатків з тривимірною графікою;
- підібрати компоненти для розробки комплексу;

– розробити програмне забезпечення для реалізації внутрішньої навігації.

Об'єкт: навігація у внутрішніх приміщеннях зі складною структурою.

Предмет: комплекс апаратно-програмних засобів для навігації у медичних закладах.

Практичне значення отриманих результатів полягає в полегшенні зовнішньої та внутрішньої навігації, плануванні будівельних та ремонтних робіт в медичних комплексах. Запропонована модель також може бути використана для розробки інтерактивного 3D-застосунку, який дозволить візуалізувати, навігувати та взаємодіяти з 3D-моделлю. Цей застосунок дозволить використовувати віртуальний тур, що подібний до комп'ютерної гри, для переміщення по тривимірній моделі. З високою ступенем реалістичності 3D-моделі, користувач отримає ефект зануреності в представлений об'єкт. Застосунок вирішує проблему позиціонування в просторі, навігації до бажаного об'єкту та отримання інформації про нього.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття та використання 2D та 3D навігації

1.1.1 Розвиток 2D навігації

Історія 2D навігації пов'язана з розвитком картографії, яка почалася ще в давнину. Перші карти були створені для зображення землі та морських просторів. У середньовічній Європі картографія стала популярним видом мистецтва, але картографи не мали можливості вимірювати точні координати місцевості. Тому їх карти були неповними та недостатньо точними [4].

У XVIII столітті відбувся значний розвиток науки та технології, що дозволило створити більш точні карти. Було винайдено криву Меркатора (рис. 1.1), яка дозволила передавати географічну інформацію відповідно до відстаней на поверхні Землі. Виникли перші системи координат та картографічні проєкції, які забезпечували точність при зображенні місцевості [14].

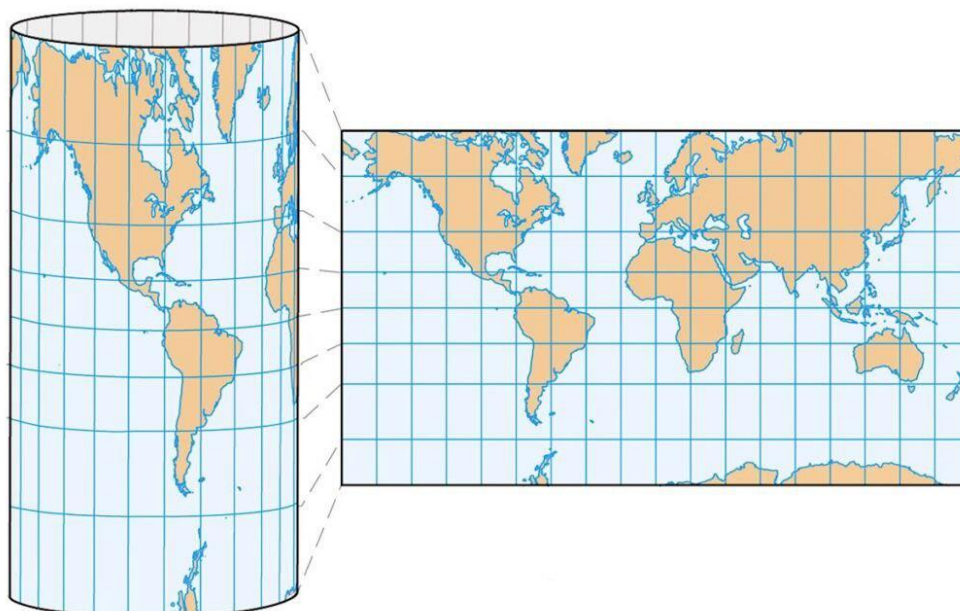


Рисунок 1.1 – Проекція Меркатора.

З розвитком технологій використання карт здійснювалося все швидше та з точністю, що забезпечувало їх використання в навігації. У XIX столітті

виникли перші карти масштабу 1:50000, які дозволяли здійснювати точну навігацію на території певної країни або регіону. З початку ХХ століття з'явилися аерофотозйомки та супутникова зйомка з повітря (рис. 1.2), що дозволило створити детальніші та точніші карти [8].



Рисунок 1.2 – Аерофотозйомка 20 століття

Сьогодні з 2D картами можна зустріти в різних форматах – від традиційних паперових карт до цифрових карт в різних мобільних додатках та на сайтах. Карти стали необхідним інструментом для багатьох людей, які займаються подорожами, навігацією транспортних засобів та іншими сферами, пов'язаними з переміщенням.

З появою автомобілів у 20 столітті, картографія почала змінюватися для відповіді на потреби автомобілістів. Були створені спеціальні дорожні карти та атласи, які містили інформацію про дорожню мережу, відстані та час подорожі між містами.

У другій половині 20 століття з появою супутникової навігації, картографія стала ще більш точною та детальною. GPS-технології дозволили створити карту з точністю до кількох метрів, що значно полегшило навігацію транспортними засобами.

Картографія продовжує розвиватися, із зростанням популярності мобільних технологій та інтернету, доступність карти стала ще більшою. Мобільні додатки та онлайн сервіси надають можливість швидко та зручно знаходити шлях до потрібного місця, оцінювати час та відстань подорожі, а також отримувати іншу корисну інформацію про маршрут. З появою смартфонів та планшетів, мобільні додатки стали дуже популярними серед користувачів, які шукають зручний та швидкий спосіб навігації.

Перші мобільні додатки для навігації були розроблені в 90-х роках. Тоді вони були дуже простими та мали обмежений функціонал. Проте з появою нових технологій та високошвидкісних мереж, мобільні додатки для навігації стали більш розширеними та функціональними.

Сьогодні на ринку доступно безліч мобільних додатків та онлайн сервісів для навігації. Найпопулярніші з них – Google Maps, Waze, Apple Maps, MapQuest, Here WeGo та інші (рис. 1.3). Вони надають користувачам детальну інформацію про маршрут, дозволяють планувати подорожі з урахуванням трафіку, оцінювати відстань та час подорожі, знаходити найближчі ресторани, готелі, магазини та інші місця.

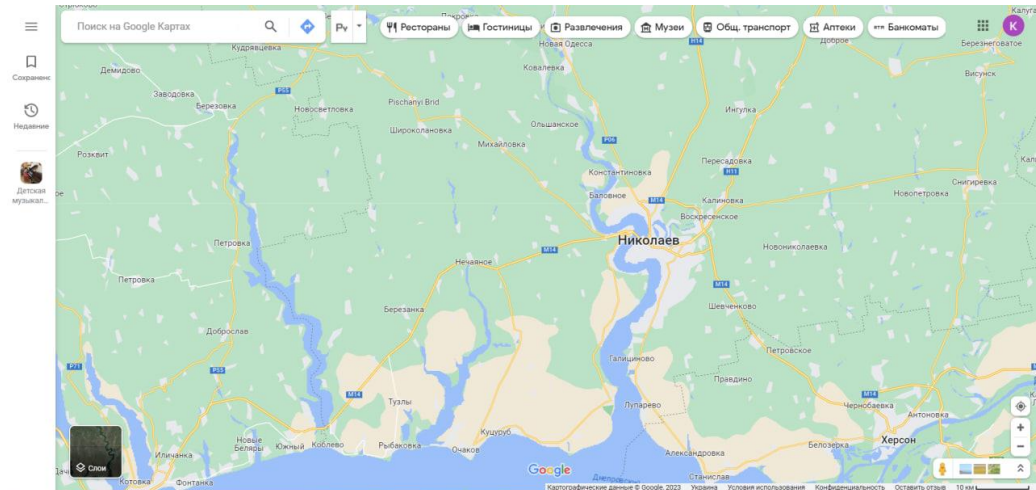


Рисунок 1.3 – Приклад онлайн сервісу для навігації

Також, деякі мобільні додатки та сервіси дозволяють користувачам створювати свої власні маршрути, додавати відомості про пам'ятки та цікаві місця, а також ділитися своїм місцезнаходженням з іншими користувачами [6].

У цілому, мобільні додатки та онлайн сервіси для навігації стали незамінними інструментами для багатьох людей, які подорожують або просто шукають швидкий та зручний спосіб знайти потрібне місце.

1.1.2 Технології візуальної навігації

Навігація у внутрішньому просторі будівель може бути важкою задачею, особливо якщо мова йде про великі торгові центри, аеропорти, музеї та інші складні структури [11]. Технології візуальної навігації можуть допомогти користувачам орієнтуватися в середині будівель та знайти потрібне місце. Ось деякі з таких технологій:

- QR-коди: QR-коди можуть бути розміщені по всій будівлі і містити інформацію про місцезнаходження, напрямок руху, інформацію про послуги, які надаються та інше. Користувач може відсканувати QR-код за допомогою смартфона або планшета, щоб отримати необхідну інформацію та знайти потрібне місце;

- Bluetooth-бейкони: Bluetooth-бейкони – це пристрої, які можуть бути розміщені по всій будівлі та відправляти сигнали до мобільного пристрою

користувача. Ці сигнали допомагають визначити місцезнаходження користувача та надати напрямок до потрібного місця;

– AR-технології: AR-технології використовують камеру смартфона або планшета, щоб відобразити віртуальні об'єкти на екрані, які допомагають користувачеві орієнтуватися в будівлі та знайти потрібне місце;

– електронні карти: Електронні карти можуть бути збережені на смартфоні або планшеті та містити інформацію про будівлю, включаючи місцезнаходження, напрямок руху, поверх та іншу корисну інформацію;

– 3D-моделі: 3D-моделі будівлі можуть бути створені за допомогою спеціального програмного забезпечення, які дозволяють користувачеві досліджувати будівлю з будь-якого кута і переглядати детальну інформацію про місцезнаходження, поверхи та інші атрибути будівлі. Ці 3D-моделі можуть бути використані для створення візуальних екскурсій та інтерактивних турів для відвідувачів, що дозволяє їм отримати більш детальну інформацію про будівлю та її місцезнаходження.

Ці технології можуть бути використані окремо або в комбінації, щоб допомогти користувачам орієнтуватися в будівлі та знайти потрібне місце. Вони дозволяють зробити навігацію більш ефективною та приємною для користувачів та допомагають зменшити кількість запитів до персоналу.

1.1.3 Розвиток 3D навігації

Історія 3D навігації простежується з появою комп'ютерної графіки та розвитком комп'ютерних систем. У 1960-х роках почалася робота над розробкою комп'ютерних систем із візуалізацією 3D об'єктів.

Перші системи, такі як Sketchpad, дозволяли користувачам створювати та маніпулювати об'єктами у 3D просторі. Sketchpad – це комп'ютерна програма, написана Іваном Сазерлендом у 1963 році під час його докторської дисертації, за яку він отримав премію Тюрінга в 1988 році та премію Кіото в 2012 році (рис. 1.4) [9]. Вона стала піонером у спілкуванні людино-

комп'ютерна взаємодія (HCI) і вважається родоначальником сучасних програм автоматизованого проектування (CAD), а також великим проривом у розвитку комп'ютерної графіки загалом. Наприклад, графічний інтерфейс користувача(GUI) був похідним від Sketchpad, а також сучасного об'єктно-орієнтованого програмування. Використовуючи програму, Іван Сазерленд показав, що комп'ютерну графіку можна використовувати як для художніх, так і для технічних цілей на додаток до демонстрації нового методу взаємодії людини з комп'ютером.



Рисунок 1.4 – Sketchpad і Іван Сазерленд

В 1970-х і 1980-х роках з'явилися перші графічні інтерфейси, які дозволяли користувачам взаємодіяти з 3D об'єктами за допомогою миші та клавіатури. Розроблено такі системи, як Xerox Alto, які вперше дозволили користувачам переглядати та рухати об'єкти у 3D просторі.

В 1980-х роках з'явилися перші програми для 3D моделювання, які дозволяли створювати складні 3D об'єкти та моделі. Програми, такі як AutoCAD та 3D Studio, дозволили дизайнерам та художникам створювати реалістичні 3D сцени та об'єкти.

У 1990-х роках із з'явленням потужних графічних карт та 3D-прискорювачів, комп'ютерні ігри стали використовувати 3D графіку для

створення реалістичних віртуальних світів. Разом з цим з'явилися нові методики та алгоритми для обробки та рендерингу 3D сцен. В останні роки віртуальна реальність (VR) та доповнена реальність (AR) стали важливими напрямками розвитку 3D навігації. VR дозволяє користувачам погрузитись у повністю віртуальне середовище, де вони можуть взаємодіяти з 3D об'єктами та простором. AR поєднує віртуальні об'єкти з реальним світом, дозволяючи користувачам взаємодіяти з 3D об'єктами у реальному оточенні [1].

З появою смартфонів та планшетів з потужними графічними можливостями, розвиток мобільної 3D навігації став значимим. Користувачі можуть використовувати мобільні пристрої для взаємодії з 3D об'єктами та рухатись у віртуальних просторах. Розвиток сенсорних технологій, таких як жести, голосові команди та відстеження руху, вплинував на розвиток 3D навігації. Користувачі можуть використовувати свої рухи, жести та голос для взаємодії з 3D об'єктами та простором.

З розвитком хмарних технологій, 3D навігація стала доступною через веб-браузери та інтернет. Користувачі можуть віддалено отримувати та взаємодіяти з 3D контентом без необхідності встановлення спеціального програмного забезпечення. Застосування машинного навчання та штучного інтелекту в 3D навігації дозволяє створювати більш інтелектуальні та реалістичні системи навігації. Машинне навчання може використовуватися для автоматичного розпізнавання об'єктів та середовищ, а також для прогнозування руху та поведінки об'єктів у 3D просторі. Штучний інтелект дозволяє створювати розумних агентів, які можуть приймати рішення, планувати маршрути та взаємодіяти з користувачем у 3D середовищі.

3D навігація стає все більш популярною та використовується в різних галузях, включаючи ігрову індустрію, віртуальний туризм, медицину, архітектуру, автомобільну промисловість та багато інших. Вона надає нові можливості для візуалізації та взаємодії зі складними 3D даними та моделями.

Загальна тенденція розвитку 3D навігації полягає в поєднанні реалістичності, інтерактивності та інтелекту для створення більш високоякісного та іммерсивного досвіду для користувачів. Технологічний прогрес, поширення мобільних пристроїв, розширення сенсорних технологій та застосування штучного інтелекту впливають на постійне вдосконалення 3D навігаційних систем та їх використання у різних сферах життя.

1.1.4 Навігація у приміщенні

Навігація у приміщенні – це процес руху і орієнтування людини у внутрішньому просторі будівлі або іншого закритого приміщення, щоб знайти потрібне місце або вихід [5].

