НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Навчально-методичний комплекс "Інститут післядипломної освіти"

До захисту допущено

Завідувач кафедри

_____ О.В. Коваль
(підпис) (ініціали, прізвище)

«___»_____2020р.

ДИПЛОМНА РОБОТА

на здобуття ступеня бакалавра

з напряму підготовки 122 "Комп'ютерні науки та інформаційні технології"

на тему " Відновлення вигляду геометричного контору об'єкта за архівними документами "

Виконав (-ла): слухач (-ка) ___4___курсу, групи __ТМ-62

Гончарук Роман Володимирович _____
(прізвище, ім'я, по батькові) (підпис)

Керівник Сидоренко Юлія Всеволодівна, к.т.н., _____
(посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Рецензент к.т.н., ст.викл.каф. ТЕУТ та АЕС Рачинський А.Ю. _____
(посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Слухач _____
(підпис)

Київ – 2020

Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший , бакалаврський

зі спеціальності 122 “ Комп’ютерні науки та інформаційні технології ”

за спеціалізацією – Комп'ютерний еколого-економічний моніторинг

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ О.В. Коваль
(підпис)

«___»_____2020р.

ЗАВДАННЯ
на дипломну роботу слухачу

Гончаруку Роману Володимировичу
(прізвище, ім’я, по батькові)

1. Тема роботи “ Відновлення вигляду геометричного контору об’єкта за архівними документами “

керівник роботи Сидоренко Юлія Всеволодівна, к.т.н., доцент
(прізвище, ім’я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від ”__”____2020 р. № __

2. Строк подання слухачем роботи _____

3. Вихідні дані до роботи Форма реалізації - Програма з графічним інтерфейсом доповненої реальності, реалізована за допомогою фреймворка для створення мобільних застосунків Apache Cordova, розроблена на мові JavaScript. Інтегроване середовище розробки – Android Studio.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Задача відновлення геометричного контору об’єкта. Технології відновлення реальних об'єктів. Обгрунтування засобів реалізації системи створення додатку доповненої реальності. Опис програмної реалізації. Робота користувача с програмою.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Підходи та інструменти для створення доповненої реальності у застосуваннях. Результати порівняння інструментів за якісними характеристиками. Результати порівняння інструментів за кількісними характеристиками. Технології та засоби використані для розробки програмної системи.

6. Дата видачі завдання "10" жовтня 2019р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Вивчення та аналіз задачі	16.11.19 - 10.01.20	
2	Розробка архітектури та загальної структури системи	11.01.20 - 15.03.20	
3.	Розробка структур окремих підсистем	16.03.20 - 04.04.20	
4.	Програмна реалізація системи	05.04.20 - 15.05.20	
5.	Оформлення пояснювальної записки	16.05.20	
6.	Захист програмного продукту	20.05.20 - 29.05.20	
7.	Передзахист	09.06.20	
8.	Захист		

Студент _____ Гончарук Р.В.
(підпис) (прізвище та ініціали,)

Керівник роботи _____ Сидоренко Ю.В.
(підпис) (прізвище та ініціали,)

Анотація

до бакалаврської дипломної роботи Гончарука Романа Володимировича
на тему: «Відновлення вигляду геометричного контору об’єкта за архівними документами»

Дана дипломна робота присвячена дослідженню підходів та інструментів для відновлення реальних об’єктів у доповненій реальності.

Метою роботи є створення та реалізація мобільного додатку для відновлення геометричного контору тривимірної моделі «Десятинної церкви» у доповненій реальності.

Для програмної реалізації було вирішено використати мови програмування JavaScript. Для побудови мобільного додатку був обраний фреймворк для мобільної розробки Apache Cordova, який дозволяє використовувати додаток на більшості операційних систем та платформах.

Особливу увагу приділено можливості розширення функціоналу даної системи за рахунок створення нових розрахункових модулів на основі поточного інтерфейсу.

У роботі розглянуто способи створення додатку, в результаті чого було визначено, що прикладним програмним інтерфейсом, який забезпечує можливість створення доповненої реальності, є бібліотеки ar.js та three.js. Також було детально описано та порівняно фреймворки, які використовуються разом з даною бібліотекою для генерації та обробки тривимірної графіки у застосунках.

Загальний обсяг роботи: 76 сторінки.

Ключові слова: Доповнена реальність, AR, BP, геометричний контур, мобільний додаток, платформа Android.

Abstract

to the bachelor diploma work of Honcharuk Roman Vladimirovich
on the topic: " Restoration of the geometric office of the object according to archival documents"

This abstract is devoted to the study of approaches and tools for the restoration of real objects in augmented reality.

The aim of the work is to create and implement a mobile application for the restoration of the geometric office of the three-dimensional model of the "Tithe Church" in augmented reality.

It was decided to use JavaScript programming languages for software implementation. Apache Cordova mobile development framework was chosen to build the mobile application, which allows the application to be used on most operating systems and platforms.

Particular attention is paid to the possibility of expanding the functionality of this system by creating new calculation modules based on the current interface.

The paper considers ways to create an application, as a result of which it was determined that the application software interface, which provides the ability to create augmented reality, are libraries ar.js and three.js. The frameworks used with this library to generate and process 3D graphics in applications have also been described in detail and compared.

Total volume of work: 76 pages.

Keywords: augmented reality, AR, BP, office geometry, mobile application, Android platform.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ ТА ОДИНИЦЬ

ДР	доповнена реальність
ВР	віртуальна реальність
ОС	операційна система
AR	augmented reality
VR	virtual reality
GPS	систе́ма глоба́льного позиціонува́ння
3D	трив́имірна гра́фіка
2D	двов́имірна гра́фіка
SDK	набір засобів розробки
API	прикладний програмний інтерфейс
CLI	інтерфейс командного рядка
NPM	менеджер пакунків для мови програмування JavaScript
PC	personal computer
JS	JavaScript

ВСТУП

Питання збереження пам’ятників у нашій державі є дуже актуальним. Нажаль багато причин є того, що у нас цьому питанню приділяється мало уваги. До основних з причин можна віднести корупцію у верхніх ешелонах влади. Забудовники зносять пам'ятники архітектури і будують на його місці нову багатоповерхівку, щоб отримати більше коштів. А для цього вони заносять хабар чиновникам і безкарно руйнують культурну спадщину Києва.

Віртуальна реконструкція пам'яток культури може частково вирішити цю проблему - на базі сучасних комп'ютерних технологій, що використовують методи 3D моделювання. Туристична сфера завжди була відкрита для нових технологій. Тому коли з'явилися додатки з технологією AR, їх почали активно використовувати по всьому світу. Такий вид туризму метою якого є подорож по історичним місцям свого міста дуже полюбився туристам Західної Європи, останнім часом цей напрямок починає набирати чиності і в нашій країні. Під час таких подорожей туристи будуть мати змогу побачити пам’ятки архітектури, які або ще не були відновлені або, на жаль, не збереглися до наших часів.

Реконструкція об'єкта історико-культурної спадщини – складний науково-дослідний процес з пошуком ресурсів і аналогів, чітким алгоритмом, унікальним для кожного проекту. Розглядаючи віртуальну реконструкцію як допоміжний інструмент історичного дослідження і аналізуючи досвід вітчизняного наукового співтовариства, що займається проблематикою ВР та ДР, варто відзначити, що в цьому випадку мета відтворенні геометричного контору максимально конкретизується в рамках одного дослідження як точне відтворення об'єкта в тривимірному середовищі.

Саме тому актуальним є питання відновлення пам’ятників архітектури хоча б у режимі електронного перегляду. Але ця задача є трудомісткою. Для її реалізації необхідно розробити спеціальний додаток за допомогою библиотек та фреймворків мов программування JavaScript використовуючи інтегроване середовище розробки для мобільних застосунків Android Studio.

1. ЗАДАЧА ВІДНОВЛЕННЯ ГЕМЕТРИЧНОГО КОНТУРУ ОБ’ЄКТА

Оскільки питання про збереження історично-культурної спадщини сьогодні являється важливим у зв'язку з постійно зростаючими загрозами його існування, зумовленими браком коштів для реставрації або відновлення, виробничим освоєнням територій, на яких розташовані пам'ятники була поставлена задача створити систему відновлення вигляду геометричного контору об’єкта за архівними документами, ідея якого полягає в відтворенні геометричного контору тривимірної моделі історичної пам'ятки нашого міста «Десятинна церква», яка нажаль не зберіглась до наших днів, за допомогою AR технологій і додаванні цієї моделі у сцену додатку, яка відображується на відео з камери телефону. Особливістю даної віртуальної реконструкції є те, що «Десятина церква» була повністю втрачена і до цього дня не увійшла в список відновлених об'єктів.

Віртуальна реконструкція зруйнованих церков та храмів дозволить не тільки зберегти вигляд пам'яток культової архітектури для створення візуальних проекцій, а й зверне увагу громадськості на необхідність термінової реставрації тих храмів, які ще можна врятувати.

Мета цієї дипломної роботи - розробити мобільний додаток на мові програмування JavaScript для віртуальної реконструкції Десятиної церкви із застосуванням технологій доповненої реальності.

У даній роботі передбачається використання архівних документів та даних з інформацією про об'єкт.

Основна складність в реалізації технології доповненої реальності пов'язана з трекінгом. Трекінг - це складний процес, пов'язаний з відстеженням положення спостерігача щодо навколишнього оточення.

При реалізації доповненої реальності використовують два головних принципи її побудови:

• на основі маркера;

• на основі координат користувача

Планується для визначення місця розташування камери буде використаний GPS-датчик телефону, для визначення напрямку – компас, щоб визначати місце положення потрібно створити ексклюзивну мітку на об'єкт закріпити її на ньому для взаємодії з ним. Використовується підключення до баз даних Google.

Отримуючи ці дані, програма відтворюватиме об'єкт у вигляді елемента доповненої реальності, при наведеній камері на вказане місце на карті. Доповнена реальність реалізується за допомогою кроссбраузерної бібліотеки Three.js [1] та фреймворка A-frame [2].

При розробці мобільного додатку необхідно вирішити такі завдання:

1. Розглянути теоретичні та технологічні аспекти створення віртуальних реконструкцій;

2. Проаналізувати існуючі інтегровані середовища розробки і обґрунтувати їх використання при розробці;

3. Розробити концепцію віртуальної реконструкції Десятиної церкви;

4. Розробити мобільний додаток для графічного відображення «Десятиної церкви».

Дана розробка буде призначена для смартфонів та планшетів для використання в межах міста. Потенційними користувачами мобільного додатку можуть бути фізичні особи, які володіють мобільним телефоном, які прагнуть отримати інформацію про об'єкт історично-культурної спадщини та побачити його 3d модель в доповненій реальності[3].

2. ТЕХНОЛОГІЇ ВІДНОВЛЕННЯ РЕАЛЬНИХ ОБ’ЄКТІВ

У цьому розділі будуть розглянуті технології відновлення реальних об’єктів.

2.1 Поняття доповненої реальності

Доповнена реальність (AR) - це технологія накладення інформаційних елементів, таких як відео, зображення або супутникові дані, згенерованих програмно на зображення фізичної, реальної довкілля. Вона є перспективним напрямком IT-розробок, будучи новим способом отримання доступу до даних.

Поняття доповненої реальності пов'язано з поняттям змішаної реальності, тобто зміною сприйняття реальних об'єктів світу за допомогою програми. На відміну від поняття віртуальної реальності, що має на увазіпідміну реального оточення симулюєтья[4].

Доповнення відбувається зазвичай в реальному часі відповідно дозмінами реальних об'єктів. За допомогою просунутих технологій доповненої реальності доповну інформацію можна зробити інтерактивною. Доповнена інформація може бути як віртуальної, так і реальної - наприклад, візуалізація радіохвиль в просторі в точній відповідності з їх реальним станомв реальному світі.

Відмінность між доповненою реальністю і змішаною реальністю полягає в тому, що доповнена реальність являє собою процес розгляду віртуального і реального обєкта одночасно, в результаті чого віртуальна інформація накладається та вирівнюється інтегрується в фізичному світі, як показано на рисунку 2.1. Взаємодія доповненої реальністі знаходиться в безперервному діапазоні інтерфейсів від «реальності» до віртуальної реальності до «повного занурення» [5].

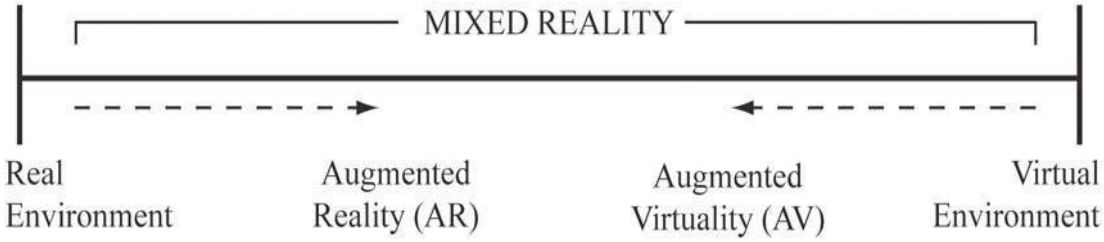

Рис 2.1 – Принцип змішаної реальності

Доповнена реальність - це технологія, яка працює по принципу доповнювати зображення об'єктами машинної графіки, та поєднувати зображення, які можуть накладатися на обчислювальні і виробничі дані і процеси і змінювати уявлення реального світу, що дає кінцевому користувачу абсолютно новий досвід [6].

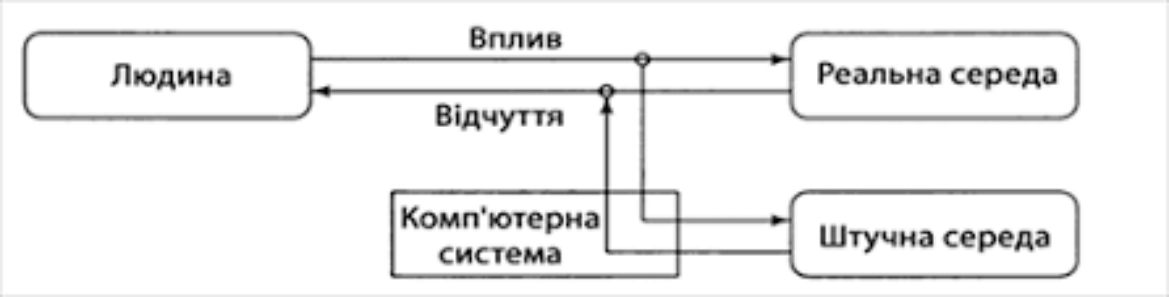

Рис 2.2 - Діапазон технологій змішаної реальності

У прикладному плані основним завданням доповненої реальності є не відділення кінцевого користувача від реального світу і занурення в якесь віртуальне оточення, а створення майданчика для інтерактивної взаємодії з об'єктом, який цікавить, як показано на рисунку 2.2. В зв'язку з цим одним з головних переваг технологій доповненої реальності єто, що за допомогою комп'ютерної бази можна виробляти взаємодія з якимсь фізичним чином в режимі реального часу. ці технології здатні зробити сприйняття інформації людиною набагато простіше і наочніше.

AR - це розпізнавання образів і відстеження маркерів [7]. Якщо додаток має розпізнавати стіл, то досить завантажити на сервер бібліотеки фотографій столів, позначити загальну структуру, колір, довільні параметри і привласнити цього набору даних певним чином впливати при виявленні на зображенні. Друга частина - це відстеження маркерів.

Рис 2.3 - Діапазон технологій змішаної реальності

На рисунку 2.3 показано як маркерами можуть виступати як спеціально надруковані зображення, так і будь-які об'єкти. Обкладинку журналу додаток розпізнає по простій формі з прямими кутами і конкретному малюнку, і буде відслідковувати її положення в просторі, відзначаючи зміщення відносно фону. У цьому випадку сама обкладинка і є маркер. Завдання системи побудувати зображення, яке може бути використане людиною для досягнення його цілей. той вид систем є більш поширеним в зв'язку з особливостями людського сприйняття - зображення для людини є більш інформативним і зрозумілим [8]. Інтерактивність може бути виражена в різному ступені. Існують системи, що дозволяють користувачеві активно змінювати віртуальне середовище. Взагалі це можуть бути системи-симулятори для будь-яких реальних дій. Як було зазначено використання можливо в разі, коли використання реальних об'єктів неможливо. то можуть бути, наприклад, медичні тренажери, які дозволять початківцям лікарям виробляти необхідні навички. Існують і інші системи, в яких користувач має можливість вибрати, які об'єкти він хоче відобразити візуально. Користувач в такому випадку може проводити геометричні перетворення, такі як лінійні афінні перетворення, переміщення та обертання. В цій групі можно разглянути наступні архітектурні системи, що дозволяють побачити, як впишеться в існуючу обстановку нова споруда або його частина, а також навігаційні і геоінформаційні системи [10]. Взагалі за типом відображення інформації в ДР розрізняються:

Аудіо. Дана система орієнтована на слухове відчуття. Взагалі така система має перевагу в використані в навігації. В наслідку цього, вони мають відтворювати спеціальні сигнали, коли людина потрапляє в вказане місце. Також розглядається можливість використання стереоскопічного ефекту, який дозволить людині при прямувані в потрібному напрямку, орієнтуватися на джерело звуку. Прикладом існуючих систем є широковідома система Hear & There [9].

Візуальні. Принцип візуального відображення полягає в зоровому сприйняті людини. Такі системи працюють з метою створити зображення, яке буде використано людиною. Взагалі зображення для людини є більш інформативним і зрозумілим, коли вид систем є більш поширеним.

Аудіовізуальні. Ця система комбінує два попередніх типа, окільки аудіоінформація в них має лише допоміжний характер.

Класифікація переглянутих типів системи розділяє сенсори не по типам реєстрованих фізичних величин, а за їх призначенням. Типи сенсорів поділяються на наступні системи на оптичні і геопозиційні. Оптичні системи обробляють зображення, отримане з камери, і мають можливість переміщатися разом з системою або незалежно від неї. Геопозиційні системи взаємодіють з сигналами систем позиціонування GPS, для визначення кута повороту щодо вертикалі і азимута використовується компас і акселерометр.

Взагалі для цих типів систем визначається різновидніть функцій системи. Можна розлянути приклад з симулятором хірургічного столу який в основі системи не повинен бути мобільним, а повинен відтворювати спеціальні умови для людини, максимально наближені до реальних. Оскільки навігаційна система має бути в основному мобільною, то її принцип полягає в тому, що вона повина переміщатися разом з людиною та транспортним засобом, яка не створює додаткових витрат на її переміщення.

Можна виділити системи додаткової реальності за ступенем мобільності. З одного боку, це стаціонарні системи, призначені для роботи в фіксованому місці; переміщення таких систем веде до часткової або повної припинення їх

функціонування [11]. З іншою боку, мобільні, які можуть бути легко переміщені і часто їх переміщення і лежить в основі виконуваної ними функції. Таким чином, вище були представлені різні підстави для класифікації різних типів систем доповненої реальності. Належність до того чи іншого типу визначається, перш за все, функціями системи, які наповнюють об'єкти якимись візуально новими властивостями, створюють можливість більш детального інформативного ознайомлення з об'єктом і дають можливість переміщатися індивіду(Туристу, екскурсанту), наприклад, разом з транспортним засобом в міському середовищі, не створюючи додаткових витрат на її переміщення.

2.2 AR пристрої

Для роботи з додатками доповненої реальності найчастіше використовуються портативні пристрої, такі як смартфони та планшети. В даний час досить смартфона або планшета, для того щоб скористатися усіма можливостями AR, при цьому досить щоб пристрій мав чотири складові:

• пристрій введення;

• дисплей;

• процесор;

• пристрій відстеження.

Оскільки системи доповненої реальності включають в себе мобільні додатки для телефонів то використання мобільних телефонів для доповненої реальності має як переваги так і недоліки. В наш час в більшісті мобільні пристрої обладнані камерами, що робить мобільний телефон найбільш зручною платформою для взаємодії або реалізації систем доповненої реальності. Більшість сучасних телефонів мають додаткові вбудовані датчики такі як: акселерометри, магнітометри і GPS-приймачі, які можуть поліпшити роботу AR програми[12].

Для мобільної реалізації доповненої реальності (AR), користувачі орієнтуються на пряме зображення, отримане з відеокамери на їх мобільному пристрої і сцени, яке вони отримують, тобто в реальному світі, доповнюються інтегрованими тривимірними віртуальними об'єктами доповненої реальності. На сьогоднішній день технологія перспективна потенціально в областях реклами, навігації, розваги, культурно-виставкова сфера і т.д.

Робота відбуваєтья по принцину такому, віртуальний об'єкт накладається на реальне зображення, а не інтегрується в нього. А для створення середовища доповненої реальності використовуються додаткові сенсори, присутні в сучасних мобільних пристроях, такі як акселерометр, компас, GPS. Користувач може переміщатися по світу доповненої реальності використовуючи інформацію про місцезнаходження. Взагалі для такої доповненої реальності необхідна додаткова інформація, така як кордони пустиря і його розміри, якщо віртуальні об'єкти мають безпосередній зв'язок з реальним світом, більшу ніж просто глобальне положення, наприклад віртуальне будівля, побудована на реальному пустирі [13]. Отримання потрібної додаткової інформації зазвичай досягається за допомогою спеціальних функцій розпізнавання або за допомогою спеціальних маркерів.

Рис. 2.4 - AR на мобільному пристрої

Такимм чином маркерами можуть бути зображення, які завчасно підготували для взаємодії, тривимірні елементарні фігури або об’єкти, які можуть розпізнаватися завдяки додатковим алгоритмам.

Перевага системи GPS в тому, що вона забезпечує простоту відстеження, незважаючи на малу точність. Для користувача головна задача спростити візуальне відстеження для більш точної оцінки його місцезнаходження і його орієнтації, внаслідок цього GPS працює в поєднанні з різними інерційних датчиками, для візуального відстеження найкращий результат досягає при низькій частоті руху, а інерційні датчики як з'ясувалося краще працюють при високій частоті руху. А вот повільний рух не дає добрих результатів через дрейф зсуву [14]. В більшості гібридних систем взаємодоповнюючий характер цих систем призводить до спільного їх використання.

Системи доповненої реальності повинні мати достатній обсяг оперативної і відео пам'яті, а також володіти потужним процесором для обробки зображень з камери. Сучасні пристрої мають достатню потужність для таких завдань, тому в даний час кожен володар смартфона може дозволити собі використання доповненої реальності.

2.3 Існуючі додатки AR

Зараз існує безліч способів інноваційного використання доповненої реальності, виділити можна чотири типи додатків, в яких використання AR технології відбуваєтья найчастіше: комерційні, розважальні та рекламні спеціалізовані та мобільне програми та освітні [15].

Комерційні та рекламні додатки

В основному доповнена реальність використовується маркетологами для просування нових продуктів. В більшості додатків спотерігається використання маркерів. Користувач розміщує перед камерою маркер, який спеціальне програмне забезпечення розпізнає і доповнює його будь-якою інформацією. Наприклад, можно розглянути як проводилася досить інноваційна реклама телеканалу National Geographic в одному з супермаркетів.

Суть цієї реклами полягає в тому, що посеред холу магазина був розміщений екран і, до нього була під’єднана камера з комп’ютером. Люди мали знаходитися поряд з

монітором перед яким знаходився маркер. Коли вони підходили до маркеру, на ньому починали з’являтися різні тварини які відображувалия на екрані. Тварини які з'являлися у доповненій реальності починали бігати серед людей, взаємодіяти з ними [16]. (Рис. 2.5)

Рис. 2.5 - Реклама від National Geographic

Одним з вирішенням проблеми в побудові та поданні макетів є доповнена реальність. Оскільки виробники стикаються з необхідністю виявляти необхідні зміни або просто продемонструвати макет і визначити, чи відповідає продукт очікуванням, при дорогому виробництві продукту. Приймається рішення про внесення змін так як зазвичай доводиться виготовляти новий прототип, що означає додаткові витрати часу і грошей [17].

В Інституті Промислової Технологій та Автоматики в Мілані проходять дослідження з AR і VR системами в якості інструменту для віртуального прототипування. Оскільки за допомогою VR і AR, в реальному часі, використовують 3D-моделювання для тестування продуктів, їх розвитку та оцінки, ITIA-CNR бере участь у дослідженнях промислових систем і додатків [18]. (Рис 2.6)

Рис 2.6 - Прототип фабрики

Освітні та розважальні додатки

Освітні та розважальні області включають в себе «культурні» додатки для огляду визначних пам'яток і відвідування музеїв. Таким чином ігрові програми являють собою ігрові додатки з додаванням AR інтерфейсів. Так само використовують AR для розважальних або освітніх цілей мобільні додатки [19].

Дійсно для реконструкції стародавніх руїн або для надання віртуальної історичної довідки існує кілька систем серед «культурних» додатків, в яких AR використовуються.

3. ОБҐРУНТУВАННЯ ЗАСОБІВ РЕАЛІЗАЦІЇ СИСТЕМИ СТВОРЕННЯ ДОДАТКУ ДОПОВНЕНОЇ РЕАЛЬНОСТІ

Для реалізації мобільного додатку було обрано мову програмування JavaScript, мова розмітки та стилю сторінок HTML і CSS відповідно. Інтегроване середовище розробки Android Studio та фреймворк мобільної розробки Apache Cordova.

3.1 Обґрунтування вибору мови програмування JavaScript

В данній роботі було обрано мову програмування JavaScript оскільки гнучкість мови дозволяє використовувати безліч шаблонів програмування стосовно до конкретних умов.

JavaScript - об'єктно-орієнтована скриптова мова програмування і є діалектом мови ECMAScript.

Мова розробки JavaScript має кілька суттєвих переваг перед класичними мовами програмування. Перш за все, це високорівнева мова програмування, що означає динамічну типізацію.