Основні методи навігації в приміщенні включають в себе:

- використання плану будівлі: це може бути фізичний план, який надрукований на папері, або електронна карта, яка зберігається на смартфоні або планшеті;
- орієнтування за зовнішніми орієнтирами: цей метод передбачає використання зовнішніх орієнтирів, таких як прилеглі будівлі, знаки на вулиці або рекламні конструкції, щоб знайти потрібне місце;
- використання внутрішніх орієнтирів: цей метод передбачає використання внутрішніх орієнтирів, таких як сходи, ліфти, кам'яні стовпи або двері, щоб знайти потрібне місце;
- використання електронних систем навігації: деякі будівлі можуть мати електронні системи навігації, які використовують технологію GPS або Bluetooth для допомоги в орієнтуванні у приміщенні.

Важливою складовою успішної навігації у приміщенні є збереження спокою і не панікувати, коли ви здаєтеся загубленим. Якщо ви не можете знайти потрібне місце або вихід, зверніться до персоналу будівлі або використайте мобільний телефон, щоб знайти додаткову допомогу.

1.1.5 Конкурентний робочий аналог

На зараз найрозвинутішою сучасною реалізацією навігації в будівлях, в тому і числі у закладах охорони здоров'я, продукт французької компанії Visioglobe [15]. Visioglobe використовує технології комп'ютерного зору та машинного навчання для створення детальних 3D-моделей приміщень. Компанія працює з клієнтами з різних галузей, включаючи торгівлю, транспорт, музеї, готелі, і на закладах охорони здоров'я. Компанія пропонує своє власне рішення, яке містить в собі компоненти візуальної навігації, маршрутизації та відстеження локації.

Одним з ключових продуктів компанії є VisioMapEditor – інструмент для створення 2D та 3D карт будівель, включаючи плани поверхів, розміщення кімнат та обладнання, маршрути та інші деталі. VisioMapEditor має вбудований редактор 3D-моделей, що дозволяє користувачам створювати віртуальні тури будівлі, включаючи внутрішні приміщення. Користувачі можуть легко додавати об'єкти, такі як двері, вікна та сходи, і встановлювати точки цікавості на карті для додаткової інформації. VisioMapEditor також дозволяє додавати додаткову інформацію, таку як графіки роботи лікарень, медичні ресурси та інші дані.

Крім того, компанія розробляє мобільні додатки для візуальної навігації, які дозволяють пацієнтам та відвідувачам швидко знаходити потрібні кімнати, кабінети, лікарів та інші об'єкти в лікарні.

Можливості сервісу:

- створення 2D- та 3D-моделей: Візуалізація будівель та об'єктів може бути виконана у форматі 2D або 3D;
- керування шаром: VisioMapEditor дозволяє створювати та редагувати окремі шари для різних елементів, наприклад, ліфтів, сходів, дверей тощо;

- додавання міток та іконок: В сервісі є можливість додавати різні мітки та іконки для позначення різних об'єктів, наприклад, кабінетів лікарів, туалетів, зон відпочинку та інших;
 - відстеження маршрутів: VisioMapEditor дозволяє визначати маршрути з однієї точки до іншої та відображати їх на навігаційній схемі;
 - перегляд в режимі реального часу: Завдяки інтеграції з іншими системами, VisioMapEditor дозволяє відслідковувати рух людей та обладнання у режимі реального часу на навігаційній схемі;
 - можливість інтеграції з іншими системами: VisioMapEditor може бути інтегрований з різними системами управління будівлями, такими як системи безпеки, контролю доступу, системи управління енергоспоживанням та інші;
 - аналітика та статистика: Сервіс надає можливість збирати статистику щодо використання навігаційної схеми та аналізувати дані для оптимізації роботи закладу охорони здоров'я;
 - віддалений доступ: завдяки хмарній технології.
- Головні проблеми аналогу:
- складність використання: VisioMapEditor може бути складним у використанні для користувачів, які не мають досвіду з роботою з подібними програмами. Для того, щоб створювати відповідні карти, потрібно мати знання в області графіки, 3D-моделювання, редагування та відображення даних;
 - обмеження щодо джерел даних: VisioMapEditor може мати обмеження щодо джерел даних, з яких можна отримати інформацію для створення карт. Це може обмежувати точність та деталізацію карт;
 - обмеження щодо доступності: VisioMapEditor може бути обмеженим у доступності для багатьох користувачів через високу вартість або обмежену кількість ліцензій;

- нестабільність: Іноді VisioMapEditor може бути нестабільним та працювати повільно на старіших комп'ютерах або на тих, що мають обмежені ресурси;
- обмеження щодо інтеграції: VisioMapEditor може мати обмеження щодо інтеграції з іншими програмами, що може ускладнити створення програмно-апаратних комплексів навігації в будівлях.

1.2 Огляд сучасного програмного забезпечення для розробки 3D графіки

1.2.1 ПЗ для створення 3D моделей

3D-моделювання – це процес створення тривимірної реконструкції об'єкта. Завданням 3D-моделювання є створення візуального образу об'єкта з різних ракурсів.

Категорію "3D" складають програми для створення і редагування тривимірних моделей. Сюди входять як професійні пакети для моделювання, анімації і обробки 3D, так і програми для початківців. За допомогою цих програм можна створити 3D-моделі різного рівня складності. Також в цю категорію входять спеціалізовані інструменти, які використовуються для конкретних цілей.

Особливістю та новаторством запропонованого підходу є використання різноманітних програмних середовищ для досягнення більш точного та реалістичного зображення створеної моделі.

Для створення 3D об'єктів існує велика кількість програм, починаючи з простих інструментів і закінчуючи професійними пакетами розробки, такими як Autodesk Maya, AutoCAD, 3Ds Max, Blender, SketchUp, Wings 3D та інші. Проте, лише обмежена кількість програм підходить для виконання даної роботи.

AutoDesk 3Ds Max (або 3D Studio Max) є програмою для моделювання, анімації та рендерингу 3D-об'єктів, розроблена компанією Autodesk (Рис 4).

Це один з найпопулярніших інструментів у галузі 3D-графіки, який використовується для створення ігор, архітектурних візуалізацій, фільмів та інших проектів [12].

3Ds Max має ряд функцій для моделювання, включаючи інструменти для створення форм, редагування мешів та текстур, анімації та взаємодії зі світлом та тінями. Крім того, програма має різноманітні інструменти для рендерингу, що дозволяють створювати реалістичні зображення та анімацію.

3Ds Max має велику спільноту користувачів та підтримує різні формати файлів, включаючи OBJ, FBX, 3DS та багато інших. Програма також підтримує інтеграцію з іншими програмами, такими як Photoshop та After Effects, що дозволяє більш ефективно працювати з проектами.

Усі ці функції роблять 3Ds Max потужним інструментом для розробки 3D-графіки, що дозволяє користувачам створювати складні 3D-моделі, анімації та реалістичні візуалізації проектів (рис. 1.5).

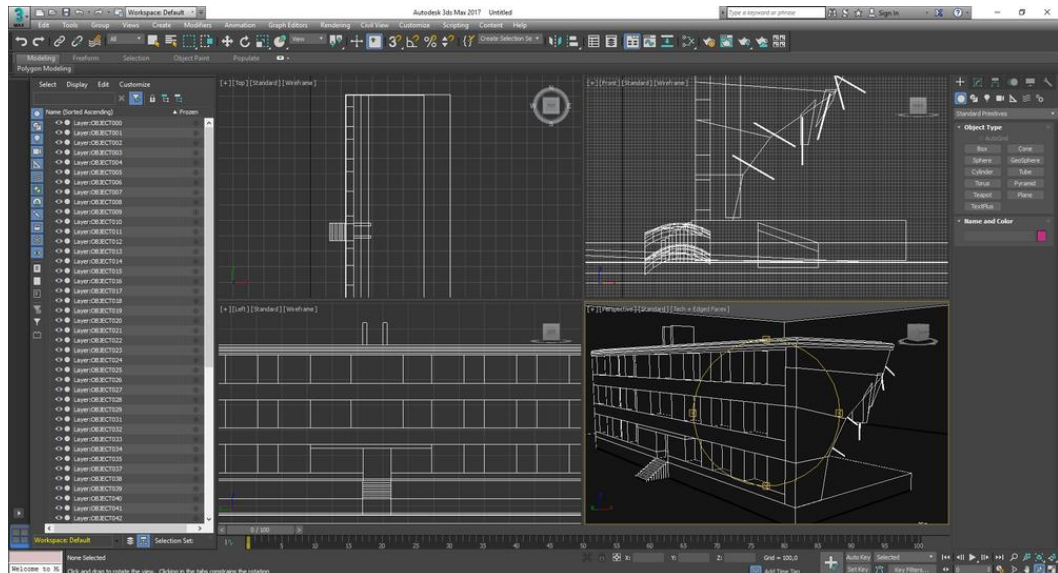


Рисунок 1.5 – 3Ds Max

3Ds Max має у своєму розпорядженні значні засоби для створення різноманітних за формою і складністю тривимірних комп'ютерних моделей, реальних чи фантастичних об'єктів навколишнього світу, з використанням різноманітних технік і механізмів, що включають наступні:

- полігональне моделювання, в яке входять Editable mesh (редагована поверхня) і Editable poly (редагований полігон) – це найпоширеніший метод моделювання, використовується для створення складних моделей і низькополігональних моделей для ігор;
- моделювання на основі неоднорідних раціональних В-сплайнів (NURBS). Технологія побудови тривимірних моделей за допомогою неоднорідних раціональних В-сплайнів (особливого виду кривих ліній) дозволяє створювати моделі плавних форм, які не мають гострих країв;
- моделювання на основі так званих «Сіток шматків» або поверхонь Безьє (Editable patch) – використовується для моделювання тіл обертання;
- моделювання з використанням вбудованих бібліотек стандартних параметричних об'єктів (примітивів) і модифікаторів;
- моделювання на основі сплайнів (Spline) з подальшим застосуванням модифікатора Surface – примітивний аналог NURBS, зручний для створення об'єктів зі складними перетікаючими формами, які важко створити методами полігонального моделювання;
- моделювання на основі сплайнів з подальшим застосуванням модифікаторів Extrude, Lathe, Bevel Profile або створення на основі сплайнів об'єктів Loft. Цей метод широко застосовується для архітектурного моделювання. Методи моделювання можуть поєднуватися один з одним.

Моделювання сцени (віртуального простору моделювання) включає в себе кілька категорій об'єктів:

- геометрія (побудована за допомогою різних технік);
- матеріали (інформація про візуальні властивості моделі);
- джерела світла (настройки напрямку, потужності, спектра освітлення);
- віртуальні камери (вибір точки і кута побудови проекції).

1.2.2 ПЗ для створення програмного застосунку

Редактор Unity має зручний інтерфейс Drag & Drop, який легко налаштовується, складаючись з різних панелей, що дозволяють виконувати налагодження гри прямо в редакторі. Для скриптування використовуються дві мови: C# і JavaScript (модифікований варіант). Раніше підтримувався також діалект Python під назвою Boo, але він був видалений у 5-й версії. Фізичний рух моделей відповідає фізичному движку PhysX від NVIDIA [19].

Проект в Unity поділяється на сцени (рівні) – окремі файли, що містять свої власні ігрові світи з набором об'єктів, скриптів та налаштувань. Сцени можуть містити як об'єкти (моделі), так і порожні ігрові об'єкти – об'єкти, які не мають моделі (так звані "пустоти"). Об'єкти включають набір компонентів, з якими взаємодіють скрипти. Кожен об'єкт має свою назву (в Unity допускається наявність декількох об'єктів з однаковими назвами), може мати тег (мітку) і шар, на якому він має бути відображений. Так, у кожного об'єкта на сцені обов'язково присутній компонент Transform, який зберігає координати розташування, обертання і розміру об'єкта в тривимірному просторі. У об'єктів, що відображають геометрію, також за замовчуванням присутній компонент Mesh Renderer, який робить модель об'єкта видимою.

Об'єкти можуть мати колайдери (в Unity вони називаються колайдерами), які відповідають за обробку колізій.

Unity також підтримує фізику твердих тіл, м'яких тіл а також фізику типу «Ragdoll» (тряпична лялька). В редакторі Unity є система успадкування об'єктів, де дочірні об'єкти автоматично повторюють зміни позиції, обертання і масштабу батьківського об'єкта. Скрипти в редакторі можна прикріплювати до об'єктів у вигляді окремих компонентів.

Під час імпорту текстури в Unity можна генерувати альфа-канал, MIP-рівні, normal-мапу, light-мапу, карту відображень. Проте безпосередньо прикріпити текстуру до моделі не можна – замість цього створюється матеріал, до якого застосовується шейдер, а потім матеріал прикріплюється до моделі.

Редактор Unity підтримує написання і редагування шейдерів, що дозволяє вам контролювати візуальний вигляд об'єктів у грі.

Редактор Unity має простий Drag & Drop інтерфейс, який легко налаштовувати, що складається з різних вікон, завдяки чому можна проводити налагодження гри прямо в редакторі. Движок підтримує два скриптових мови: C #, JavaScript (модифікація). Раніше була підтримка Boo (діалект Python), але його прибрали в 5-й версії. Розрахунки фізики виробляє фізичний движок PhysX від NVIDIA [19].

Проект в Unity ділиться на сцени (рівні) – окремі файли, що містять свої ігрові світи зі своїм набором об'єктів, сценаріїв, і налаштувань. Сцени можуть містити в собі як, власне, об'єкти (моделі), так і порожні ігрові об'єкти – об'єкти, які не мають моделі («пустушки»). Об'єкти, в свою чергу містять набори компонентів, з якими і взаємодіють скрипти. Також у об'єктів є назва (в Unity допускається наявність двох і більше об'єктів з однаковими назвами), може бути тег (мітка) і шар, на якому він повинен відображатися. Так, у будь-якого об'єкта на сцені обов'язково присутній компонент Transform – він зберігає в собі координати місця розташування, повороту і розмірів об'єкта по всіх трьох осях. У об'єктів з видимою геометрією також за замовчуванням присутній компонент Mesh Renderer, що робить модель об'єкта видимою.

Також Unity підтримує фізику твердих тіл і тканини, а також фізику типу Ragdoll (тряпичная лялька). У редакторі є система успадкування об'єктів; дочірні об'єкти будуть повторювати всі зміни позиції, повороту і масштабу батьківського об'єкта. Скрипти в редакторі прикріплюються до об'єктів у вигляді окремих компонентів [2].

При імпорті текстури в Unity можна згенерувати alpha-канал, mip-рівні, normal-map, light-map, карту відображень, проте безпосередньо на модель текстуру прикріпити не можна – буде створено матеріал, з яким буде призначений шейдер, і потім матеріал прикріпиться до моделі. Редактор Unity підтримує написання і редагування шейдерів.

Unity має модуль для створення анімацій, але також можна попередньо створити анімацію у 3D-редакторі і імпортувати її разом з моделлю, а потім розбити на файли. Unity 3D підтримує систему рівня деталізації (LOD), що дозволяє замінювати високодеталізовані моделі на менш деталізовані при великій відстані від гравця та навпаки. Також в ньому присутня система Occlusion culling, яка приховує геометрію та колізію об'єктів, які не знаходяться в полі зору камери, зменшуючи навантаження на центральний процесор і покращуючи продуктивність проекту.