При написанні веб-додатків, програмування на JavaScript використовується найбільш часто. Якщо коротко перерахувати ключові особливості даного мови, то слід виділити наступне:

- Об'єктно-орієнтованість. Виконання програми є взаємодія об'єктів;
- Приведення типів даних проводиться автоматично;
- Функції виступають об'єктами базового класу. Ця особливість робить JavaScript схожим на багато функціональні мови програмування, такі як Lisp і Haskell;
- Автоматичне очищення пам'яті. Так звана, прибирання сміття робить JavaScript схожим на C # або Java.

Ключовою перевагою JavaScript в порівнянні з іншими мовами прикладного програмування є кроссплатформенность створених додатків. Ігри та програми, написані за допомогою JS, виглядають і працюють однаково на всіх операційних системах. JavaScript має чималу кількість готових бібліотек, які дозволяють значно спростити написання коду і нівелювати недосконалості синтаксису

На відміну від Flash, яка не підтримується розробниками мобільних ОС, додатки на HTML5 підтримуються всіма мобільними браузерами. Якщо програми, написані на Java, C #, C / C ++, Flash вимагають заздалегідь встановлених Java Runtime Environment, .NET Framework, Visual C ++ Redistributable Package або Adobe Flash Player то додатки, розроблені на HTML5, вимагають тільки сучасний Інтернет-оглядач [20].

Згідно з даними сайту www.caniuse.com, близько 96% браузерів підтримують Canvas і інші елементи HTML5.

HTML Canvas - один з нововведених в HTML5 елементів, який якраз програмується на JavaScript

Недоліки

- Необхідність забезпечувати кроссбраузерность. JavaScript виступає як інтернет-технологія. Код повинен коректно виконуватися у всіх, або хоча б найпопулярніших, браузерах;
- Система успадкування в мові викликає труднощі в розумінні того, що відбувається. В JavaScript реалізовано успадкування, засноване на прототипах. У інших об'єктно-орієнтованих мовах програмування «клас нащадок успадковує батьківський клас» але в JavaScript такими речами займаються безпосередньо об'єкти;
- Відсутня стандартна бібліотека. JavaScript не надає ніяких можливостей для роботи з файлами, потоками введення-виведення і іншими корисними речами;

- Синтаксис в цілому ускладнює розуміння.

3.2 Опис бібліотек та фреймворків JavaScript для генерації AR контенту в застосуванні

Оскільки способи створення саме контенту доповненої реальності є предметом дослідження даної роботи, аналіз здійснювався саме по відношенню до інструментів розробки, а саме бібліотек та фреймворків[21]. Для генерації геометричного контору об’єкта було обрано бібліотеку Three.js і фреймворк A-frame, оскільки є можливість оперувати з більш звичними і зручними поняттями сцени, світла і камери, об'єктами і їх матеріалами.

Бібліотека Three.js

Three.js це бібліотека JavaScript, що містить набір готових класів для створення і відображення інтерактивної комп'ютерної 3D графіки. Найбільш популярна JavaScript бібліотека для роботи з графікою, вона була також першою серед бібліотек цієї мови, яка була розширена до підтримки функцій AR [22].

Бібліотека Three.js підтримує відображення готових тривимірних моделей формату Collada (який забезпечує сумісність моделей таких програм, як Maya, 3ds Max, Blender, Unreal Engine, і т.д.)

Three.js включає такі функції:

- Створення простих і складних 3D геометрій
- Анімаційні і рухомі об'єкти в 3D-сцені
- Застосування текстур і матеріалів для ваших об'єктів
- Використання різних джерел світла для освітлення сцени
- Завантаження об'єктів, 3D-моделей і цілих сцен
- Додавання розширеної постобработки ефектів
- Робота з вашими власними шейдерами
- Створення хмари точок (спрайт)
- Геометрія: площина, куб, сфера, тор, 3D текст і інше

- Камери: перспективна або ортографічна
- Завантажники даних: двійковий, зображення, JSON і сцена

Одна з ключових особливостей Three.js те, що вона надає велику кількість функцій і API, які можна використовувати для створення 3D-сцени прямо в браузері. Three.js – це JavaScript API для візуалізації інтерактивних 2D і 3D графік, які вбудовуються всередині елемента HTML <Canvas>, як показано на рисунку 3.1 [23].

Структура сцени:

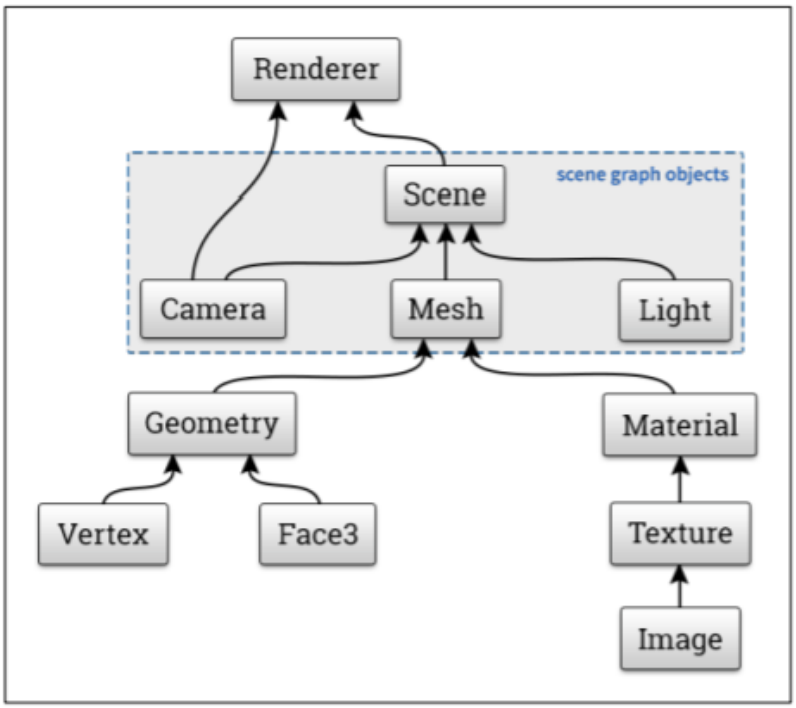

Рис.3.1

На базі даної бібліотеки було створено потужний фреймворк для розробки спеціалізованих застосунків доповненої реальності в мережі – A-Frame, який описано далі в цій роботі.

Фреймворк A-Frame

При створенні додатку на базі даного фреймворку відбувається автоматичне підвантаження бібліотеки Three.js для забезпечення підтримки браузером даного API.

Фреймворк написаний з нуля за допомогою WebGL, оскільки в основному він базується на бібліотеці Three.

HTML використовується для додавання основних елементів у A-Frame. Щоб додати до головної сцени такі елементи, як 'сцена', 'сфера' або 'камера', слід застосувати омонімічні атрибути в HTML-коді з префіксом 'a-' [24].

Однією з головних переваг A-Frame використання системи управління завантаженням файлів. Ця система дозволяє браузеру кешувати різні файли (зображення, відео, 3d моделі). A-Frame відповідає за те, щоб всі ці файли були завантажені до рендеринга.

Переваги роботи з фреймворком A-Frame:

- 3D і VR створення контенту з HTML без кроків збірки.
- Налаштування сцени з одного рядка HTML (<a-scene>) для обробки полотна візуалізації, циклу візуалізації, освітлення, блоку управління, налаштування WebVR.
- Сумісність з більшістю існуючих веб-бібліотек і фреймворків (React.js, Angular.js, D3.js, Vue.js).
- Архітектура entity component system, яка передбачає використання композиції замість наслідування для визначення складних 3D об'єктів за допомогою багаторазово використовуваних компонентів.
- Розширювана плагінова сутність-компонент екосистема.
- Візуальний інструмент інспектора, який може бути викликаний в браузері з будь-якої сцени A-Frame

Особливість фреймворка полягає в тому, що одиниці обертання в A-Frame - це градуси (не радіани як в Three.js), тому вони автоматично будуть перераховані в радіани при потряплянні в Three.js. Для визначення позитивного напряму обертання використовується правило правої руки.

Налаштувати сцени відбувається за допомогою A-frames a-scene -tag з додаванням artoolkit. SourceType повинен бути веб-камерою. З його допомогою також підтримується камера-шрифт смартфона.

3.3 Обґрунтування вибору середовища для розробки

Середа розробки Android Studio

В даній роботі для створення програми на ОС Android, можуть бути обрані різні середовища розробки, такі як, Eclipse, Embarcadero JBuilder, JDeveloper і т.д, але для роботи обрана Android Studio від компанії Google. Причиною вибору став простий спосіб підключення Google-карт, а так само зручність інтерфейсу і швидкодії програми.

Android Studio - програма, що є середовищем розробки додатків для мобільної платформи Android.

Android Studio перевершує конкурента за багатьма параметрами, до яких можна віднести:

- Гнучкість середовища розробки;
- Більший набір функцій;
- Процес розробки, який підлаштовується під розробника.

Під час створення додатків і утиліт для операційної системи Android, користувач програмного забезпечення може спостерігати за змінами в проекті, в режимі реального часу. Android Studio - універсальне середовище розробки, так як дозволяє оптимізувати роботу майбутніх додатки для роботи не тільки на смартфонах, але і на планшета, портативних ПК, які працюють на основі даної операційної системи [25].

Особливость Android Studio полягає в тому, що у програму вбудований емулятор, що дозволяє перевірити коректну роботу програми на пристроях з різними екранами, з різними співвідношеннями сторін. Відмітна особливість емулятора - перегляд приблизних показників продуктивності при запуску програми на найпопулярніших пристроях.

Також треба відмітити, в програмі реалізовані всі сучасні засоби для упаковки коду, його маркування. Локалізація додатків стає значно простіше з функцією SDK, яка також входить до переліку переваг Android Studio.

Отже основні переваги, які допомогають у розробці:

- Середовище розробки підтримує роботу з декількома мовами програмування, до яких відносяться найпопулярніші - C / C ++, Java.
- Редактор коду, з яким зручно працювати;
- Дозволяє розробляти програми не тільки для смартфонів / планшетів, а й для портативних ПК, приставок для телевізорів Android TV, пристроїв Android Wear, новомодних мобільних пристроїв з незвичайним співвідношенням сторін екрану;
- Тестування коректності роботи нових ігор, утиліт, їх продуктивності на тій чи іншій системі, відбувається безпосередньо в емуляторі;
- Рефакторинг вже готового коду;
- Досить велика бібліотека з готовими шаблонами і компонентами для розробки ПО;
- Попередня перевірка вже створеного додатки на предмет помилок в ньому;
- Великий набір засобів інструментів для тестування кожного елемента програми, ігри;

Незважаючи на наявність вбудованого Android-емулятора в самому середовищі розробки, з тестуванням додатку можуть виникнути труднощі. Так, для його запуску необхідна досить значна за продуктивністю апаратна основа ПК, на якому планується тестування.

Ще один недолік - це неможливість написати серверні проекти на мові Java для ПК, Android пристроїв.

Отже спираючись на переваги і зручності які надає Android Studio, було обрано саме її для розробки мобільного додатку на платформу Android.

Фреймворк Apache Cordova

Головною причиною обрати для розробки Apache Cordova було через можливість підключення плагінів які підтримують додавання додаткової функціональності й розширення API.

Cordova - це фреймворк, з яким можна створювати мобільні додатки з використанням веб-технологій, таких як HTML, Javascript і CSS. Використання Apache Cordova дозволяє створювати застосунки, що функціонують на широкому спектрі мобільних платформ, включаючи Tizen, webOS, Android, Apple iOS, Blackberry, Samsung Bada і Windows Phone. Оскільки код це HTML і JavaScript, то можна досить легко використовувати Javascript бібліотеки з Cordova.

Через спеціальний прошарок підтримується використання специфічних для кожної платформи функцій, а також доступ до API для взаємодії з обладнанням, телефонним стеком, адресною книгою, GPS, звуковою підсистемою, камерою та іншими компонентами мобільних платформ [26].

Під ча розробки за допомогою Cordova гібридний мобільний застосунок поєднує в собі стабільність нативного мобільного додатку та переваги веб-програми. У своїй основі гібридні застосунки — це веб додатки загорнуті в нативну оболонку.

Головною перевагою Cordova в тому, що основна структура мобільного додатка є нативною, а контент розміщений у веб застосунку, тому гібридний мобільний додаток нативний з вмонтованим інтернет браузером (WebView). Гібридний застосунок так як і нативний також має доступ до основних функції та даних смартфону чи планшету, таких як GPS, камера, пуш нотифікації, списку контактів. Може працювати офлайн без доступу до інтернету. Хоча дані не можуть поновлюватись, але користувач може використовувати раніше завантажені дані. Дані створені за час роботи офлайн зберігаються і поновлюються після відновлення інтернет-зв’язку.

3.4 Платформа Node.js

Node.js - це кросс-платформенный рушій JavaScript з відкритим початковим кодом. Призначений для розробки серверних застосувань. Він використовує Google V8 для компіляції коду JavaScript в нативний код. Node.js включає декілька модулів, в яких реалізовані типові операції для взаємодії з операційною та файловою системою, мережею і протоколами, утиліти для обробки даних. Крім того доступно безліч модулів від незалежних розробників, які можна отримати з відкритих репозиторіїв і використати у своїх проектах, як показано на рисунку 3.2 [27].

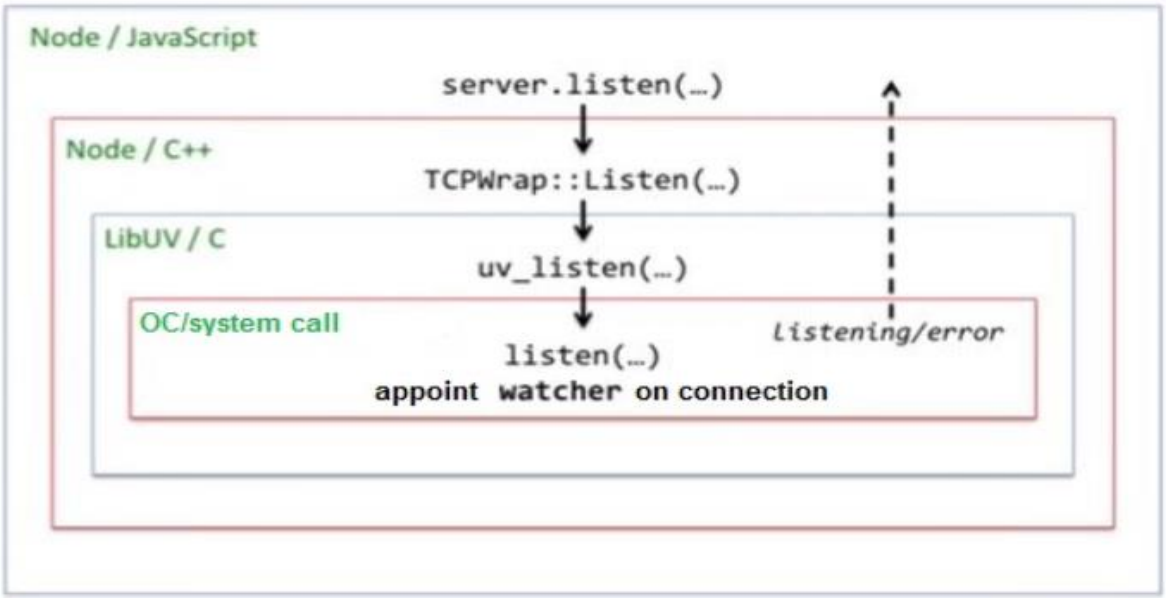

Рисунок 3.2 — Виконання системних викликів за допомого node js

Node.js грунтований на неблокуючій, керованій подіями архітектурі, яка допомагає розробникам створювати надійні розподілені застосування.

Цикл подій (Event Loop) - це те, що дозволяє Node.js виконувати неблокуючі операції введення / виводу (незважаючи на те, що JavaScript є однопоточним) шляхом вивантаження операцій в ядро системи, коли це можливо.

Оскільки більшість сучасних ядер є багато-поточними, вони можуть обробляти кілька операцій, які виконуються у фоновому режимі. Коли одна з цих операцій завершується, ядро повідомляє Node.js, що відповідна цій операції функція

зворотного виклику («коллбек») може бути додана в чергу опитування, щоб в кінцевому підсумку бути виконаною як показано на рисунку 3.3 [28].

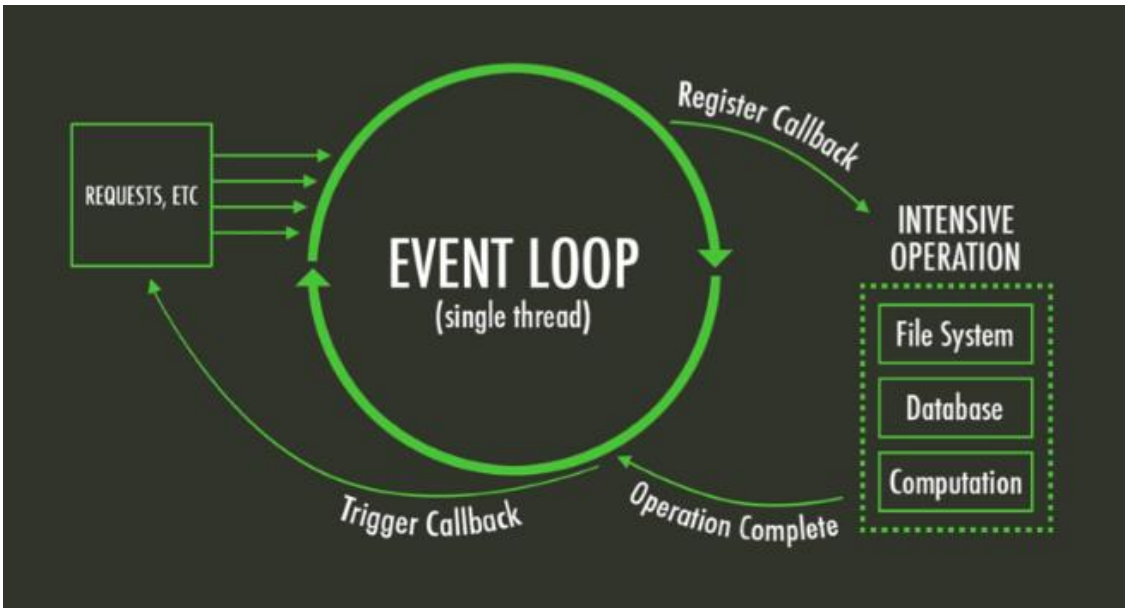

Рисунок 3.3 — Цикл подій

Коли платформа Node.js запускається, вона ініціалізує цикл подій, обробляє наданий на вхід код, який може виконувати виклики асинхронного API, налаштовувати таймери або викликати process.nextTick (). Потім починається обробка циклу подій.

Кожна фаза має FIFO (Англ. First in, first out - «першим прийшов - першим пішов») чергу коллбеков для виконання. Хоча кожна фаза є по-своєму особливою, зазвичай, коли цикл подій входить в дану фазу, вона буде виконувати будь-які операції, що відносяться до цієї фазі, а потім виконувати коллбеки в черзі цієї фази, поки черга не буде вичерпана, або максимальну кількість коллбеків не буде оброблено. Коли черга вичерпана або досягнута межа коллбеков, цикл подій переміститься на наступну фазу і так далі.

Так як будь-яка з цих операцій може планувати більше операцій, а нові події, оброблені на етапі опитування, ставляться ядром в чергу, нові події опитування можуть бути поставлені в чергу під час обробки поточних подій опитування. В результаті коллбеки, які довго виконуються можуть дозволити фазі опитування тривати набагато довше, ніж встановлений поріг таймера.

Розташована нижче діаграма (на рисунку 3.4) спрощено показує порядок виконання операцій в циклі подій.

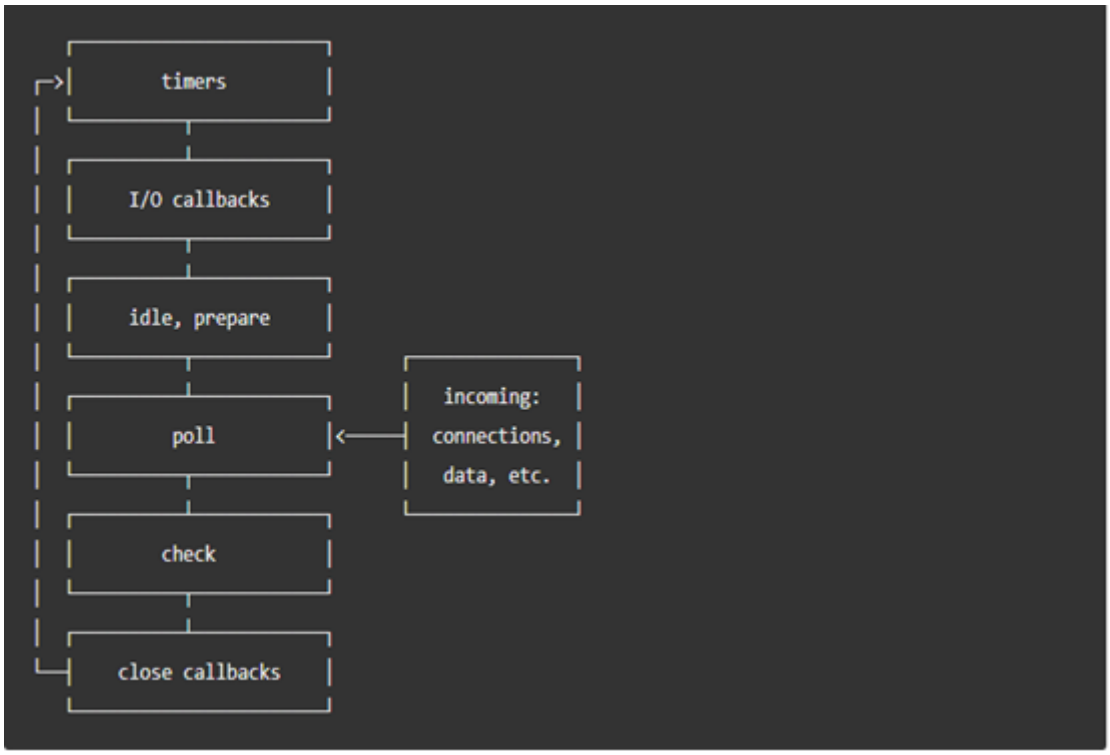

Рисунок 3.4 — порядок виконання операцій в циклі подій

Огляд фаз

- таймери: у цій фазі виконуються коллбекі, заплановані setTimeout () і setInterval ();
- I / O коллбеки: виконуються майже всі коллбеки, за винятком подій close, таймерів і setImmediate ();
- очікування, підготовка: використовується тільки для внутрішніх потреб;
- опитування: отримання нових подій введення / виведення. Node.js може блокуватися на цьому етапі;
- перевірка: коллбекі, викликані setImmediate (), викликаються на цьому етапі;
- коллбекі події close: наприклад, socket.on ('close', ...);

Між кожною ітерацією циклу подій Node.js перевіряє, чи очікується завершення будь-яких асинхронних операцій введення / виводу або таймерів, і завершує роботу, якщо їх немає [29].

4. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

У цьому розділі буде описано архітектуру та основні модулі програмного продукту.

4.1 Опис взаємодії компонентів програми

Cordova CLI, як показано на рисунку 4.1, дозволяє використовувати HTML, CSS та JavaScript, а також широкі ресурси спільноти, щоб створювати додатки для Android, Amazon Fire, iOS, Mac OS X, Windows, Windows Phone, BlackBerry, веб-браузерів тощо. Хоча веб-додатки мають обмежені можливості лише для браузера, плагінова модель Cordova забезпечує повний доступ до рідних API платформ.

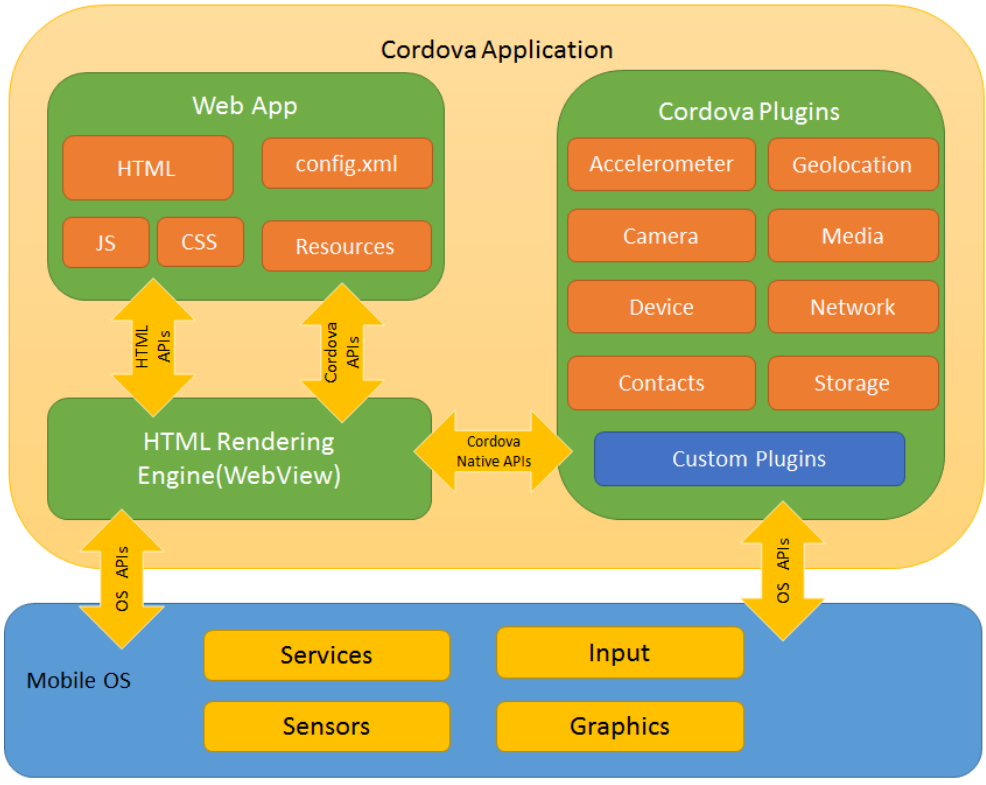

Рис 4.1 – Принцип роботи API з Cordova CLI

Таким чином, Cordova дає багатоплатформові переваги без шкоди для функціональності. Робота з Cordova, як правило, здійснюється через Менеджер пакетів вузлів (npm) та інтерфейс командного рядка Cordova (CLI). Він автоматизує процеси Cordova в Android Studio, такі як встановлення плагінів, запуску збірок та розгортання на пристрої та емулятори [30].