Під час компіляції проекту в Unity створюється виконуваний файл (.exe) гри для Windows, а також окрема папка зі всіма даними гри, включаючи всі рівні та спільні бібліотеки. Unity підтримує безліч популярних форматів файлів. Моделі, звуки, текстурі, матеріали та скрипти можна запакувати у формат .unityassets і передавати іншим розробникам або розміщувати у вільному доступі. Цей же формат використовується у внутрішньому магазині Unity Asset Store, де розробники можуть публікувати різні елементи, необхідні для створення ігор, як безкоштовно, так і за певну плату. Для використання Unity Asset Store потрібно мати обліковий запис розробника Unity.

Unity має вбудовані компоненти для створення мультиплеєрних ігор. Також можна використовувати засоби контролю версій, такі як Tortoise SVN або Source Gear. В комплект Unity входить Unity Asset Server – інструментарій для спільної розробки на базі Unity, який надає можливість керувати версіями та іншими серверними рішеннями. Unity Asset Server допомагає зберігати й керувати версіями ресурсів, спільною роботою розробників і відстеженням змін у проекті.

Крім того, Unity надає можливості для розробки ігор з мультиплеєрним режимом. Розробники можуть використовувати спеціальні компоненти та функції Unity для створення мережевого взаємодії між гравцями, синхронізації даних і обміну повідомленнями.

У процесі розробки ігор в Unity розробники також можуть скористатися різними засобами контролю версій, такими як Tortoise SVN або Source Gear. Ці інструменти дозволяють зберігати і відстежувати зміни в коді, сценах, ресурсах і спільно працювати над проектом з іншими розробниками.

Окрім цього, Unity Asset Store є централізованою платформою, де розробники можуть обмінюватися і продавати свої ресурси, такі як моделі, текстури, звуки, скрипти та інші елементи. Це надає зручний спосіб доступу до готових ресурсів та покращує швидкість розробки, оскільки розробники можуть використовувати вже існуючі ресурси замість того, щоб створювати їх з нуля.

Unity пропонує розширені інструменти для спільної розробки, контролю версій, створення анімацій та імпорту ресурсів, що допомагають розробникам створювати високоякісні ігри з оптимізованою продуктивністю та зручним інтерфейсом розробки.

Висновки до розділу 1

В даному розділі проведено аналіз предметної області дослідження, зокрема розглянуто різновиди систем навігації та їх практичну реалізацію, а також вивчено існуючі програмні засоби для комп'ютерного проектування.

Були розглянуті аналогічні продукти від комерційних компаній, які користуються високою популярністю на ринку. Були проаналізовані їх особливості, характеристики та недоліки, з метою надання власному програмному забезпеченню певних переваг і зроблення його конкурентоздатним у порівнянні з ними. Серед недоліків аналогових продуктів, які потрібно усунути, можна відзначити наступні:

- Висока вартість створення;
- Відсутність 3D виду та обмежений функціонал інтерфейсу для взаємодії з об'єктами;
- Неможливість інтеграції у більш складні проекти.

В результаті аналізу програмного забезпечення, що використовується для створення складних тривимірних моделей, було прийнято рішення використовувати не одне програмне середовище, а набір програмних засобів, таких як Autodesk 3Ds Max, Blender та Adobe Photoshop, при розробці 3D моделей комплексу лікарні. Це дозволить забезпечити більший функціонал, розширені можливості взаємодії з об'єктами і більш гнучкий підхід до процесу розробки.

2 РОЗРОБКА ТЕХНОЛОГІЇ 3D МОДЕЛЮВАННЯ ТА АПАРАТНОЇ ЧАСТИНИ НАВІГАЦІЇ

2.1 Проектування 3D моделі комплексу у закладах охорони здоров'я

2.1.1 Об'єкт проектування

Дитяча лікарня, що складається з 20 будівель, була заснована в 1972 році в інгульському районі міста Миколаїв. Вона спеціалізується на наданні медичної допомоги дітям, які потребують лікування і догляду.

Лікарня надає широкий спектр медичних послуг для дітей різного віку, включаючи діагностику, лікування та реабілітацію різних захворювань. Вона обладнана сучасними медичними технологіями та має висококваліфікований медичний персонал, який працює на благо здоров'я дітей [3].

Миколаївська обласна дитяча клінічна лікарня забезпечує медичну допомогу в різних спеціалізаціях, таких як педіатрія, неонатологія, неврологія, хірургія, кардіологія, отоларингологія та інші (рис. 2.1).



Рисунок 2.1 – Миколаївська обласна дитяча клінічна лікарня

Дана територія площею 30 111,645 м². Заклад складається з 3 основних будівель – це з головного корпусу, в якому 3 поверхи, хірургічного корпусу, 7 поверхів та поліклініки, в якому 2 поверхи. Сьогодні ліжковий фонд складає 430 ліжок. Поліклінічне відділення на 150 відвідувань. Було зроблено багато фотографій споруди зовні та зсередини, проте 3D модель – відсутня. Таким чином можна зробити висновок, що моделювання та візуалізація тривимірної моделі лікарняного закладу є актуальним завданням, оскільки подібної моделі наразі не існує та не планується. Реалізація даного проекту продемонструє технічні навички, архітектуру даної будівлі та інфраструктуру. Також робота буде цікавою для лікарів та пацієнтів. Використання моделі можливе буде в найрізноманітніших сферах.

2.1.2 Збір інформації (підготовчий етап)

Основне завдання – це створення програми, якою можна буде керувати за допомогою мобільного застосунку. Щоб поліпшити збір інформації та реалізацію в таблиці представлені вимоги до готового продукту.

Таблиця 2.1 – Вимоги до програмного продукту

Об'єкт	Миколаївська обласна дитяча клінічна лікарня
Завдання	Створення 3D моделі споруди лікарні та програмного застосунку для внутрішньої та зовнішньої навігації. Розробка навігаційної панелі для управління застосунком.
Основні елементи візуальної комунікації	Інформаційні стенди та таблички кабінетів Поверхові карти із зазначенням кабінетів і маршрутом руху (можуть бути об'єднані з інформаційними стендами) Умовні позначення, включаючи унікальне колірне оформлення (для тематичних зон, туалетів, швидкої медичної допомоги, інформаційних стійок та ін.) Показчиків за основними пішохідним напрямками
Ціль	Забезпечити спрямований рух відвідувачів Забезпечити легку орієнтацію відвідувачів на території

<p>Вимоги до навігаційних елементів</p>	<p>Простота конструкції Доступні, недорогі матеріали Назви точок повинні легко читатися (великий шрифт) Всі елементи повинні бути виконані в єдиному стилі, надаючи індивідуальність закладу, відповідно в фірмовому стилі. Кольорові рішення конструктивних елементів сігнажа повинні поєднуватися з єдиним фоном закладу</p>
<p>Отримати в результаті</p>	<p>Схему розміщення навігаційних елементів (варіанти) Дизайн окремих конструкцій і їх прив'язка Оціночні розміри елементів</p>

Побудова почалася з розуміння масштабу споруди, тобто збору інформації. Основою для створення 3D моделі миколаївської обласної дитячої клінічної лікарні служили будівельні креслення поверхів. Для відповідності з реальним будинком було використано 29 креслень. Деякі з них зображені на рисунках 2.2–2.4.

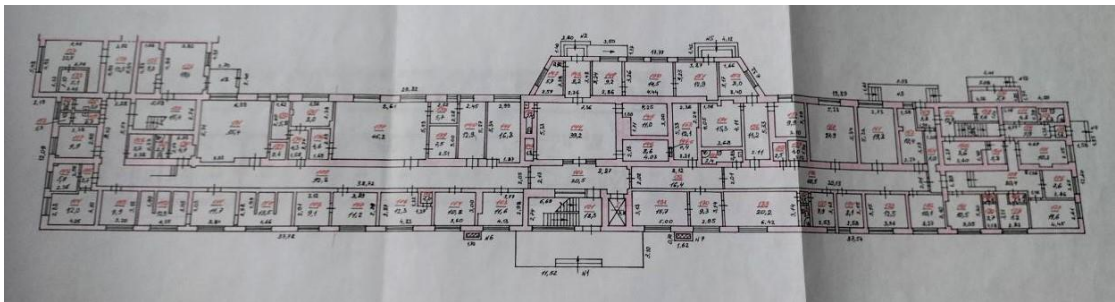


Рисунок 2.2 – Креслення головного корпусу

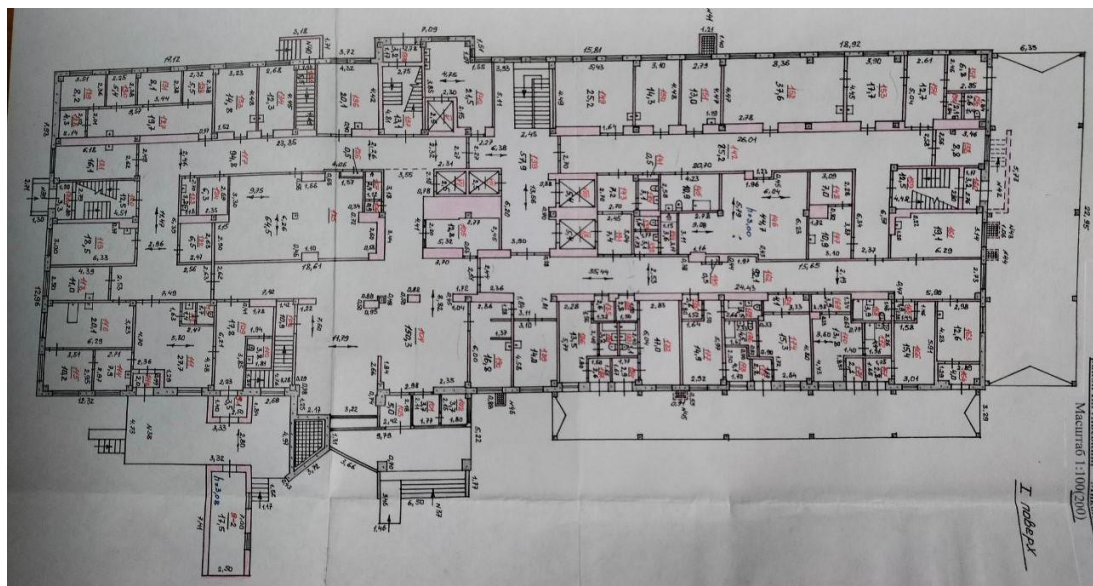


Рисунок 2.3 – Креслення хірургічного корпусу

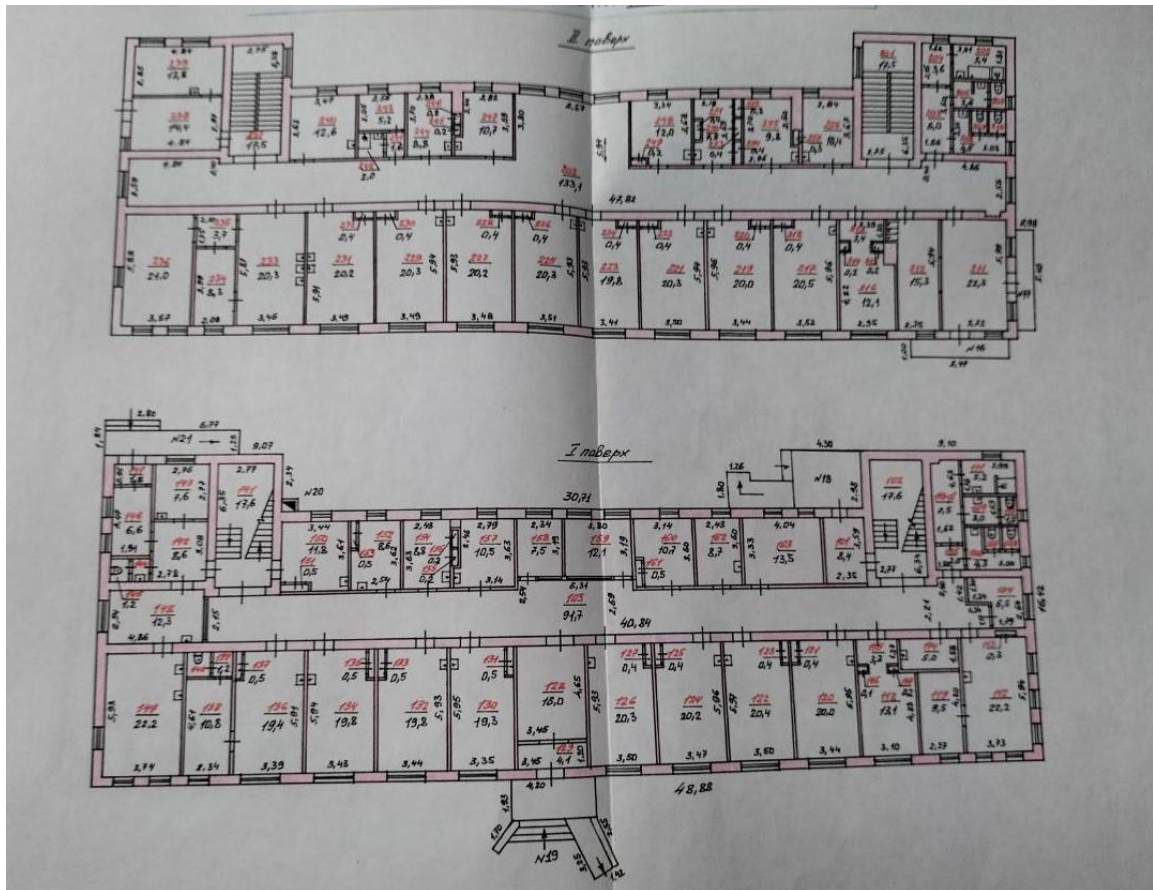


Рисунок 2.4 – Креслення поліклініки

Найбільше часу в даний етап було витрачено на з'єднання всіх корпусів в залежності від розташування висот. Так як на більшій частині креслень відсутня інформація про висоту поверхів корпусів, розмірів віконних прорізів і т.п.

Наступний етап збору інформації – це створення достатньої кількості фотографій для розуміння масштабу та вигляду кінцевого результату.

Зроблено близько 500 фотографій екстер'єрів та інтер'єрів корпусів, приміщень, коридорів, фасадів та парків. На рис. 2.5–2.8 наведено приклади. Головний корпус було сфотографовано в панорамному режимі, тому що будівля має велику довжину, а на великій відстані це не можливо зробити і за того, що на території дуже багато дерев.



Рисунок 2.5 – Головний корпус



Рисунок 2.6 – Хірургічний корпус



Рисунок 2.7 – Поліклініка.

Кожна фотографія використовувалася для аналізу з кресленнями та реальними замірами елементів будівлі при створенні 3D моделей.