В ході розробки виникають проблеми із захопленням камери через два основні обмеження WebView:

1. WebView забороняє JavaScript отримувати доступ до будь-яких власних API платформи, оскільки завантажений код JavaScript з потенційно віддаленого джерела, становить загрозу безпеці.
2. WebViews зазвичай підтримують лише підмножину API HTML5 і багато власних можливостей далеко не мають відповідних стандартів HTML5.

Завдяки плагінам Cordova містить універсальний інтерфейс JavaScript, завантажений у WebView, який, у свою чергу, використовує канал зв'язку платформи для спілкування зі складеним кодом для цієї платформи і може використовувати вбудовані API, показано на рисунку 4.2.

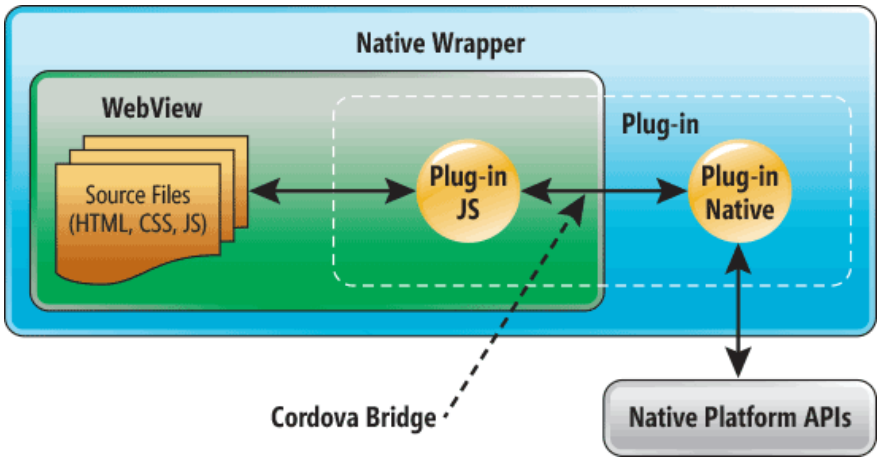

Рис 4.2 – Схема WebView

4.2 Архітектура системи

Для функціонування даної системи необхідні наступні модулі (рисунок 4.3) такі як камера, конвертер зображення, трекер, відрисовка текстури, код додатку та база маркерів.

Кожен перегляд кадрів фіксується і передається на трекер, що щабезпечує компонент камери. Щоб почати і зупинити транслювання, кадр камери автоматично конвертується в апаратно-залежний формат і задає потрібний розмір зображення для цього розробник тільки ініціалізує камеру. Придатний для рендеринга WebGL і для

відстеження перетворює кадри з камери формату у формат завдяки конвертору форматів. Це перетворення в різних дозволах доступних в перетвореному стеку кадрів також включає в себе зменшення зображення з камери.

Для того щоб виявляти і відстежувати об'єкти реального світу в рамках відеокамери використовується компонент трекеру який містить алгоритми комп'ютерного зору. Різні алгоритми піклуються про виявлення нових цілей на основі зображення з камери або маркерів. Для зручності трекер має можливість завантажити декілька наборів даних одночасно і активувати їх.

Модуль візуалізації створює зображення, що зберігається в об'єкті.

Для кожного обробленого кадру об'єкт оновлюється і викликається метод відтворення для цього у коді повинні бути ініціалізовані всі перераховані вище компоненти і виконані три необхідні умови.

- для виявлених цілей визначати об'єкти, маркер або оновлення станів цих елементів;
- забеспечити оновлення логіки програми з новими вхідними даними;
- Зручне накладення текстури зображення на фігуру.

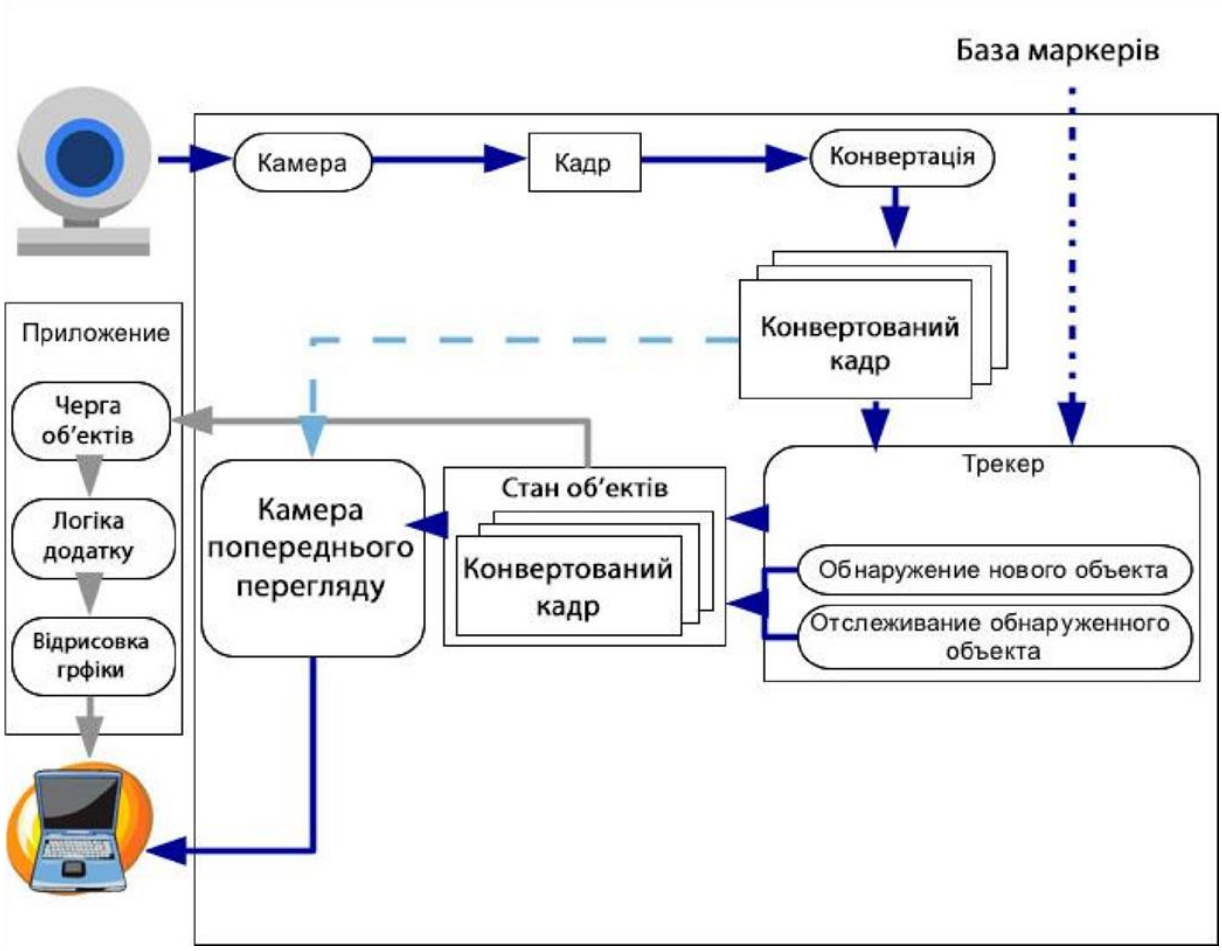

Рис 4.3 – Схема роботи системи

Програмний додаток для ОС Android складається з набору активностей, кожній з яких відповідає вікно додатку. Кожна активність представлена в проекті класом, реалізований на мові Java, що зберігається в однойменному файлі з розширенням .java. Кожній активності відповідає xml файл-опис. В xml-файлі описано у вигляді xml-коду розташування візуалізованого об'єкта [31]. При запуску активності система Android автоматично розпізнає розмір екрану мобільного пристрою і призводить виведений контент у відповідність з розміткою, описаної в xml-файлі. Таким чином, одна і та ж активність буде виглядати однаково незалежно від діагоналі використовуваного пристрою. Також, для кожного додатки Android повинен існувати xml-файл, в якому у вигляді xml-коду будуть прописані мінімальні вимоги до системи, а також активність, яка викликається при запуску додатку.

Структура системи візуалізації засобами доповненої реальності має на увазі наявність наступних базових компонентів (рис. 4.4):

1. Підсистема трекінгу, що забезпечує коректну інтеграцію віртуального об'єкта в реальне оточення.

2. Місце тривимірних моделей та іншої інформації по об'єктах.

3. Підсистема візуалізації, що забезпечує промальовування об'єктів засобами комп'ютерної графіки.

4. Графічний інтерфейс, що забезпечує взаємодію з користувачем.

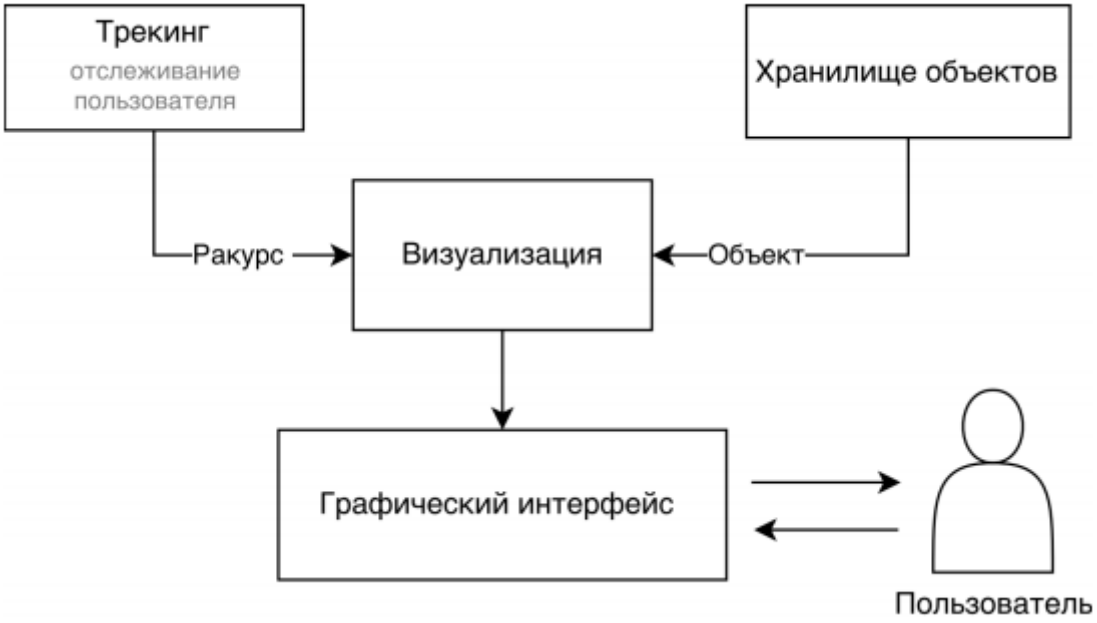

Рис 4.4 – Структура роботи системи

При оптичному трекінгу відеопотік з камери надходить в підсистему трекінгу. Проводиться обробка та аналіз кожного кадру відеопотоку на предмет наявності заданого маркера. Якщо розпізнавання маркера обчислюється матриця перетворень моделі за допомогою розтягування, повороту і перенесення. Вона дозволяє однозначно задати положення об'єкта в просторі.

За нульову точку в тривимірному середовищі може бути взята як позиція віртуальної камери, так і центр маркера. При другому варіанті дані про положені об'єкта перетворюються в дані про становище камери щодо об'єкта. В результаті,

положення віртуальної камери щодо об'єкта і реальної камери щодо маркера синхронізуються.

Тривимірний об'єкт витягується зі сховища і його положення і масштаб встановлюються на задані заздалегідь значення щодо маркера.

Потім відбувається візуалізація фінального двомірного зображення. Воно утворюється шляхом накладання віртуального об'єкта на зображення реального оточення з відеопотоку [32].

Після цього на верхньому шарі проводиться рендеринг графічного інтерфейсу користувача.