Рисунок 2.8 – Хол хірургічного корпусу

Для створення фотографій використовувалася смартфон ОРРО А74. Основний об'єктив: 48-мегапіксельна камера з діафрагмою $f/1.7$. Цей об'єктив призначений для основної зйомки та забезпечує високу якість зображень (рис. 2.9) [7].

Основна камера Орро А74 забезпечує різноманітні можливості зйомки та дозволяє вам отримувати якісні фотографії у різних умовах освітлення.



Рисунок 2.9 – ОРРО А74

2.1.3 Вибір інструментів реалізації.

При виборі інструментів для реалізації завдання було розглянуто декілька найпопулярніших 3D редакторів, які працюють в ОС Windows: 3ds Max та Maya від компанії Autodesk, Blender та ZBrush.

Для наглядного порівняння складено таблицю характеристик всіх перерахованих програм (табл. 2.2).

Таблиця 2.2 – Порівняння програм 3D моделювання

№	Назва ПП	Переваги	Недоліки
1	3ds Max	<ul style="list-style-type: none"> – безкоштовний(тільки для навчання); – працює на ОС Windows; – простий та зрозумілий у використанні; – призначений для 3D моделювання; – багато способів вирішення задач. 	<ul style="list-style-type: none"> – потребує високих технічних можливостей.
2	Maya	<ul style="list-style-type: none"> – безкоштовний (тільки для навчання); – працює на ОС Windows; – багато способів вирішення задач. 	<ul style="list-style-type: none"> – зазвичай використовується для анімації; – потребує високих технічних можливостей
3	Blender	<ul style="list-style-type: none"> – безкоштовний; – працює на ОС Windows та Linux; – призначений для тримірного моделювання та анімації. 	<ul style="list-style-type: none"> – взаємодіє лише зпрограмним продуктом V-Ray; – використовується для створення ігор.
4	ZBrush	<ul style="list-style-type: none"> – багатофункціональний; – можливість створення високоякісного полігонального об'єкта. 	<ul style="list-style-type: none"> – платний продукт; – складний у використанні; – призначений для моделювання персонажів; – потребує високих технічних можливостей.

Отже, після порівняння програм, визначаємо, що найбільш підходить для даної задачі програма Autodesk 3ds Max. Крім програми для безпосередньо моделювання, необхідні засоби для отримання кінцевої візуалізації. Виходячи з того, що обрано програму 3ds Max, для візуалізації в ній широко застосовують такі додатки, як Corona Render [18] та V-Ray [20]. Їх порівняльну характеристику наведено в таблиці 2.3.

Таблиця 2.3 – Порівняння програм 3D моделювання

№	Назва ПП	Переваги	Недоліки
1	Corona Render	<ul style="list-style-type: none"> – легкий у використанні; – інтуїтивно зрозумілий інтерфейс; – безкоштовний; – вбудована бібліотека матеріалів; – висока якість рендерингу. 	<ul style="list-style-type: none"> – рендер займає багато часу; – потребує високих технічних можливостей; – платний продукт
2	V-Ray	<ul style="list-style-type: none"> – висока якість рендерингу; – великий вибір текстур (матеріалів); – взаємодіє з багатьма альтернативними програмами. 	<ul style="list-style-type: none"> – складний у використанні; – платний продукт; – потребує високих технічних можливостей – рендер займає багато часу.

Розглянувши характеристики візуалізаторів можна зробити висновок, що в даному випадку більш підходить Corona Render. Таким чином, для реалізації поставленої мети проекту було обрано Autodesk 3ds Max 2023 для проектування моделей та Corona Render – для візуалізації сцени.

2.1.4 Моделювання Low Poly моделей

Low poly (low polygon) є терміном, який описує стиль графіки, що використовує обмежену кількість полігонів для створення 3D моделей. У цьому стилі моделі мають малу кількість вершин і полігонів, що призводить до спрощеної топології та меншої деталізації [16].

Основна ідея лежить у використанні мінімальної кількості полігонів для зображення об'єктів. Замість детальних моделей з великою кількістю полігонів, у стилі low poly використовуються геометричні форми з низькою роздільною здатністю. Це створює абстрактний, стилізований вигляд моделей.

Стиль low poly часто використовується в інді-іграх, мобільних додатках, анімації та візуальних ефектах. Він має свої переваги, такі як ефективність в ресурсоемних середовищах, легкість моделювання та анімації, а також унікальний візуальний стиль. Він також може надати почуття ностальгії та викликати естетичне задоволення.

Щоб створити ефект low poly, моделі зазвичай мають гранулярні ребра та витягнуті кути. Колірні палітри можуть бути обмеженими, використовуючи різні відтінки одного кольору або декілька основних кольорів.

У процесі створення моделей в стилі low poly можна використовувати програми для моделювання, такі як Blender, Maya або 3ds Max. Також можна застосовувати техніки текстурування та освітлення, щоб підкреслити стиль low poly.

Загалом, стиль low poly надає можливість створювати унікальні та стилізовані 3D моделі, якігрових персонажів, архітектурних споруд, транспортних засобів та багатьох інших об'єктів. Він знаходить застосування в різних галузях, включаючи інтерактивне медіа, відеоігри, анімацію, візуалізацію даних та багато інших.

Основні переваги стилю low poly включають:

- 1) ефективність: через малу кількість полігонів моделі вимагають менше ресурсів обчислювальної потужності і пам'яті, що дозволяє їх

використовувати в ресурсоємних середовищах, таких як мобільні платформи або веб-браузери;

2) стилізований вигляд: low poly надає унікальний візуальний стиль з абстрактними формами, гранулярними ребрами та яскравими кольорами. Цей стиль може створювати ефект вираженості та ностальгії;

3) легкість створення: порівняно з деталізованими моделями, створення low poly моделей вимагає менше часу і навичок. Завдяки спрощеній геометрії та обмеженій деталізації, процес моделювання та текстурування може бути швидшим і доступнішим для початківців;

4) ігрова оптимізація: low poly моделі можуть допомогти забезпечити стабільну продуктивність в ігрових проектах, особливо на платформах з обмеженою потужністю, де важлива швидкість рендерингу та оптимізація ресурсів.

Узагальнюючи, стиль low poly дозволяє створювати привабливі, стилізовані 3D моделі з обмеженою кількістю полігонів.

З початку створення треба вже знати для чого буде використана модель. В цьому випадку для доробки та експорту в ігровий двигун. Тому слід дотримуватися правильну топологію полігонів. Модель складається з сітки квадратів (полігонів), кожен з яких повинен мати 4 сторони.

Також слід врахувати, що модель поділяється на внутрішню та зовнішню оболонки, як окремі об'єкти.

На рисунках 2.10–2.12 зображено поетапне моделювання різних елементів комплексу.

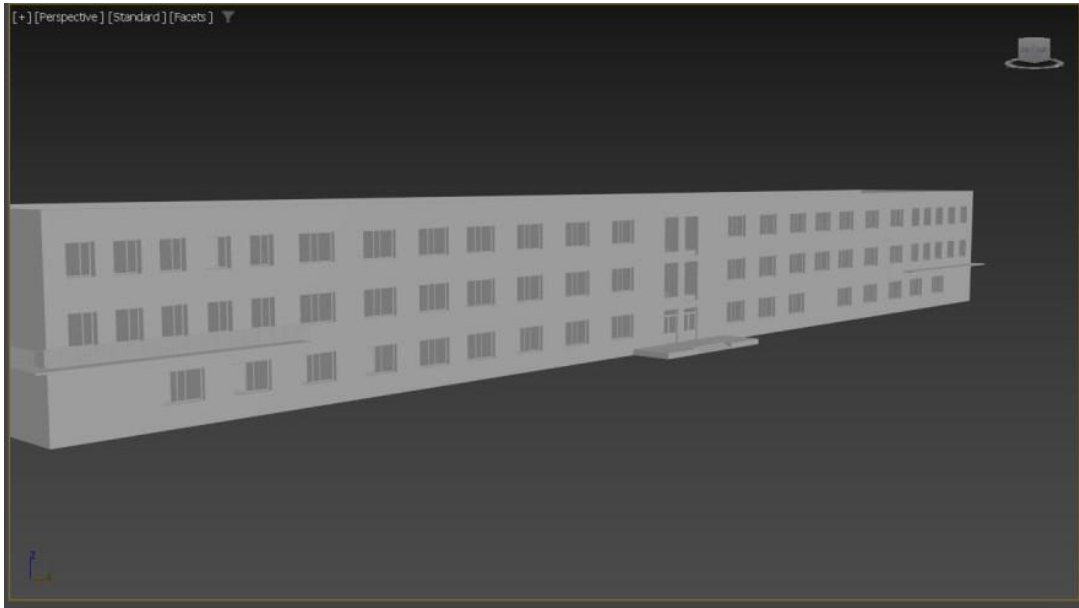


Рисунок 2.10 – Процес моделювання головного корпусу

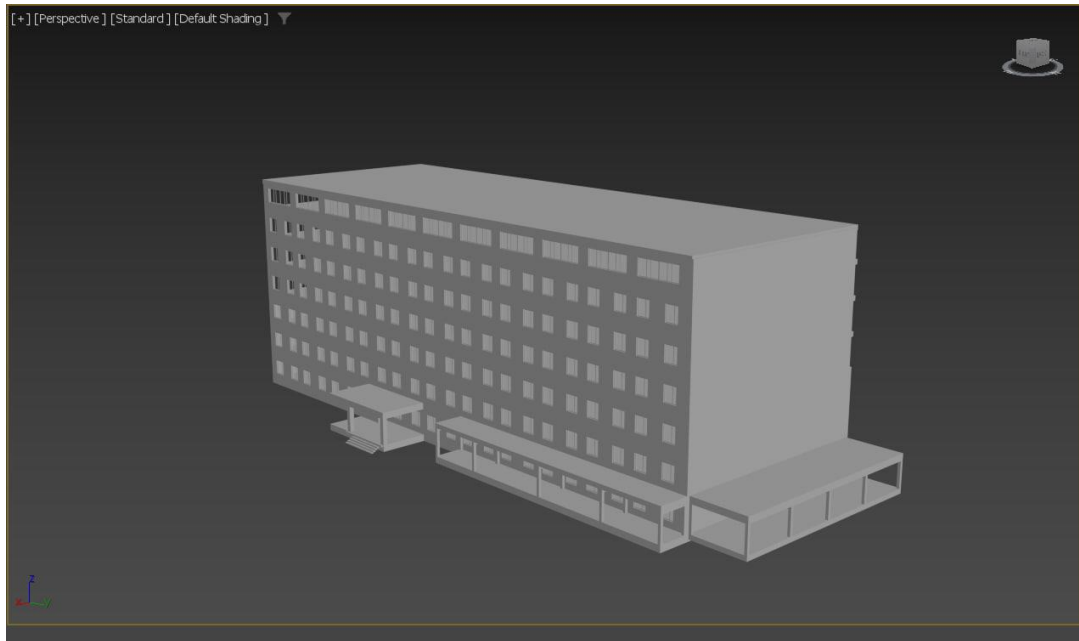


Рисунок 2.11 – Модель хірургічного комплексу

Корпуси змодельовані в масштабі 1:1 з урахуванням всіх креслень, заздалегідь зроблених фото, які використовувалися в якості орієнтування, з точністю до мм.

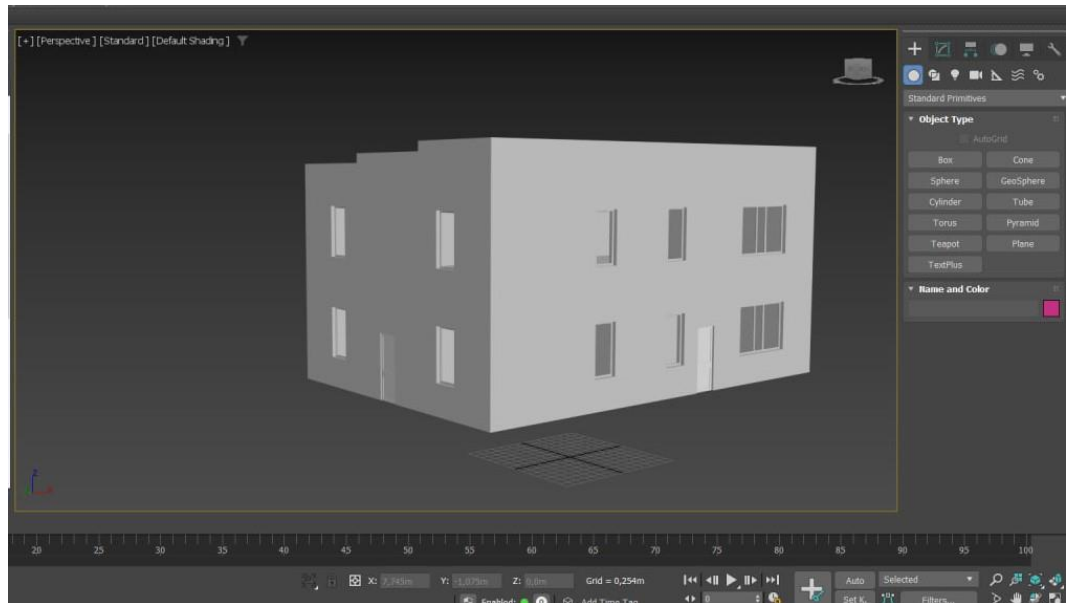


Рисунок 2.12 – Модель Патологоанатомічного корпусу

Кожен поверх моделювався окремо. Всі дверні та віконні прорізи на поверхах 300 мм, зі сторони корпусу (зовнішньої оболонки) 200мм. Товщина несущих стін 500 мм.

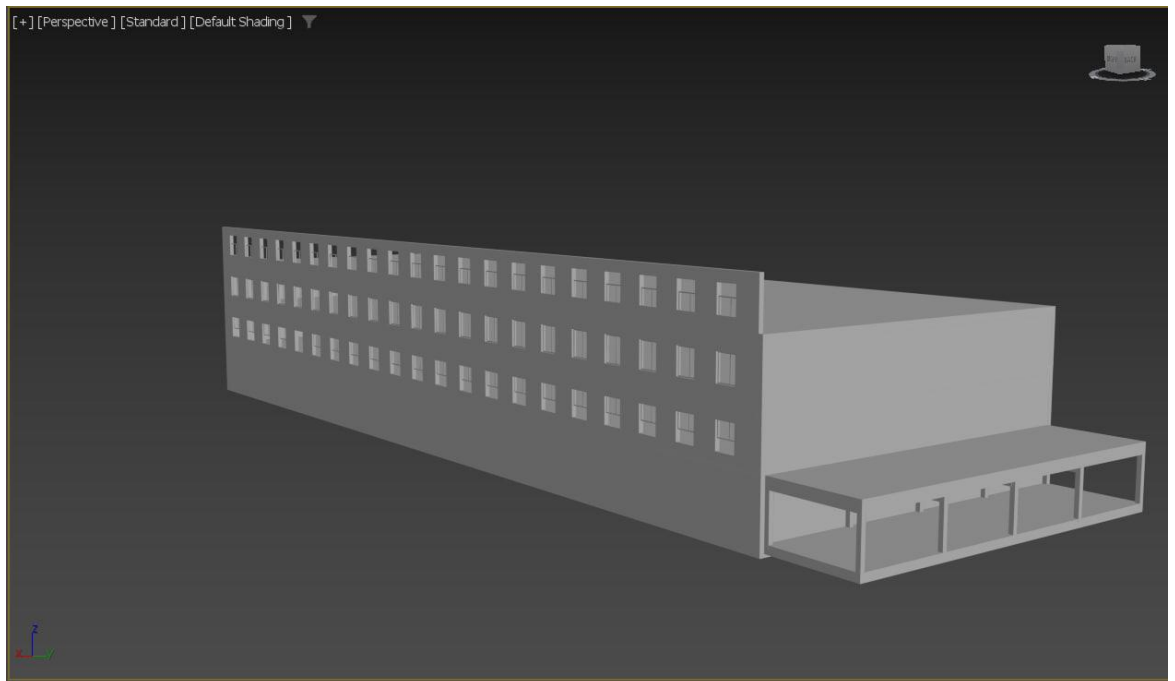


Рисунок 2.13 – Етап моделювання поверхів

Поверхи теж змодельований окремо по шарах, щоб взаємодіяти і редагувати, як самостійний об'єкт.