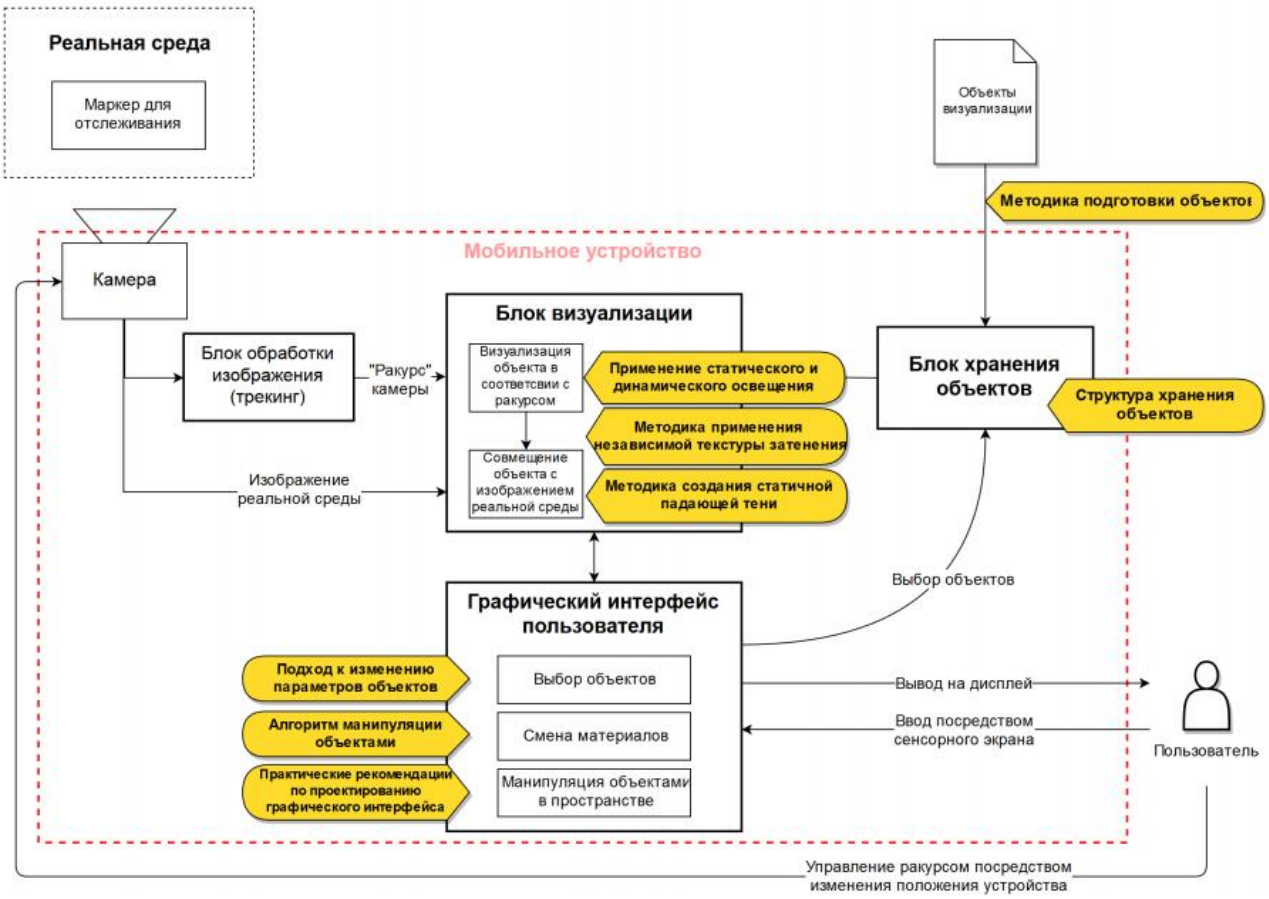

Рис 4.5 –Архітектура системи

Модифікації блоку візуалізації стосуються підвищення реалістичності без істотних витрат ресурсів. Застосування незалежної текстури затінення дозволяє

візуалізувати власні тіні об'єктів, що утворюються в внаслідок глобального освітлення, і тим самим істотно підвищити реалістичність. При цьому незалежна реалізація дозволяє використовувати одну текстуру затінення з необмеженою кількістю основних текстур будь-яких параметрів, в тому числі плиткових (тайлових) [33]. Методика реалізації статичної падаючої тіні дає можливість також без додаткових ресурсозатрат симулювати в реальному часі падаючу тінь об'єкта, утворену в внаслідок розсіяного освітлення, і тим самим візуально підвищити реалістичність вбудовування об'єкта в реальну навколишнє середовище. Використання попередньо візуалізованими статичних текстур затінення і падаючої тіні дозволяє звести використання ресурсномістких динамічних джерел освітлення до мінімуму, обмежуючись одним найменш вимогливим спрямованим джерелом, необхідним для симуляції деяких відображають матеріалів.

Відповідно до вимог по візуалізації, об'єкт перед попаданням в блок зберігання об'єктів проходить підготовку згідно з відповідною методикою. При цьому застосовується особлива структура зберігання об'єкта, що включає текстуру затінення, падаючу тінь і кілька можливих колірних рішень.

Спроектований відповідно до результатів дослідження графічний інтерфейс забезпечує взаємодію користувача з системою. З його допомогою можливо отримання інформації за доступними для візуалізації об'єктам, вибір об'єкта, що цікавить, а також можливість зміни матеріалів (Колірних рішень) об'єкта.

Сукупність засобів графічного інтерфейсу і особливої структури зберігання об'єкта створює основу для розробленого підходу до зміни параметрів візуалізуються об'єктів в реальному часі. Користувач має можливість змінювати параметри об'єкта безпосередньо в процесі візуалізації.

Також реалізований алгоритм зміни положення об'єктів з використанням жестових методів введення, що дозволяє безпосередньо переміщувати і обертати об'єкт за допомогою торкань сенсорного екрану.

4.3 Особливості проектування графічного інтерфейсу користувача при візуалізації засобами доповненої реальності

При візуалізації засобами ДР графічний інтерфейс повинен забезпечувати доступ користувачу до наступних основних завдань:

- перегляд інформації за доступними для вибору об'єктів для візуалізації і можливість вибору відповідного об'єкта,

- перегляд інформації за різними варіантами матеріалів поточного об'єкта і можливість вибору відповідного матеріалу,

Для масового застосування дуже важливими є питання зрозумілості та зручності роботи для широкого кола користувачів. Внаслідок цього, в дослідженні завжди робився наголос на даний момент, і всі питання вирішувалися з максимальною орієнтацією на користувача.

Ще однією рекомендацією при проектуванні інтерфейсів доповненої реальності, заснованої на маркерах, стала зрозуміла для користувача індикація поточного стану трекінгу.

Однак при неуспішному розпізнаванні маркера, положення об'єкта не може бути оновлено і, відповідно, об'єкт не може бути коректно відображено. В цій ситуації можливі декілька підходів (відображення об'єкта в останньому положенні, перехід на трекінг по датчикам прискорення), але частіше за все виконується приховування об'єкта до тих пір, поки маркер не буде знову розпізнаний [35].

Непідготовленому користувачеві необхідно розуміти, який стан системи в даний момент, особливо при першому самостійному знайомстві з ДР. Для цього були вироблені можливі підходи до індикації трекінгу.

Звісно ж, що для користувача найбільш зрозумілим буде графічне відображення стилізованої мішені, стилізоване зображення шуканого маркера, а також на додаток можлива реалізація анімації пошуку.

5. РОБОТА КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ

Даний програмний продукт створений з використанням потужностей бібліотеки Three.js та фреймворка A-Frame.js.

Для можливості використання функціоналу мобільного додатку користувач повинен провести установку мобільного додатка на мобільному пристрої користувача. Використовуючи мобільний додаток, користувач дозволяєте збирати, аналізувати і зберігати дані, пов’язані з наданням послуг та співробітництво в рамках мобільного додатку.

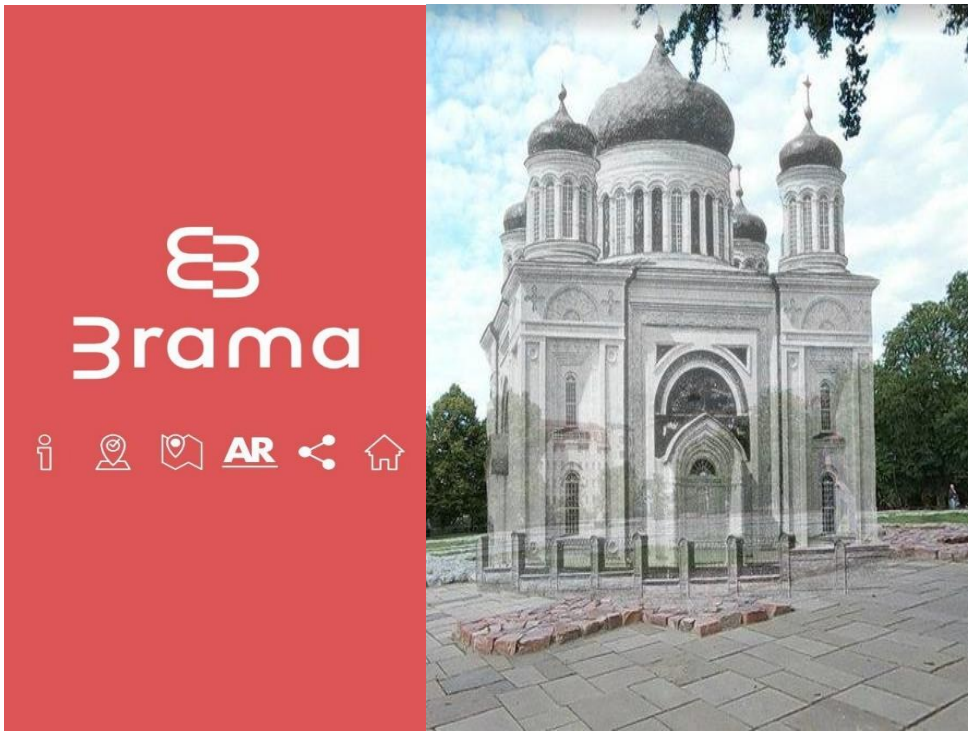

Рис 5.1 – Головна сторінка додатку

Після запуску програми і натискання на кнопку «AR» (показано на рисунку 5.1), ініціалізується запуск сцени з доповненою реальністю. Поки сцена не сгенерувалася, працює сцена з анімацією завантаження. В залежності від мітки може з'явитися певний статичний контент, або картка експоната з кнопкою. З цієї самої

кнопки починається розширення вбудованого функціоналу. При натисканні на кнопку включається режим перегляду додаткового контенту певного експоната. На сцені навколо користувача з'являтья об'єкт, який можна «знайти», спрямовуючи камеру телефону в різні боки. Можливість «оглядатися» в AR реалізована в бібліотеках three.js.

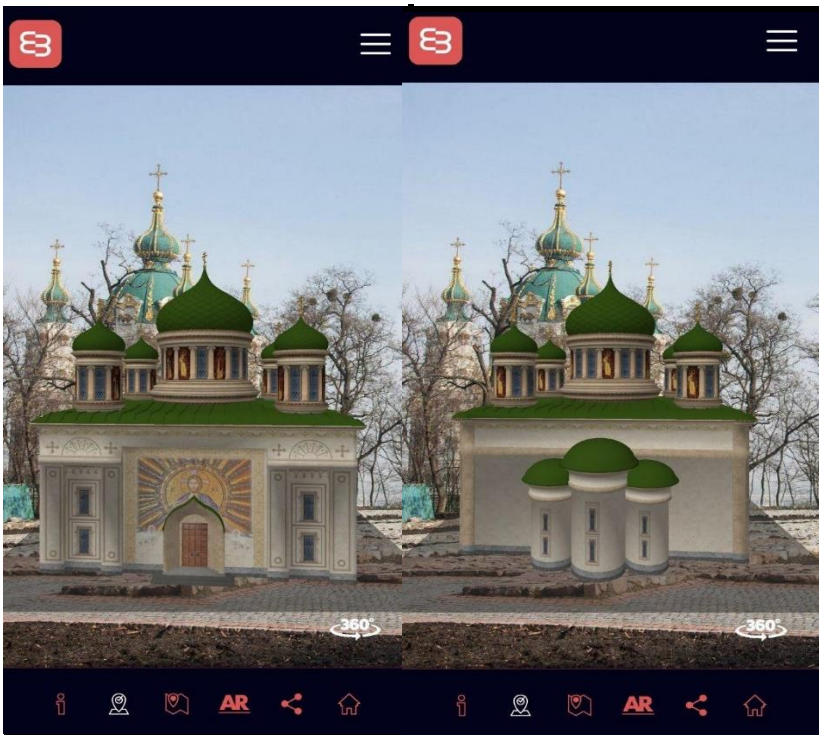

Рис 5.2 – Сторінка додатку вітворення тривірної моделі

Після ініціалізації сцени з доповненою реальністю запускається задня камера смартфона і починається процес розпізнавання, який повністю реалізований в бібліотеках Three.js показано на рисунку 5.2. Для перегляду контенту доповненої реальності через мобільний додаток необхідно навести камеру смартфона на одну з міток. Залежно від мітки в доповненої реальності можуть з'явитися фотографії, тексти або картка експоната, яка містить назву експоната і кнопку «AR». Натискання на кнопку «AR» активує режим перегляду доповненого контенту.