2.1.5 Створення High Poly

Наступним етапом йде створення High Poly моделі в 3Ds max 2023.

High poly (високополігонний) – це стиль в комп'ютерній графіці, який використовує багато полігонів для створення деталізованих та реалістичних моделей. У порівнянні зі стилем low poly, де моделі мають обмежену кількість полігонів, стиль high poly дозволяє створювати більш складні та докладні моделі [2].

Створення high poly моделей вимагає більшої обчислювальної потужності та ресурсів, оскільки більша кількість полігонів потребує більше пам'яті та обробки. Однак, цей стиль дозволяє отримати більш реалістичні та деталізовані результати, особливо коли розглядається модель зблизька.

High poly моделі використовуються в різних галузях, включаючи відеоігри, фільми, архітектуру та промисловий дизайн. Вони дозволяють створювати віртуальні об'єкти та світи, які виглядають максимально реалістично та детально.

Створення high poly моделей може виконуватися за допомогою спеціалізованих програм для 3D моделювання, таких як Autodesk Maya, 3ds Max, ZBrush та інші. Ці програми надають розширені можливості для створення складних геометричних форм, додавання деталей, текстурування та освітлення.

Використання high poly моделей дозволяє створювати вражаючі візуальні ефекти, особливо при використанні передових технологій рендерингу та освітлення. Однак, через велику кількість полігонів, ці моделі можуть вимагати більше часу для обробки та обчислень.

Узагалі, стиль high poly дозволяє створювати реалістичні та деталізовані моделі з високою кількістю полігонів. Це важливо для ситуацій, коли потрібна максимальна деталізація, наприклад, в кінематографічних сценах або при розробці детальних архітектурних моделей.

Однак, використання high poly моделей має свої обмеження. Велика кількість полігонів може призвести до значного збільшення обсягу файлу моделі і вимагати більше обчислювальних ресурсів для рендерингу та анімації. Тому, для оптимізації продуктивності, часто використовуються техніки зниження полігонів, такі як retopology або normal mapping.

Крім того, high poly моделі можуть бути використані як основа для створення low poly версій. Це дозволяє зберегти деталізацію та форму об'єкта, але зменшити кількість полігонів для покращення продуктивності та оптимізації для реального часу.

Високополігонні моделі широко застосовуються у відеоіграх, анімації, візуалізації, медичній та інженерній графіці. Вони дозволяють створювати детальні та реалістичні об'єкти, які захоплюють увагу глядачів та користувачів.

З розвитком технологій комп'ютерної графіки, високополігонні моделі стають ще більш детальними та складними. Вони використовуються для створення реалістичних персонажів, середовищ, транспорту та інших об'єктів. У майбутньому можна очікувати подальшого розвитку високополігонної графіки з використанням нових технологій та методів створення моделей.

Для перевірки якості моделі на відображення та співвідношення в 3D Max використовується сторонній візуалізатор Corona Render.

Corona Render – один з відносно молодих рендерер (движків візуалізації), який швидко завоював любов архітектурних візуалізаторів і склав серйозну конкуренцію V-ray. Все більше початківців 3D фахівців стартують саме зі зв'язкою 3ds Max + Corona Render [18].

Створюємо копію та перетворюємо low poly в high poly методом збільшення полігональної сітки, тим самим створюючи гострі кути більш гладкими і плавними.

2.2 Використання застосунку Blender

Blender – це вільний та відкритий 3D-графічний пакет, який використовується для моделювання, анімації, візуалізації та рендерингу об'єктів 3D. Blender розроблений Фондом розвитку вільного програмного забезпечення (Blender Foundation) і має активну спільноту користувачів і розробників [13].

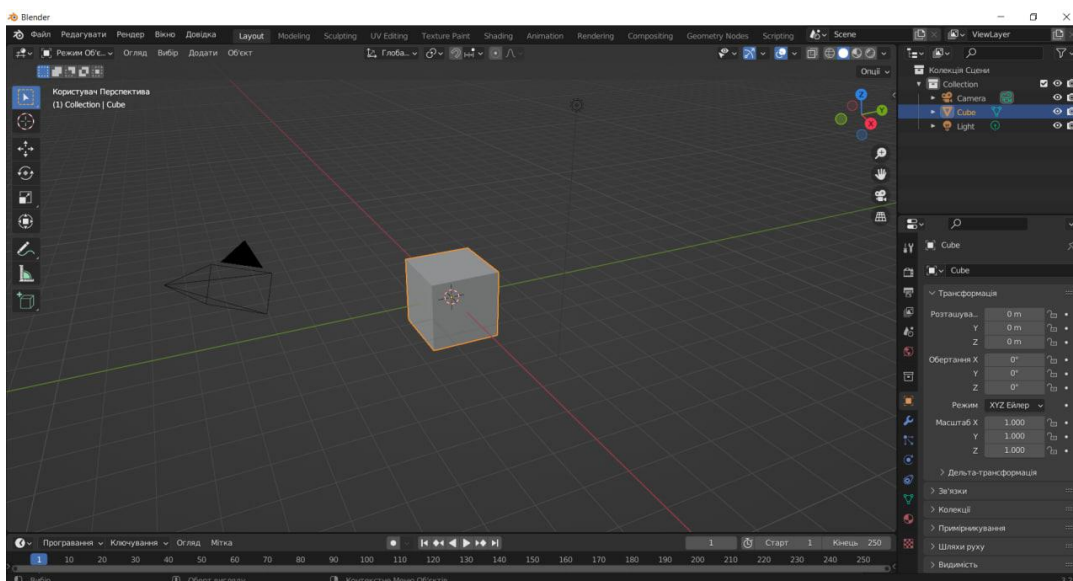
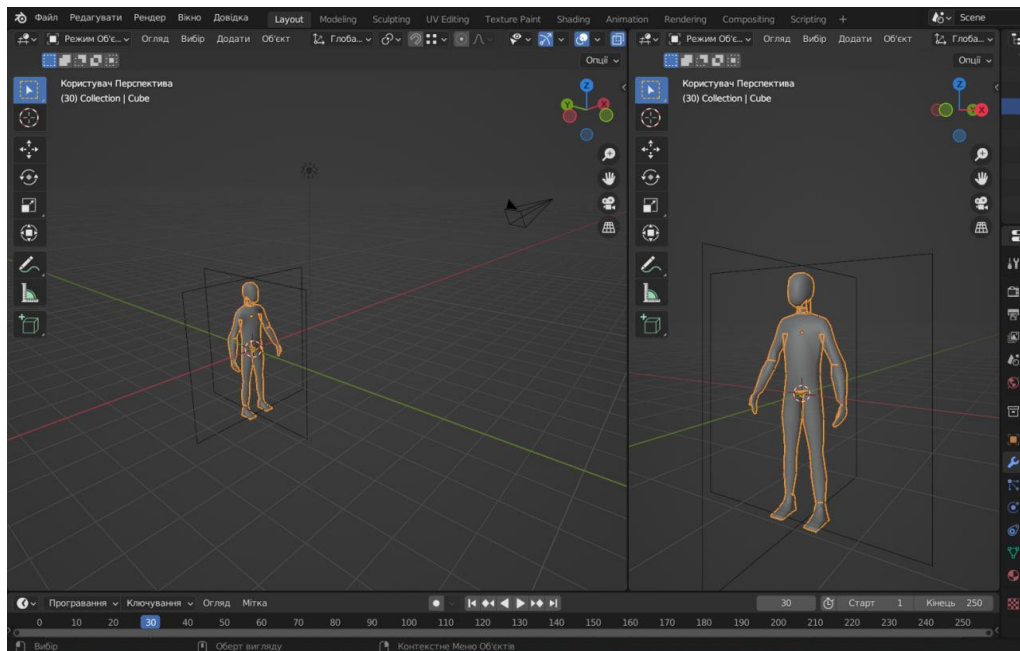


Рисунок 2.14 – Blender

Blender надає широкий набір інструментів для створення 3D-моделей, включаючи полігональне, поверхнєве, каркасне моделювання та багатошаровий режим скульптування. Ви можете створювати складні анімації за допомогою різних методів, включаючи ключові кадри, криві анімації, розіграші фізики, симуляції руху та багато іншого. Він підтримує потужні рендерингові двигуни, такі як Cycles та Eevee. Вони дозволяють створювати фотореалістичні зображення та анімацію з реалістичним освітленням, матеріалами та ефектами. Має вбудовані інструменти для візуалізації даних та симуляцій, включаючи флюїдну симуляцію, симуляцію тканин, симуляцію частинок та багато іншого. Ви можете обробляти та складати відео, додавати ефекти, коригувати кольори та застосовувати різні фільтри та шари. Blender підтримує мову програмування Python, що дозволяє автоматизувати завдання та створювати додаткові інструменти або розширення за допомогою сценаріїв.

Blender також підтримує розширення, створення додаткових модулів та плагінів, що розширюють функціональність програми. Завдяки цьому, ви можете використовувати сторонні розширення, або навіть розробляти власні, щоб відповідати своїм потребам. Blender є потужним інструментом для 3D-моделювання, анімації та візуалізації. Він широко використовується як професійними художниками і дизайнерами, так і аматорами, які бажають створювати власні проекти у сфері 3D-графіки.

Ця програма була використана для створення моделі в вигляді людини, якою буде показуватись маршрут (рис. 2.15).



Рисинук 2.15 – Готова модель

Створено базові форми персонажа за допомогою простих геометричних примітивів, таких як куби, сфери та циліндри [17]. На цьому етапі встановили загальні пропорції та структуру персонажа. Після цього було використано інструменти моделювання Blender для вдосконалення форм та додавання деталей до персонажа. Розміри для персонажа були встановлені, такі ж самі які має середньостатистична людина.

2.3 Підбір та опис апаратної частини сервісу навігації

2.3.1 Клієнт сервер

Для обробки програмного застосунку слід підібрати необхідні компоненти що відповідатимуть наступним вимогам:

- операційна система: Windows 10/11/ Linux Debian;
- процесор: Intel Core i5-10400F2.9-4.3GHz;
- оперативна пам'ять: 2x8GB DDR--3200;
- відеокарта: Nvidia GeForce GTX 1650;
- вільне місце: 480 Гб.

Для взаємодії з інтерфейсом програми і відображення красивої картинки був використаний сенсорний монітор Asus MB16AHP, який має IPS матрицю з діагоналлю 15.6 дюйми та максимальну роздільну здатність FullHD, що дає максимальну зручність використання (рис. 2.16). Має можливість встановлення як на столі так і кріпитися на стіну.



Рисунок 2.16 – Монітор Asus MB16AHP

За прийом і обробку даних з відображенням на монітор, в якості термінального клієнта, відповідає міні комп'ютер ARTLINE Overlord V8v33 (рис. 2.17).



Рисунок 2.17 – Міні комп'ютер ARTLINE Overlord V8v33

Продуктивне рішення, як для Міні ПК, в компактному, повністю виготовлений з шліфованого алюмінію. За рахунок потужного вбудованого відеоядра, без проблем обробляє FullHD відео. Основна перевага – дуже маленький розмір 26см x 25см x 39 см. Добре підходить в якості термінального клієнта чи офісного комп'ютера. Підтримує установку: Windows 7, 8, 10, десктопних версій Linux, збірок на Android x86-x64.

2.3.2 Графічна станція

Кожну людину оточують такі предмети і речі, які виробляються не без участі комп'ютерів як спеціалізованих, так і стандартних. Кожна область сьогодні не може обійтися без обчислювальної техніки. Вона автоматизує процеси, виробляє найскладніші розрахунки, страхує від людського фактора. Графічна станція обов'язкове в архітектурі та інженерії, кіновиробництві і мультиплікації, літако- і автомобілебудуванні. Без неї не обходиться сучасна візуалізація, складні розрахунки і складання схем.

Щоб обробляти та передавати об'єм інформації знадобиться ГС, яка буде виконувати роль основного серверу.

З цим постала ціла низка завдань, для яких потрібні високопродуктивні GPU. Так, відеокарти – основа цього класу пристроїв. На жаль, не кожен графічний прискорювач підійде. Якщо раніше для такого обладнання доводилося розробляти індивідуальне рішення, що вимагало величезних витрат, то зараз ринок непогано насичений потрібними комплектуючими, які можна купити і поставити в власний персональний комп'ютер.

Застосовуються ГС в наступних сферах:

- 3D-моделювання;
- рендеринг;
- обробка графічних даних.

Відповідно, дане обладнання можна зустріти повсюдно. У модельєрів, архітекторів, мультиплікаторів і по аналогії.

Для даного ряду завдань добре підійде графічна станція Tower (рис. 2.18), яка має такі характеристики:

- встановлений процесор – AMD Ryzen 9 3950X;
- сумарна кількість ядер / потоків – 16 / 32;
- частота процесора при повному завантаженні всіх ядер – 3,9 GHz;
- максимальна частота при завантаженні одного чи декількох ядер (Turbo Boost) – 4,7 GHz;
- кеш-пам'ять третього рівня – 64 MB;
- об'єм оперативної пам'яті – 64 GB;
- тип пам'яті DDR4;
- частота оперативної пам'яті – 3600 MHz;
- кількість встановлених модулів – 2 шт;
- загальна кількість слотів для підключення оперативної пам'яті – 4 шт.
- підтримка 2-канального режиму роботи пам'яті.



Рисунок 2.18 – Графічна станція AMD Ryzen 9 3950X

Якщо користувачеві потрібні графічні станції для 3D, але він дуже рідко працює з даним форматом, його діяльність обумовлена особистим інтересом, а не спеціальністю, то купувати дорогу обчислювальну техніку нерозумно. А тим більше коли є в запасі необмежений час, цілком можна обійтися стандартним ПК або ноутбуком.

2.3.3 Економічне обґрунтування сервісу

Все обладнання для обробки і управління програмою навігації складається з таких елементів:

Таблиця 2.2 – Економічне обґрунтування сервісу

Пристрій	Назва	Використання	Ціна
Монітор	Asus MB16AHP	Сенсорний монітор для відображення та управління програмного застосунку	405\$
Міні комп'ютер	ARTLINE Overlord V8v33	Локальний сервер який отримує та віднає оброблену інформацію	690\$
Графічна станція	Робоча станція AMD Ryzen 9 3950X BALANCE	Сервер для зберігання та обробки інформації з виводом на клиент-сервер	1600\$

Якщо розглядати дану систему, як самостійний комплекс, то це є найкращім рішенням на відміну від того, що можуть запропонувати комерційні компанії.

Висновки до розділу 2

В другому розділі було розглянуто, досліджено та послідовно описано складний процес створення комп'ютерної тривимірної моделі Миколаївської обласної дитячої клінічної лікарні. Були обрані інструменти для моделювання та візуалізації – Autodesk 3ds Max та Corona Render. За допомогою програмного забезпечення Autodesk 3ds Max було проведено моделювання будівель, що розташовані на території клініки. Були створені головний корпус, поліклініка, хірургічний корпус, патологоанатомічний корпус та інші невеликі споруди. Було описано процес створення деяких елементів.

Також у цьому розділі було надано обґрунтування та опис характеристик компонентів апаратної частини навігаційного комплексу, який знаходиться у процесі розробки.

3 РОЗРОБКА ПРОГРАМНОГО ЗАСТОСУНКУ

3.1 Компіляція програми

Розробка та компіляція програмного застосунку проводилася в Unity 2022. На рисунку 3.1 показано як виглядає застосунок. Unity – це ігровий движок, на якому будується більшість сучасних ігор та застосунків на різних платформах. За допомогою движка можна створювати ігри і застосунки, спрямовані на запуск на комп'ютерах (Windows, Linux, MacOS), мобільних телефонах, планшетних комп'ютерах (Android, iOS, Windows Phone) і навіть ігрових приставках PlayStation, Xbox, Nintendo.

Створення ігор на двигуні Unity неможливо уявити без написання додаткового коду. У Unity підтримується дві мови програмування – JavaScript та C#. Раніше також підтримувалась мова Boo, але пізніше від неї відмовилися. Ви можете самостійно вибрати якоюсь мовою програмувати в Unity, оскільки особливого значення це не має. Більшість розробників віддають перевагу мові C#, тому що вона більш потужна і знайти різну документацію набагато простіше, ніж для мови JavaScript. На даному етапі програмний застосунок створюється для ОС Android.

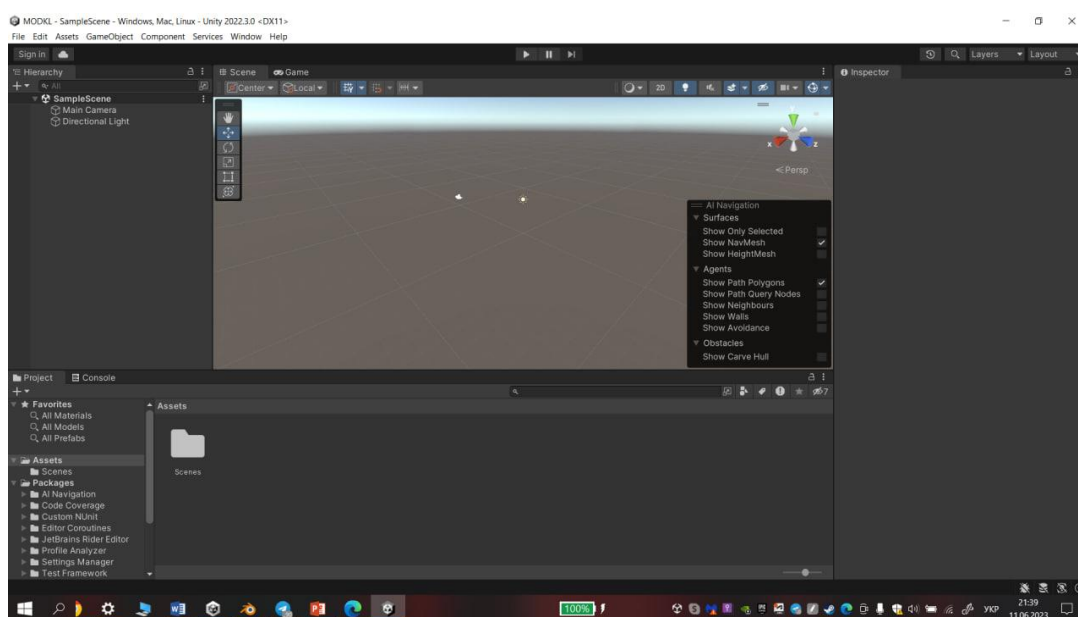


Рисунок 3.1 – Інтерфейс програмного середовища Unity

В Unity проект розділяється на сцени (рівні) – окремі файли, що містять в собі ігрові світи зі своїм набором об'єктів, налаштувань та сценаріїв. В собі сцени можуть мати як власні об'єкти (моделі), так і породні об'єкти – тобто об'єкти які не мають моделі. Об'єкти, в свою чергу мають набори компонентів, з якими працюють скрипти. У кожного об'єкта має бути назва, середовище Unity на відмінну від інших програмних застосунків дозволяє мати два, чи більше об'єктів з однаковими назвами. У об'єкта також може бути мітка, та шар на якому він має відображатися. Також у кожного об'єкта наявний такий компонент як Transform (трансформація) – і ньому зберігається інформація про координати місцезнаходження, оберту і масштабу об'єкта відповідно до осей X, Y, Z. Ще за замовчуванням у об'єктів, що мають видиму геометрію, присутній компонент Mesh Renderer, котрий і робить модель об'єкта видимою.

Всередині ігрового движку, вмонтовано фізичний движок, котрий дозволяє промоделювати взаємодію об'єктів між собою по певним фізичним законам. В грі не кожен об'єкт моделюється, як фізичне тіло і на те є дві причини. Перша причина полягає у тому, що даний процес потребує додаткових ресурсів. Друга причина, це те що в цьому банально немає ніякого сенсу.

C# є мовою програмування, яка широко використовується в розробці програмного забезпечення, включаючи розробку ігор. Вона була розроблена компанією Microsoft і вперше випущена у 2000 році.

Ось деякі важливі риси мови C# [10]:

- C# підтримує концепції об'єктно-орієнтованого програмування, такі як спадкування, поліморфізм, інкапсуляція та абстракція;
- в C# необхідно вказувати типи даних для змінних та параметрів, і компілятор перевіряє сумісність типів в процесі компіляції;
- C# використовується в рамках платформи .NET, що надає багатий набір класів і бібліотек для розробки програмного забезпечення;

- крім об'єктно-орієнтованого програмування, C# підтримує елементи функціонального програмування, такі як анонімні методи, лямбда-вирази та виразні дерева;
- C# має вбудовані механізми безпеки, такі як перевірка меж масивів, керування пам'яттю та автоматичне управління пам'яттю;
- надає можливості для асинхронного програмування з використанням ключових слів `async` та `await`.

C# широко використовується в розробці ігор та додатків з використанням рушів, таких як Unity, які надають зручні інструменти та бібліотеки для створення ігрового досвіду.

Тому для написання застосунку, використовувалася мова програмування C# в середі розробки Visual Studio.

Visual Studio – це інтегроване середовище розробки (IDE), розроблене компанією Microsoft, яке надає потужні інструменти для програмування, налагодження, тестування та розгортання програмного забезпечення. Це одна з найпопулярніших та найбільш використовуваних IDE серед розробників (рис. 3.2).

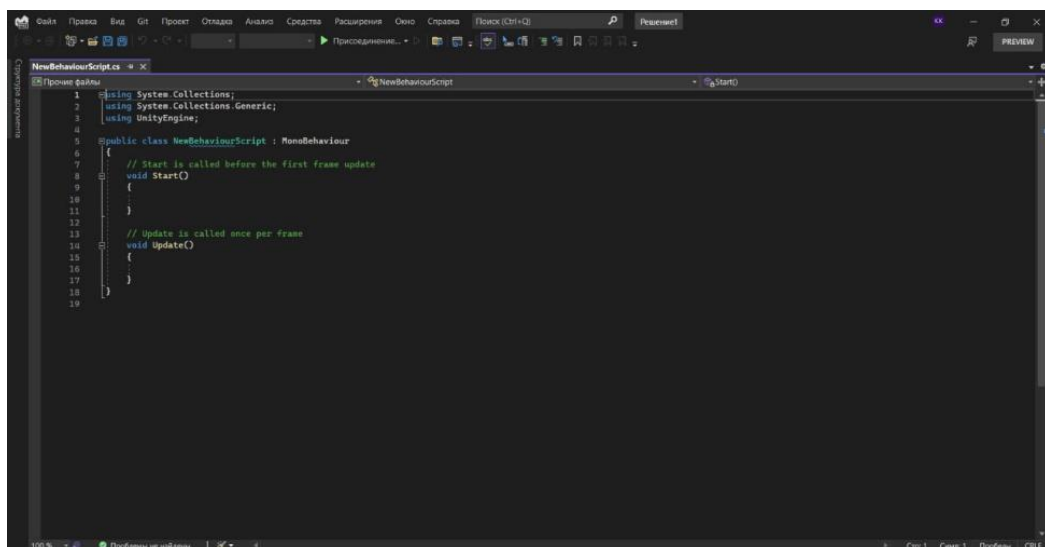


Рисунок 3.2 – Інтерфейс програмного середовища Visual Studio

Основні риси та можливості Visual Studio включають:

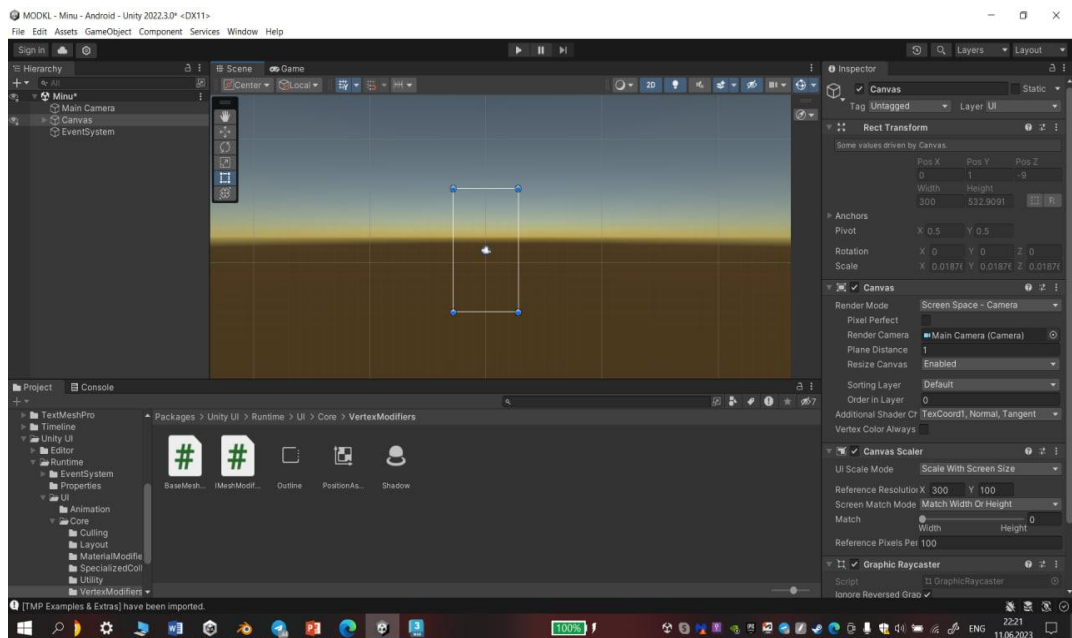
- мови програмування: Visual Studio підтримує широкий спектр мов програмування, включаючи C#, C++, Visual Basic, F#, JavaScript, TypeScript та багато інших;
- редактор коду: Він надає потужні засоби редагування коду з підсвічуванням синтаксису, автодоповненням, форматуванням та іншими функціями, які полегшують написання коду;
- відладка: Visual Studio надає розширені інструменти для відладки програм, що дозволяють знаходити та виправляти помилки, аналізувати стан змінних та виконувати крок за кроком;
- управління проектами: За допомогою Visual Studio ви можете створювати, управляти та налаштовувати проекти програмного забезпечення, включаючи налаштування компіляції, залежностей та інших параметрів;
- інтеграція з іншими інструментами: Visual Studio має вбудовану підтримку систем контролю версій, таких як Git, інтегровані засоби тестування, створення звітів, розгортання та багато інших;
- розширюваність: Visual Studio дозволяє розширювати функціональність шляхом встановлення розширень та плагінів, що дозволяють вам налаштовувати своє робоче середовище та додавати нові функції. Visual Studio доступний в різних версіях, включаючи Visual Studio Community (без обмежень для некомерційних проектів та для студентів), Visual Studio Professional (для комерційної розробки) та Visual Studio Enterprise (з розширеними можливостями для командної розробки та підприємств). Кожна версія має свої особливості та доступні функції.

Visual Studio також підтримує платформу .NET Framework, яка надає великий набір бібліотек та інструментів для розробки програмного забезпечення для різних платформ, включаючи Windows, веб-додатки, мобільні пристрої та хмарні сервіси.

Окрім того, Visual Studio має інтегровану підтримку Azure, платформи хмарних обчислень Microsoft, що дозволяє легко створювати, розгортати та керувати хмарними додатками та сервісами.

Загалом, Visual Studio є потужним інструментом для розробки програмного забезпечення з широким спектром функцій та можливостей. Він дозволяє розробникам ефективно працювати над проектами, незалежно від їх розміру та складності.

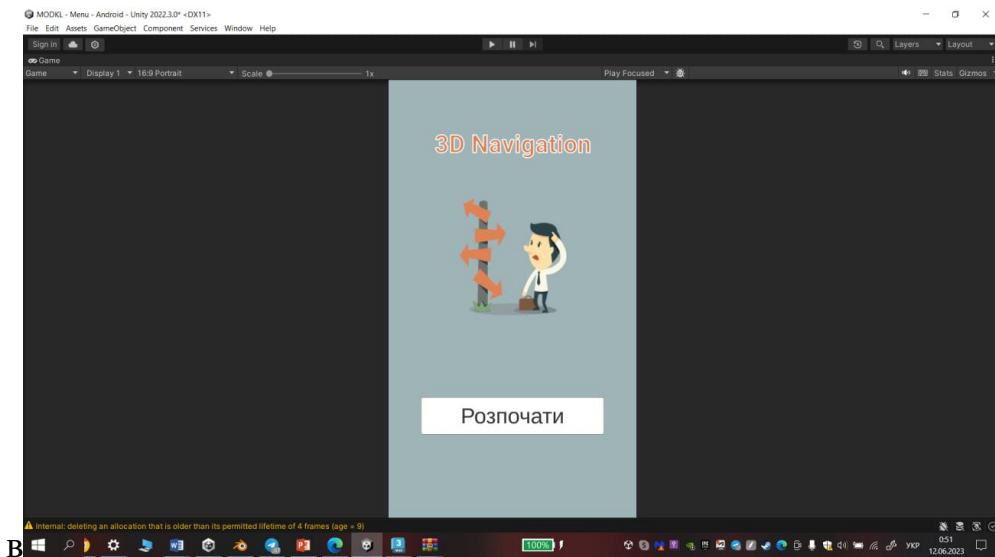
Для початку роботи треба зробити головну сторінку і її інтерфейс. Додається нова сцена в Unity в 2D вигляді і налаштовується (рис3.3).