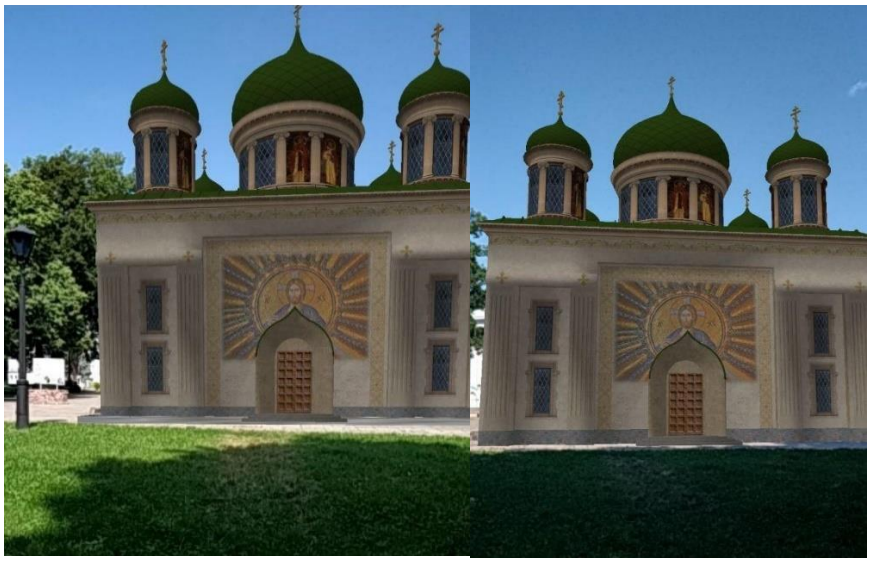

Рис 5.3 – Застосування додатку в реальному часі

У цьому режимі для перегляду додаткового контенту необхідно повертати камеру телефону в різні боки, як показано на рисунку 4.2.. Слід зазначити, що для підтримки такого функціоналу смартфон повинен володіти акселерометром і гіроскопом.

Рис 5.4 – Сторінка з інформацією про об'єкт

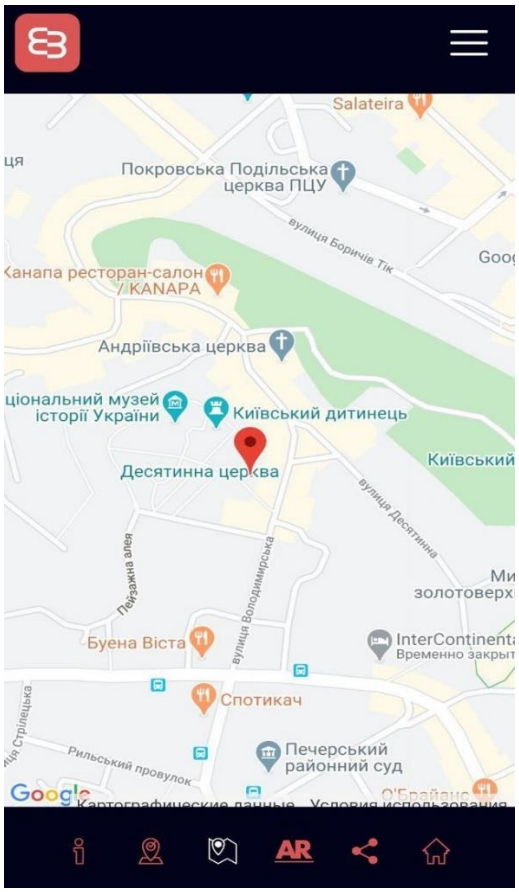

Рис 5.5 – Сторінка місцезнаходження об’єкта

Розробка додатку надасть можливість туристам не тільки пройтися по вулицями міста, відвідавши місця і об'єкти, задовольняючи почуття ностальгії, а й, завдяки технології доповненої реальності, зможуть побачити місто в історичній ретроспективі, порівняти сучасний вигляд будівлі, пам'ятника з його зображенням на старих фотографіях, маючи на руках лише смартфон або планшет.

Рис 5.6 – Схема будівлі та історичний вигляд об'єкта

При цій системі відкритий для всіх не тільки в плані використання, але і в плані поповнення: унікальність системи в тому, що кожен користувач, який має в своєму активі цікаві фотографії, може зайти на сайт додатку, завантажити фотографії і таким чином розширити доступ до спільної культурної спадщини.

Ринок пристроїв для роботи додатків доповненої реальності в даний час обмежується смартфонами, планшетами та деякими очками доповненої реальності. Ці обмеження зобов'язують підходити більш креативно і винахідливо до пошуку рішень різних проблем.

Описана в даній роботі система - спосіб показати, що технології доповненої реальності здатні органічно вписатися в музейній та туристичній практики. Більш того, при грамотному підході вони здатні перетворити музеї і вдихнути в них нове життя, атакож привернути увагу нових поколінь. Про архітектурну пам’ятку міста тепер можна дізнатися швидко і легко за допомогою мобільного телефону. В результаті роботи було розроблено мобільний додаток, а також супутні матеріали для його використання.

ВИСНОВКИ

В результаті виконання бакалаврської роботи було розроблено програмний продукт на мові програмування JavaScript для відтворення геометричного контору тривірної моделі об’єкта.

Було проаналізовано технології відтворення реальних об’єктів, проведений порівняльний аналіз кроссплатформенних фреймворків і огляд відмінностей між нативной і кроссплатформенной розробкою додатків. В роботі запропонована архітектура технології розробки кросплатформених додатків з динамічною структурою і змістом контенту.

За допомогою створеної системи вдалося відтворити об'єкт історико-культурної спадщини за допомогою AR-технологій.

Додаток було протестовано на працездатність як на стандартних емуляторах, взятих з SDK Android, так і на реальних пристроях на платформі Android (планшетному ПК і смартфоні).

Програмний продукт не поступається за функціональністю конкурентам і забезпечує зручне і просте використання. Надалі планується підтримка працездатності програмного продукту шляхом випуску оновлень, а також оптимізація вже існуючих функцій.

Підсумок всієї розробки - .apk файл, призначений для публікації мобільного застосування в Google Play.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Документація до Three.js. – Режим доступу: https://threejs.org/docs/index.html#manual/introduction/Creating-a-scene.
2. Документація до A-Frame. – Режим доступу: https://aframe.io/docs/0.5.0/introduction/.
3. Download Android Studio and SDK Tools: [Електронний ресурс] // Android Developers. Режим доступу: http://developer.android.com/intl/ru/sdk/index.html/.
4. Official Site ARCore. [Електронний ресурс] URL: https://developers.google.com/ar/discover/.
5. Occlusion in augmented reality [Електронний ресурс] URL: https://hackernoon.com/why-is-occlusion-in-augmented-reality-so-hard7bc8041607f9.
6. Амелін К. С., Гранічін О. Н., Кіяев В. І., Корявко А. В .. Введення в розробку додатків для мобільних платформ. Видавництво ВВМ.
7. Мельникова О.М .: Смартфони на Android. Видавництво Ексмо, 2013.
8. Девід Фленаган, JavaScript. Детальне керівництво // Видавництво 'Символ-Плюс', 2012 1080С.
9. Бер Бібо, Ієгуда Кац, jQuery. Докладне керівництво по просунутому JavaScript // Видавництво 'Символ-Плюс', 2011, 623С.
10. Bootstrap, URL: http://getbootstrap.com/.
11. JQueryMobile, URL: http://jquerymobile.com/.
12. Кроссплатформений мобільний додаток. Режим доступу: http://wiki.soloten.com/.
13. Кроссплатформена розробка. Режим доступу: http://apptractor.ru/develop/crossplatform-development.
14. Офіційний сайт A-Frame. – Режим доступу: https://aframe.io/.
15. Tony Parisi. Learning Virtual Reality / Tony Parisi // O'Relly Media, Inc. – 2015. – First Edition. – С. 75-97.
16. Sitepoint, Patrick Catanzariti: How to Build VR on the Web Today. – Режим доступу: https://www.sitepoint.com/how-to-build-vr-on-the-web-today/.

17. Офіційний блог Google для розробників. – Режим доступу: https://developers.google.com/web/fundamentals/vr/getting-started-withwebvr/.
18. XinReality: Timewarp. – Режим доступу: https://xinreality.com/wiki/Timewarp.
19. 3dspace: VR for Web Developers. – Режим доступу: 3dspace.com/2016/04/vrfor-web-developers/.
20. Hear&There: An Augmented Reality System of Linked Audio / Joseph Rozier, Karrie Karahalios, Judith Donath // Online Proceedings of the ICAD – Режим доступу: http://www.icad.org/websiteV2.0/Conferences/ICAD2000/ICAD2000.html.
21. Ronald T. Azuma A Survey of Augmented Reality // In Presence: Teleoperators and Virtual Environments. – 1997. – No 4. – P. 355–385.
22. Miika Tikander Development and evaluation of augmented reality audio systems: Abstract of dissertation for the degree of Doctor of Science in Technology. – Helsinki, 2009. – 70 p.
23. Global Positioning System. – Режим доступу: http://www.gps.gov/.
24. GaitAid Virtual Walker for Movement disorder patients - Режим доступу: http://www.medigait.com/index.html
25. Exploring Visuo-Haptic Mixed Reality/ Christian Sandor, Tsuyoshi Kuroki, Shinji Uchiyama, Hiroyuki Yamamoto // IEIC Technical Report (Institute of Electronics, Information and Communication Engineers). –2007. – Vol. 106, No470. – P. 31–36.
26. Visuohaptic Simulation of Bone Surgery for Training and Evaluation / Dan Morris, Christopher Sewell, Federico Barbagli // IEEE Computer Graphics and Applications. – 2006. – Vol. 26, No 6. – P. 48–57.
27. H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, K. Tachibana, ―Virtual Object Manipulation on a Table-Top AR Environment‖, ISAR'00, 111–119, 2000.
28. Bay H., Tuytelaars T., L. Van. Gool. Surf: Speed up robust features // European Conference on Computer Vision, 2006. P. 404–417.
29. Lepetit V., Fua P., Pilet J. Point Matching as a Classification Problem for Fast and Robust Object Pose Estimation // Conference on Computer Vision and Pattern Recognition, 2004.

30. Obeysekera M. Affine Reconstruction from multiple views using Singular Value Decomposition / School of Computer Science and Software Engineering, The University of Western Australia, 2003. 138 Моделирование и анализ информационных систем Т. 20, № 2 (2013) 11. Wuest H., Vial F., Stricker D. Adaptive line tracking with multiple hypotheses for augmented reality // Proceedings of the Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality, 2005. P. 62–69.
31. Офіційний сайт проекту «Відроджений Рим» (Rome Reborn). Режим доступу: http://www.romereborn.virginia.edu.
32. The Digital Sculpture Project. Режим доступа: http://www.digitalsculpture.org.
33. Steuer J. Defining Virtual Reality: Dimensions Determining Telepresence // Journal of Communications, 42.4. 1992. P. 73–93.
34. Pasko G., Pasko A., Vilbrandt C., Ikedo T. Shape Modeling Issues of Digital Preservation of Japanese Lacquer Ware and Temples. Режим доступа: http://www.cgg-journal.com/2001 –3/01/pasko1.htm.
35. Grellert M. Synagogues in Germany: a virtual reconstruction. Birkhäuser, 2004. — 159 p.