Рисинук 3.3 – Налаштування камери

Створивши сцену Unity розміщуємо різні об'єкти, такі як кнопки, текстові поля і зображення. Для створення графічного інтерфейсу користувача (GUI) використовується компоненти UI, такі як Canvas, Button, Text, Image тощо [23]. Додаємо ці компоненти до відповідних об'єктів на сцені. Налаштовуємо розміщення, розмір, текст, зображення та інші властивості відповідно до дизайну. Додаємо обробники подій для кнопок, які будуть виконувати певні дії при їх натисканні. Наприклад, при натисканні кнопки "Розпочати" вона міняє колір і можна викликати метод, який переходить на список кабінетів куди можна потрапити. Для реалізації переходів між різними

сценами використовується функція Unity, така як *SceneManager.LoadScene()*. Функція *SceneManager.LoadScene()* в Unity використовується для завантаження нової сцени в грі. Вона дозволяє перемикатися між різними сценами під час виконання програми, де *sceneName* – рядок, що вказує ім'я сцени, яку потрібно завантажити. Використовуючи *SceneManager.LoadScene()*, ви можете динамічно переходити між різними сценами та керувати потоком гри в Unity [21]. Налаштовується логіка переходу у скрипті, щоб при натисканні відповідної кнопки виконувався перехід до необхідної сцени.



Рисинук 3.4– Готове головне меню

Наступним етапом є створення сцени на якій буде проходити навігація. Всі змодельовані об'єкти експортуються та розставляються по своїм місцям в правильному порядку, починаючи з моделей корпусів і закінчуючи елементами наповнення території. Зображено на рис. 3.5–3.7 [24].

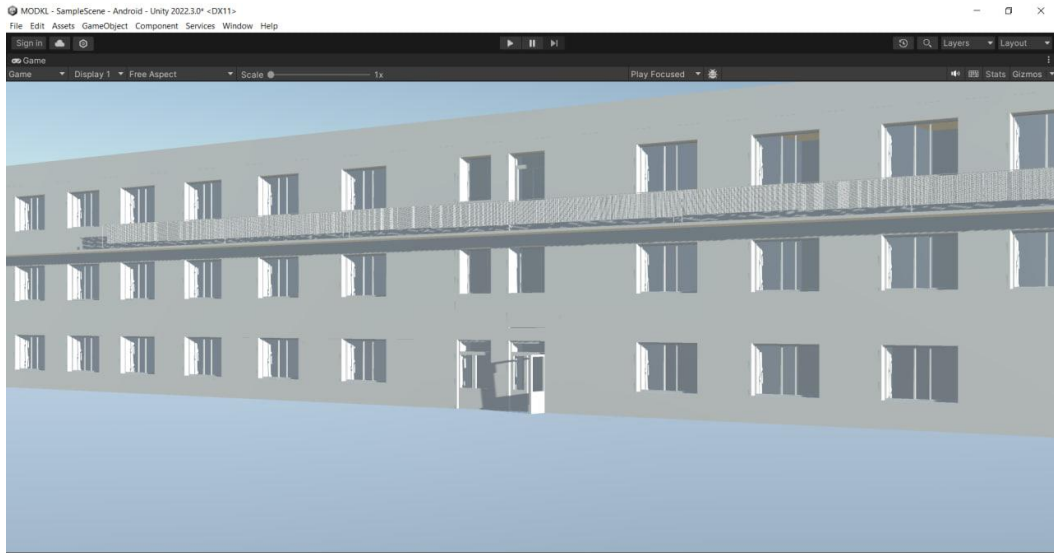


Рисунок 3.5 – Головний корпус



Рисунок 3.6 – Патологоанатомічний корпус



Рисунок 3.7 – Елементи наповнення: дерева та лавки

Додаємо нашого героя, який буде показувати маршрут, зображено на рисунку 3.8.

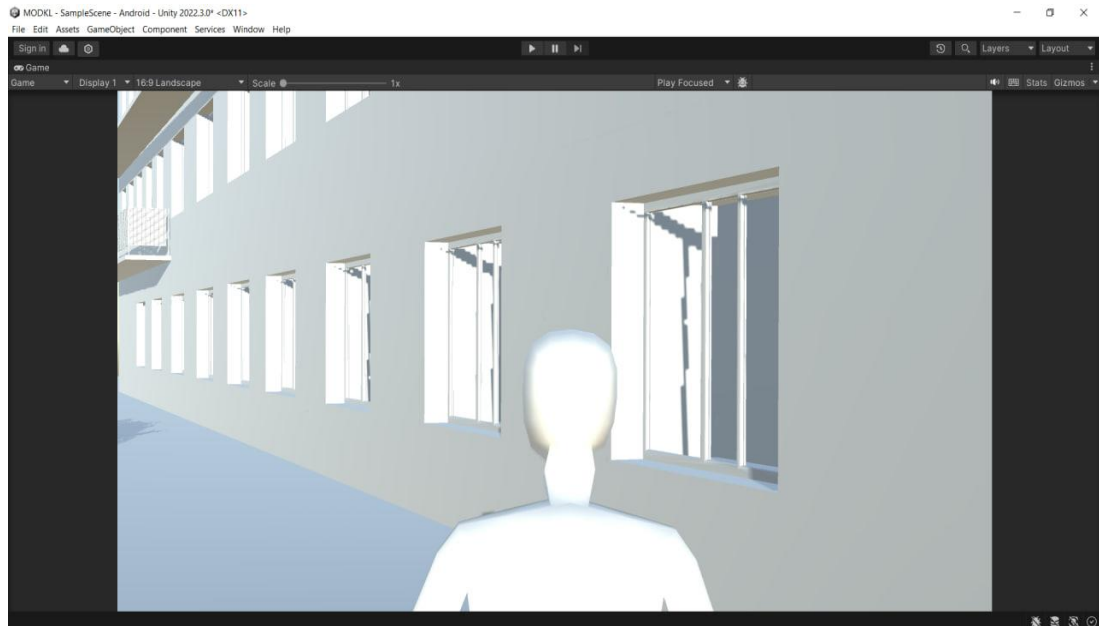


Рисунок 3.8 – Елемент навігації

3.2 Реалізація серверу

Щоб запустити додаток з Unity, треба перевірити, що сцена готова до запуску. Переконайтесь, що всі необхідні об'єкти, скрипти та налаштування готові до виконання. Вибираємо головну сцену для запуску. У верхній частині інтерфейсу Unity натискаємо на вкладку «File» (Файл) і перевіряємо, чи вибрана правильна сцена зі списку «Build Settings» (Налаштування збірки) [22]. Налаштовуємо параметри запуску. У верхній частині інтерфейсу Unity натискаємо на вкладку «File» (Файл), оберемо «Build Settings» (Налаштування збірки) і вибираємо правильні налаштування платформи (Android), роздільної здатності та інших параметрів (рис. 3.9).

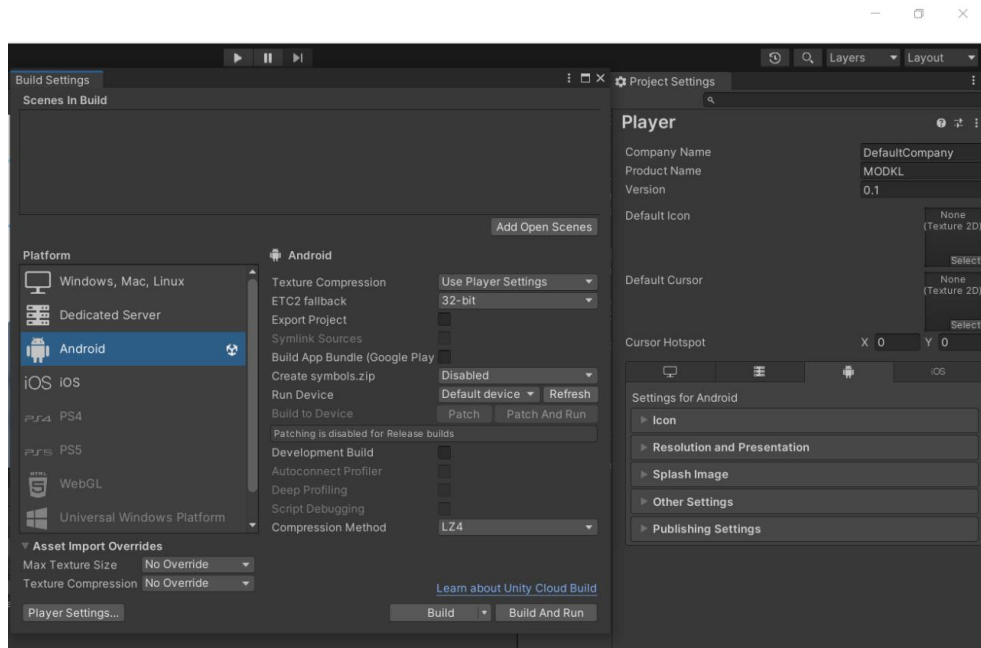


Рисунок 3.9 – Налаштування збірки

Запускаємо застосунок «Play» (Відтворити) або клавіша F5. Застосунок буде відкритий в інтерактивному режимі у вікні Unity (рис. 3.10).

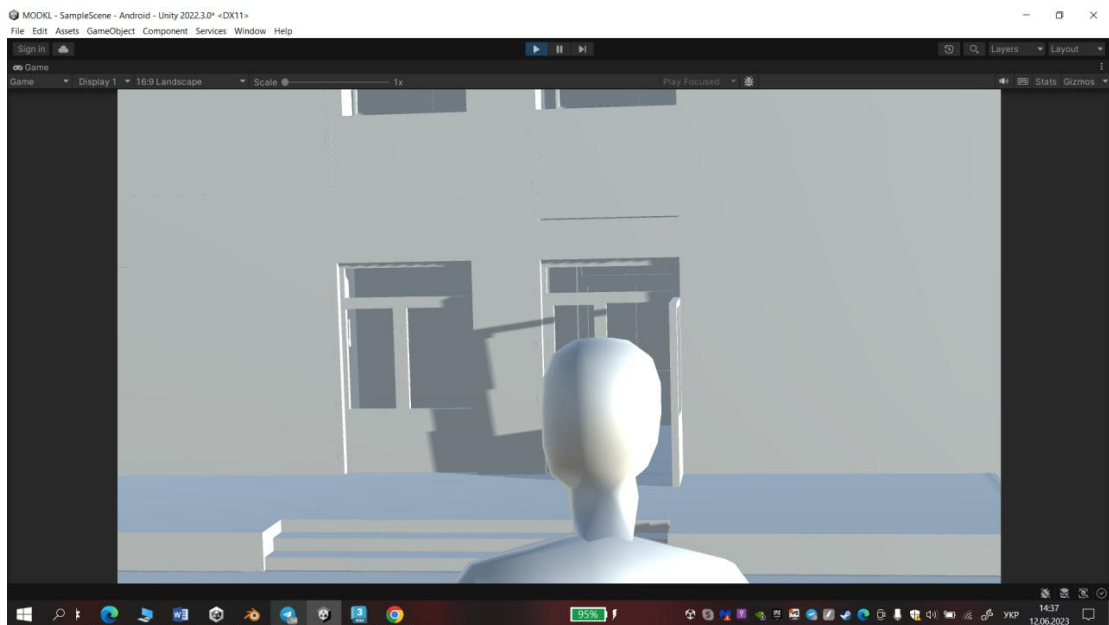


Рисунок 3.10– Запуск застосунку в Unity

Щоб запустити застосунок в повноекранному режимі або на віддаленому пристрої, натискаємо кнопку «Build and Run» (Збудувати і запустити) у вкладці «Build Settings» (Налаштування збірки). Вибераємо потрібну платформу та параметри збірки, а потім натисніть «Build and Run» (Збудувати і запустити). Unity збудує гру та автоматично запустить її на

вибраній платформі або пристрої. На рисунках 3.11-3.12 показано як виглядає це все в телефоні.



Рисунок 3.11 – Запуск застосунку на телефоні

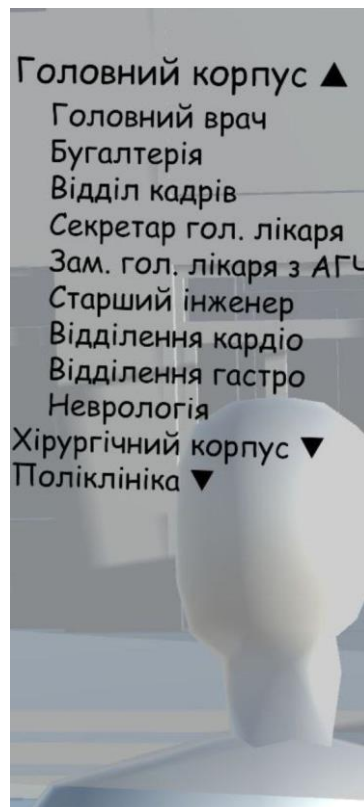


Рисунок 3.12 – Запуск застосунку на телефоні

Висновки до розділу 3

У третьому розділі надано огляд технологій, які були використані під час розробки програмної складової навігаційного сервісу. Для створення анімації було використане програмне середовище Unity як основний інструмент. Для реалізації функцій, елементів керування, відстеження та обробки даних була використана мова програмування C#. У даному розділі також наводиться опис послідовності запуску розробленого додатку для тривимірної навігації в приміщеннях Миколаївської дитячої лікарні.

4 РОБОТА КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ

У розділі розглядається особливості впровадження та інсталяції розробленого додатку, системні вимоги до пристроїв. Описується типовий сценарій користування мобільним додатком та основні його функції.

4.1 Системні вимоги та інсталяція

Наразі на данному етапі встановлення додатку на мобільний пристрій поки немає. Планується розміщення застосунку в магазині Google Play, щоб дасть змогу користувачам використовувати програмний продукт для цілей навігації в миколаївській обласній дитячій клінічній лікарні. Для роботи з іншими приміщеннями потрібно замінити плани поверхів та виконати повторну компіляцію. Пропонується розміщувати посилання на завантаження додатку поруч із маркерами синхронізації. Це дозволить новим користувачам, які ще не користувалися програмою, миттєво розпочати процес навігації. На рисунку 4.1 показано приклад відповідного плаката. У QR-коді міститься посилання для завантаження пакету програми з магазину Google Play.