ДОДАТОК 1

Відновлення вигляду геометричного контуру об’єкта за архівними документами

Специфікація

КПІ ім. Ігоря Сікорського ТМ62214_20Б

Аркушів 2

2020

Позначення	Найменування	Примітки
Документація		
КПІ ім. Ігоря Сікорського ТМ62214_20Б 81-1	Записка.doc	Пояснювальна записка
Компоненти		
КПІ ім. Ігоря Сікорського ТМ62214_20Б 12-1	visual_AR.html	Модуль візуалізації AR-контенту
КПІ ім. Ігоря Сікорського ТМ62214_20Б 12-2	obj.obj	Модуль тривірної моделі об’єкта
КПІ ім. Ігоря Сікорського ТМ62214_20Б 12-3	style.css	Модуль стилей сторінок
КПІ ім. Ігоря Сікорського ТМ62214_20Б 12-4	calculateDistance.html	Модуль обчислення дистанції та пошук координат
КПІ ім. Ігоря Сікорського ТМ62214_20Б 13-1	Опис.docx	Опис візуалізації AR-контенту

ДОДАТОК 2

Відновлення вигляду геометричного контуру об’єкта за архівними документами

Модуль візуалізації AR-контенту

Текст програмного модуля

КПІ ім. Ігоря Сікорського ТМ62214_20Б 12-2

Аркушів 10

2020

```
<link rel="stylesheet" type="text/css" href="css/fonts.css">
<link rel="stylesheet" type="text/css" href="css/font-awesome.min.css">

  <link rel="stylesheet" href="css/bootstrap.min.css" >

  <script src="lib/ar/aframe.min.js"></script>
<script src="lib/ar/aframe-ar.min.js"></script>
<script>THREEx.ArToolkitContext.baseURL                                    =
'https://jeromeetienne.github.io/AR.js/three.js/'</script>

<link rel='stylesheet'  href='css/style.css' type='text/css' media='all' />
<script src="js/jquery.min.js"></script>
AFRAME.registerComponent('gps-position', {
    watchId: null,
    zeroCrd: null,
    crd: null,
    schema: {
        accuracy: {
            type: 'int',
            default: 100
        },
        'zero-crd-latitude': {
            type: 'number',
            default: NaN
        },
        'zero-crd-longitude': {
            type: 'number',
            default: NaN
        }
```

```
},
init: function () {
    if(!isNaN(this.data['zero-crd-latitude']) &&
    !isNaN(this.data['zero-crd-longitude'])){
            this.zeroCrd = {latitude: this.data['zero-crd-latitude'],
            longitude: this.data['zero-crd-longitude']};
            }
            this.watchId = this.watchGPS(function(position){
                this.crd = position.coords;
                this.updatePosition();
            }.bind(this));
},
    watchGPS: function (success, error) {
            if(typeof(error) == 'undefined')
                error = function(err) {
                  console.warn('ERROR('+err.code+'): '+err.message); };
            if (!("geolocation" in navigator)){
                error({code: 0, message:
                        'Geolocation is not supported by your browser'});
                return;
            }
                return
                navigator.geolocation.watchPosition(success,
                error, {enableHighAccuracy: true, maximumAge: 0, timeout:
            27000});
    },

    updatePosition: function () {
            if(this.crd.accuracy > this.data.accuracy) return;
            if(!this.zeroCrd) this.zeroCrd = this.crd;
```

```
            var p = this.el.getAttribute('position');
            p.x = this.calcMeters(
                this.zeroCrd,
                {
                    longitude: this.crd.longitude,
                    latitude: this.zeroCrd.latitude
                }
            ) * (
                this.crd.longitude > this.zeroCrd.longitude
                    ? 1 : -1
            );
            p.z = this.calcMeters(
                this.zeroCrd,
                {
                    longitude: this.zeroCrd.longitude,
                    latitude: this.crd.latitude
                }
            ) * (
                this.crd.latitude > this.zeroCrd.latitude
                    ? -1 : 1
            );

            this.el.setAttribute('position', p);

        },
        calcMeters: function(src, dest) {
            var dlon = THREE.Math.degToRad(dest.longitude - src.longitude);
            var dlat = THREE.Math.degToRad(dest.latitude - src.latitude);
            var a = (Math.sin(dlat / 2) * Math.sin(dlat / 2)) +
Math.cos(THREE.Math.degToRad(src.latitude)) *
```

```
Math.cos(THREE.Math.degToRad(dest.latitude)) * (Math.sin(dlon / 2) * Math.sin(dlon / 2));
                var angle = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
                return angle * 6378160;
        },
        remove: function() {
                if(this.watchId) navigator.geolocation.clearWatch(this.watchId);
                this.watchId = null;
        }
});


AFRAME.registerComponent('compass-rotation', {
        lookControls: null,
        lastTimestamp: 0,
        heading: null,
        schema: {
                fixTime: {
                        type: 'int',
                        default: 2000
                },
                orientationEvent: {
                        type: 'string',
                        default: 'auto'
                }
        },
        init: function () {
                if(typeof(this.el.components['look-controls']) == 'undefined') return;
                this.lookControls = this.el.components['look-controls'];
                this.handlerOrientation = this.handlerOrientation.bind(this);
```

```
                if(this.data.orientationEvent == 'auto'){
                        if('ondeviceorientationabsolute' in window){
                                this.data.orientationEvent = 'deviceorientationabsolute';
                        }else if('ondeviceorientation' in window){
                                this.data.orientationEvent = 'deviceorientation';
                        }else{
                                this.data.orientationEvent = '';
                                alert('Compass not supported');
                                return;
                        }
                }
                window.addEventListener(this.data.orientationEvent,
this.handlerOrientation, false);
                window.addEventListener('compassneedscalibration',
function(event) {
                        alert('Your compass needs calibrating! Wave your device in a
figure-eight motion');
                        event.preventDefault();
                }, true);
        },
        tick: function (time, timeDelta) {
                if(this.heading === null
                || this.lastTimestamp > (time - this.data.fixTime)) return;
                this.lastTimestamp = time;
                this.updateRotation();
        }

handlerOrientation: function (evt) {
                var heading = null;
                //console.log('device orientation event', evt);
```

```
                if(typeof(evt.webkitCompassHeading) != 'undefined'){
                        if(evt.webkitCompassAccuracy < 50){
                                heading = evt.webkitCompassHeading;
                        }else{
                                console.warn('webkitCompassAccuracy is evt.webkitCompassAccuracy');
                        }
                }else if(evt.alpha !== null){
                        if(evt.absolute === true || typeof(evt.absolute == 'undefined')){
                heading = this.calcCompassHeading(evt.alpha, evt.beta, evt.gamma);
                        }else{
                                console.warn('evt.absolute === false');
                        }
                }else{
                        console.warn('evt.alpha === null');
                }
                this.heading = heading;
        },
        updateRotation: function() {
                camera.components["look-controls"].yawObject.rotation.y = THREE.Math.degToRad(
                (
                        360
                        - camera.components["compass-rotation"].heading
                        - (
                        camera.getAttribute('rotation').y
                                -
THREE.Math.radToDeg(camera.components["look-controls"].yawObject.rotation.y)
                                )
```

```
                )
                % 360
            )
            var heading = 360 - this.heading
            var camera_rotation = this.el.getAttribute('rotation').y;
            var yaw_rotation = THREE.Math.radToDeg(this.lookControls.yawObject.rotation.y);
            var offset = ( heading - ( camera_rotation - yaw_rotation ) ) % 360;
            this.lookControls.yawObject.rotation.y = THREE.Math.degToRad(offset);
        },
        remove: function () {
            if(this.data.orientationEvent)
                window.removeEventListener(this.data.orientationEvent, this.handlerOrientation, false);
        }
    });
    AFRAME.registerComponent('gps-place', {
        cameraGpsPosition: null,
        deferredInitInterval: 0,
        schema: {
            latitude: {
                type: 'number',
                default: 0
            },
            longitude: {
                type: 'number',
                default: 0
            },
            cameraSelector: {
```

```
                type: 'string',
                default: 'a-camera, [camera]'
            }
        },
        init: function () {
            if(this.deferredInit()) return;
            this.deferredInitInterval = setInterval(this.deferredInit.bind(this),
1000);
        },
        deferredInit: function () {

            if(!this.cameraGpsPosition){
                var camera =
document.querySelector(this.data.cameraSelector);
                if(typeof(camera.components['gps-position']) == 'undefined')
return;
                this.cameraGpsPosition = camera.components['gps-position'];
            }
            if(!this.cameraGpsPosition.zeroCrd) return;
            this.updatePosition();
            clearInterval(this.deferredInitInterval);
            this.deferredInitInterval = 0;
            return true;
        },
        updatePosition: function() {
        // this.cameraGpsPosition.zeroCrd.latitude = '50.465979';
        // this.cameraGpsPosition.zeroCrd.longitude = '30.515244';
            var p = {x: 0, y: 0, z: 0};
            p.x = this.cameraGpsPosition.calcMeters(
                this.cameraGpsPosition.zeroCrd,
```

```
                {
                    longitude: this.data.longitude,
                    latitude: this.cameraGpsPosition.zeroCrd.latitude
                }
            ) * (
                this.data.longitude > this.cameraGpsPosition.zeroCrd.longitude
                    ? 1 : -1
            );
            p.z = this.cameraGpsPosition.calcMeters(
                this.cameraGpsPosition.zeroCrd,
                {
                    longitude: this.cameraGpsPosition.zeroCrd.longitude,
                    latitude: this.data.latitude
                }
            ) * (
                this.data.latitude > this.cameraGpsPosition.zeroCrd.latitude
                    ? -1 : 1
            );
            this.el.setAttribute('position', p);
        }
    });

<div style=" position: fixed; top: 10px; width:100%; text-align: center; z-index: 1; text-shadow: -1px 0 white, 0 1px white, 1px 0 white, 0 -1px white;">
        <div>coords: <span id="crd_longitude"></span>, <span id="crd_latitude"></span>
            (zero coords: <span id="zero_crd_longitude"></span>, <span id="zero_crd_latitude"></span>)
        </div>
```

```
            <div>
                camera coords: <span id="camera_p_x"></span>, <span id="camera_p_z"></span>
            </div>
            <div>
                compass heading: <span id="compass_heading"></span>,
                camera angle: <span id="camera_angle"></span>,
                yaw angle: <span id="yaw_angle"></span>
            </div>
        </div>

        <div style="width: 150px; height: 150px;">

        <a-scene embedded vr-mode-ui="enabled: false" arjs='debugUIEnabled: false;' artoolkit='sourceType: webcam;  debugUIEnabled: false;'><!--  -->

        <a-camera id="camera" user-height="1.6" gps-position compass-rotation><!--  --></a-camera>

        <a-sphere gps-place="longitude: 30.522385910181814; latitude: 50.47010926068804"></a-sphere>

       <a-obj-model
        id="model"
        src="obj.obj"
        scale="1 1 1"
        position="0 0 0"
        mtl="obj.mtl"
        radus="5"
        gps-place="latitude: 50.47010926068804; longitude: 30.522385910181814;"
```

```
            >
    </a-obj-model>

            <a-sphere gps-place="longitude: 37.469820; latitude: 55.153653"></a-sphere>

    </a-scene>

    </div
```

ДОДАТОК 3

Відновлення вигляду геометричного контуру об’єкта за архівними документами

Модуль візуалізації AR-контенту

Опис програмного модуля

КПІ ім. Ігоря Сікорського ТМ62214_20Б 13-2

Аркушів 7

2020

АНОТАЦІЯ

Розроблений програмний модуль призначений для відтворення геометричного контору тривимірної моделі об’єкта; також модуль реалізовує набір функцій, які обчислюють дистанцію місцеположення користувача від об’єкта.

Вхідними даними є електронна адреса сервера, номер користувача і дані від датчиків GPS, гіроскопа і лінійного акселерометра.

Як результат модуль відтворює геометричний контур об’єкта у сцені додатку в доповненій реальності за допомогою фреймворків та біблеотек JS.

Мова програмної реалізації — JavaSript.

Середовище мобільної розробки — Android Studio.

ЗМІСТ

1. ЗАГАЛЬНІ ВІДОМОСТІ

Назва програмного модуля — “ Модуль візуалізації AR-контенту ”. Мова програмної реалізації — JavaSript. Середовище мобільної розробки — Android Studio.

Модуль “Модуль візуалізації AR-контенту” є складовою частиною мобільного програмного продукту, що відтворює геометричний контур об’єкта та обчислює дистанцію місцеположення користувача від об’єкта використовуючи дані датчиків GPS, гіроскопа і лінійного акселерометра.

Реалізація модулю представлена системою методів. Цей модуль має єдиний інтерфейс для взаємодії для всіх методів що спрощує використання його в інших модулях програмного продукту. Для кожного методу візуалізуації необхідно передати параметри 'gps-position', 'compass-rotation', 'gps-place', 'camera' змінних.

2. ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ МОДУЛЯ

Функціональним призначенням модуля “Модуль візуалізації AR-контенту” є відтворення геометричного контору тривимірної моделі об’єкта в доповненій реальноті.

Модуль за допомогою звернення до інших методів знаходить розв’язання поставленої задачі. Він викликає виконання таких методів:

— watchGPS — повертає дані датчиків GPS;

— updatePosition — повертає дані координатів розташування;

— calcMeters — виконує обчислення дистанції;

— calcCompassHeading — виконує обчислення даних, взятих з гіроскопа і лінійного акселерометра;

— updateRotation — виконує підрахунок положення при обертані.

3. ВЗАЄМОДІЯ МОДУЛЯ З ІНШИМИ КОМПОНЕНТАМИ СИСТЕМИ

Модуль “Модуль візуалізації ” взаємодіє з модулем “Модуль тривірної моделі об’єкта” через свій інтерфейс. Цей модуль є повністю незалежним та використовується модулем візуалізації для побудови тривимірної моделі об’єкта у доповненій реальності.

Модуль працює у фоновому процесі, тож для взаємодії з ним немає потреби припиняти обробку вводу та виводу до користувача.

При закінченні візуалізації з заданими параметрами модуль створює геометричний контур у сцені додатку для взаємодії іншими модулями.

4. ВХІДНІ ТА ВИХІДНІ ДАНІ

Як вхідні дані модуль отримує дані датчиків GPS, гіроскопа і лінійного акселерометра.

Як результат модуль відтворює геометричний контур об’єкта у доповненій реальності.