Додаток
навігації

Рисунок 4.1 – Приклад постера поширення додатку

У QR-коді зашифроване посилання на завантаження пакету з програмою з сервісу Google Play.

4.2 Сценарій роботи користувача з системою

Після запуску програми користувач потрапляє на головне міню, де йому потрібно натиснути «Разпочати», подалі планується робити оновлення в яких буде з'являтися більше функціоналу. Після цього людина потрапляє на список кабінетів куди він може дійти. Людина баче маркер який знаходиться біля входу на територію або біля корпусу, в залежності куди хоче потрапити. Він виглядає таким чином, як показано на рисунку 4.2. В кожному з вибірових кабінетів закодоване своє положення. Після цього, автоматично починається рух маркера та поворот камери, який має співпадати з реальним маршрутом. На рисунку 4.2 бачимо, що покажчик (фігура в виді людини) знаходиться біля входу в головний корпус, камера повернута перпендикулярно до нього, Н

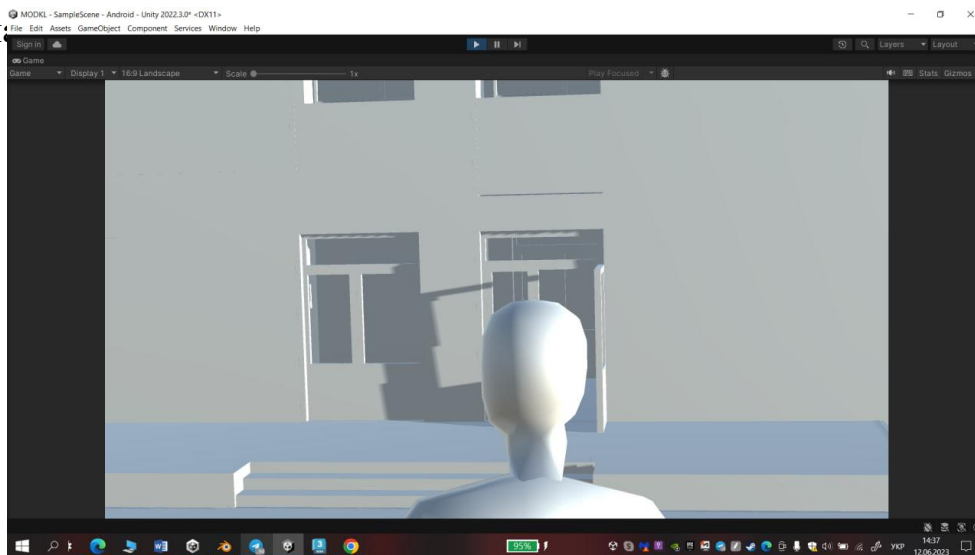


Рисунок 4.2 – Початок маршруту

Тепер переміщення мобільного пристрою відстежується та відображається як зміщення жовтого покажчика. На даному етапі користувач задає бажану кінцеву точку навігації.

Програма підтримує поки що тільки один спосіб обрання точки призначення. Це є більш очевидним та полягає у виборі доступних об'єктів зі списку в правому верхньому куті інтерфейсу (рисунок 4.3).

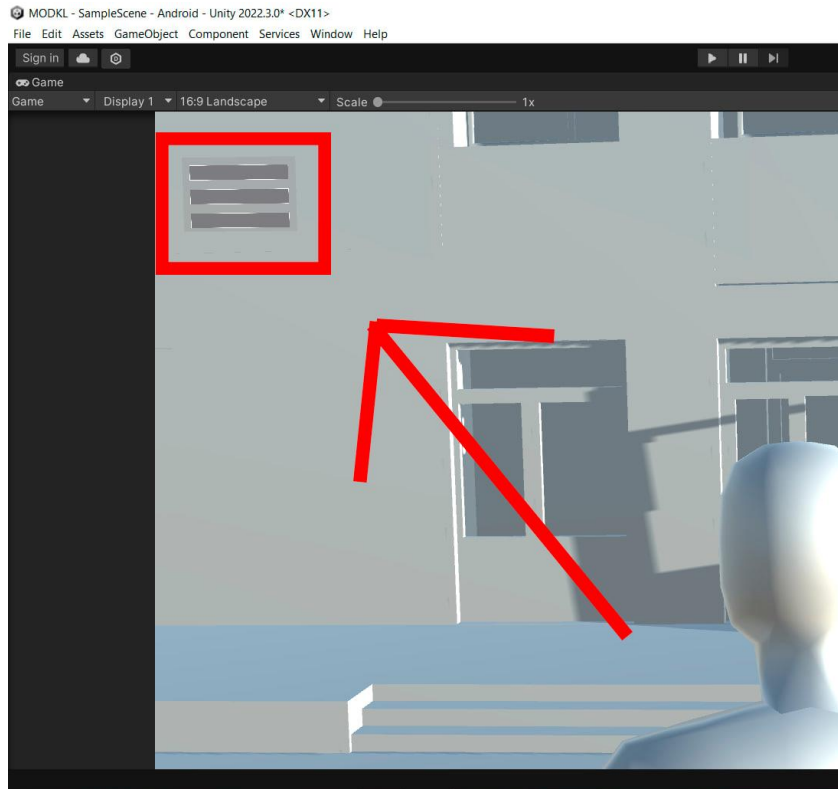


Рисунок 4.3 – Кнопка, яка викликає список доступних об'єктів

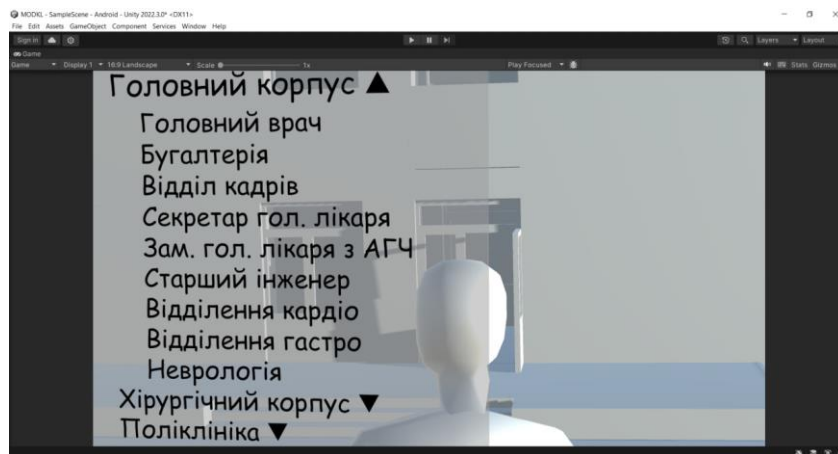


Рисунок 4.4 – Список об'єктів

Коли людина дійшла до якогось місця, ця точка зберігається і з цього ж місця вона може піти в інше.

Висновки до розділу 4

На етапі розгортання програмного продукту на мобільних пристроях, процес встановлення додатку ще не був завершений. Проте, планується розміщення застосунку в магазині Google Play з метою забезпечення доступності для користувачів.

У роботі також були описані системні вимоги та послідовність дій для розгортання розробленого програмного продукту. Був розглянутий сценарій роботи з мобільним додатком, вказано послідовність дій користувача, а також наведено основні функції системи.

ВИСНОВОК

Під час виконання кваліфікаційної дипломної роботи були досягнуті поставлені цілі та завдання, а саме: розробка програмно-апаратного комплексу для 3D навігації у будівлях. Було створено програмний засіб управління та переміщення для миколаївської обласної дитячої клінічної лікарні на платформі Unity з використанням мови C# для операційної системи Android. У майбутньому планується завантаження програмного засобу в Play Market.

Проте варто відзначити, що створений програмний засіб все ще знаходиться у стадії розробки. Для подальшого вдосконалення роботи необхідно провести його доопрацювання, враховуючи можливість заміни та обробки елементів, розробити складніші алгоритми для кожного об'єкту.

Покращення системи можна почати з модернізації апаратної частини і програмного забезпечення, а також розглянути інші фреймворки. Сьогодні також немає складностей у створенні програмного засобу для інших операційних систем.

Протягом виконання кваліфікаційної дипломної роботи було вивчено великий обсяг джерел та наукової літератури, розширено знання в галузі 3D-моделювання, створення програмного засобу та його використання для навігації в будівлях і планування.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Комп'ютерна 3D-графіка. *Вікіпедія*. URL: https://uk.wikipedia.org/wiki/Комп'ютерна_3D-графіка (дата звернення: 12.04.2023).
2. Высокополигональное моделирование против низкополигонального. *3D Studio*. URL: <https://3dstudio.co/ru/low-and-high-poly-modeling/> (дата звернення: 13.04.2023).
3. Дитяча лікарня обл. *Обласне управління охорони здоров'я*. URL: <http://oblzdrav.mk.gov.ua/index.php/oblasni-94931/dityacha-likarnya-obl> (дата звернення: 10.04.2023).
4. Історія еволюції знань людства в області навігації. *Трансконтроль Україна*. URL: <https://transcontrol.com.ua/zapytannia-vidpovidi/istoriia-evoliutsii-znan-liudstva-v-oblasti-navihatsii.html> (дата звернення: 02.05.2023).
5. Пасічник В. В., Савчук В. В., Єгорова О. І. Мобільні інформаційні технології навігації користувача в приміщеннях. *Academic Journals and Conferences*. 2016. № 854. С. 119–137. URL: <https://science.lpnu.ua/sites/default/files/journal-paper/2018/jun/12974/13119-137.pdf> (дата звернення: 12.06.2023).
6. Сервіси та додатки для туристичної навігації. *Туристичний елементаль*. URL: <https://piktor.org.ua/navigate> (дата звернення: 12.05.2023).
7. Смартфон Oppo A74 4/128GB Black. *mta.ua*. URL: https://mta.ua/telefoni-ta-smartfoni/326048-smartfon-oppo-a74-4-128-gb-black?utm_source=google&utm_medium=cpc&utm_campaign=ROMAN_Pmax_TOP-product&gclid=CjwKCAjwhJukBhBPEiwAniIcNWJIPqqGj-ucLcD_nkOdd9v3IIU4FsqT_d2R6c1rRWS4lpqI4JtjxoCCqQQA_vD_BwE (дата звернення: 23.05.2023).
8. Аерофотозйомка. *Вікіпедія*. URL: <https://uk.wikipedia.org/wiki/Аерофотозйомка> (дата звернення: 20.04.2023).

9. Sketchpad. *Вікіпедія*.

URL: <https://uk.wikipedia.org/wiki/Sketchpad> (дата звернення: 09.05.2023).

10. Хаймен М. И. *Vorland C++ для "чайников"*. Киев : Диалектика; Информейшн Компьютер Энтерпрайз, 1996. 416 с. (дата звернення: 29.05.2023)

11. Харабар В. В. Підсистема визначення позиції в просторі для мобільного додатку навігації з використанням доповненої реальності : master's thesis. 2019. 112 с. (дата звернення: 02.06.2023).

12. 3ds Max Software. *3D Design, Engineering & Construction Software*. URL: <https://www.autodesk.com/products/3ds-max/overview?term=1-YEAR&tab=subscription> (date of access: 06.06.2023).

13. About – blender.org. *blender.org*. URL: <https://www.blender.org/about/> (date of access: 01.06.2023).

14. Проекция Меркатора – Википедия. *Википедия*. URL: https://ru.wikipedia.org/wiki/Проекция_Меркатора (дата звернення: 04.05.2023).

15. Indoor Maps & Wayfinding - Visioglobe. *Indoor Maps & Wayfinding - Visioglobe*. URL: <https://visioglobe.com/> (date of access: 18.04.2023).

16. Infogra. *Infogra*. URL: <https://infogra.ru/design/17-urokov-po-sozdaniyu-low-poly-illyustratsij> (date of access: 10.06.2023).

17. Primitives – Blender Manual. *Blender Documentation*. URL: <https://docs.blender.org/manual/uk/2.82/modeling/meshes/primitives.html> (date of access: 15.05.2023).

18. Renderer for architectural visualization | Chaos Corona. *Chaos Corona*. URL: <https://corona-renderer.com/> (date of access: 07.06.2023).

19. Unity - Manual: Unity User Manual 2021.3 (LTS). *Unity - Manual: Unity User Manual 2021.3 (LTS)*. URL: <https://docs.unity3d.com/Manual/index.html> (date of access: 01.06.2023).

20. V-Ray for 3ds Max. *Documentation - Chaos Help*.
URL: <https://docs.chaos.com/display/VMAX/V-Ray+for+3ds+Max+Help> (date of access: 03.06.2023).

21. Unity - Scripting API:
SceneManagement.SceneManager.LoadScene. *Unity - Manual: Unity User Manual 2021.3 (LTS)*.
URL: <https://docs.unity3d.com/ScriptReference/SceneManagement.SceneManager.LoadScene.html> (date of access: 11.06.2023).

22. Unity - Manual: Build Settings. *Unity - Manual: Unity User Manual 2021.3 (LTS)*. URL: <https://docs.unity3d.com/Manual/BuildSettings.html> (date of access: 13.06.2023).

23. Поради щодо розробки UI-дизайна | AppMaster. *AppMaster - The no-code platform for building web & mobile apps*.
URL: <https://appmaster.io/ru/blog/covety-po-razrabotke-ui-dizaina> (дата звернення: 25.05.2023).

24. Physics - Unity Manual. *Unity - Manual: Unity User Manual 2021.3 (LTS)*. URL: <https://docs.unity3d.com/ru/2018.4/Manual/PhysicsSection.html> (date of access: 13.06.2023).

ДОДАТОК А
ДОВІДКА

ДОДАТОК Б

Код застосунку

```
using UnityEngine;

public class CharacterMovement : MonoBehaviour
{
    public Transform[] waypoints;
    public float movementSpeed = 3f;
    private int currentWaypointIndex = 0;

    private Vector3 lastWaypointPosition;

    private void Start()
    {
        lastWaypointPosition = waypoints[waypoints.Length -
1].position;
    }

    private void Update()
    {
        float step = movementSpeed * Time.deltaTime;
        transform.position =
Vector3.MoveTowards(transform.position,
waypoints[currentWaypointIndex].position, step);
        if (transform.position ==
waypoints[currentWaypointIndex].position)
        {
            lastWaypointPosition =
waypoints[currentWaypointIndex].position;
            currentWaypointIndex = (currentWaypointIndex + 1)
% waypoints.Length;
        }
    }
}
```



```
public void RotateTowardsLastWaypoint()
{
    Vector3 direction = lastWaypointPosition -
transform.position;
    direction.y = 0;
    Quaternion rotation =
Quaternion.LookRotation(direction);
    transform.rotation = rotation;
}
}
```