

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мережних технологій факультету інформатики

ПРИВ'ЯЗКА ВІРТУАЛЬНИХ ОБ'ЄКТІВ ДО РЕАЛЬНОГО СВІТУ

**Текстова частина до курсової роботи
за спеціальністю „Інженерія програмного забезпечення” 121**

Керівник курсової роботи

канд. ф.-м. н., доц. Афонін А. О.

(прізвище та ініціали)

(підпис)

“ ____ ” _____ 2020 р.

Виконала студентка _____

_____ Карлишева А. О. _____

(прізвище та ініціали)

“ ____ ” _____ 2020 р.

Київ 2020

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем факультету інформатики

ЗАТВЕРДЖУЮ

Зав. кафедри мультимедійних систем,

доцент, канд. ф.-м.наук Жежерун О.П

(підпис)

„___” _____ 2019 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студенту Карлишевій А.О. факультету інформатики _____ 3-
го _____ курсу

Розробити Систему доповненої реальності з прив'язкою віртуальних об'єктів до реальних координат

Вихідні дані:

Зміст ТЧ до курсової роботи:

Індивідуальне завдання

Вступ

1 Основні визначення

2 Доповнена реальність

3 Розробка додатку

Висновки

Список джерел

Дата видачі „___” _____ 2019 р. Керівник _____

(підпис)

Завдання отримав _____

Календарний план виконання роботи

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1.	Отримання теми курсової роботи	02.10.2019	
2.	Пошук та збір тематичних матеріалів	25.12.2019	
3.	Ознайомлення з відповідними матеріалами та обдумування структури роботи	31.01.2020	
4.	Написання вступу та плану роботи	05.02.2020	
5.	Вивчення відповідних матеріалів, що необхідні для написання розділів роботи	13.03.2020	
6.	Вивчення відповідних матеріалів необхідних для створення додатку	22.03.2020	
7.	Безпосереднє створення додатку Здоповненої реальності	09.04.2020	
8.	Написання розділів курсової роботи	01.05.2020	
9.	Коректне оформлення роботи відповідно до вимог написання курсової роботи	04.05.2020	
10.	Подання та аналіз попередньої версії роботи з керівником	05.05.2020	
11.	Корегування роботи згідно із зауваженнями керівника.	07.05.2020	
12.	Захист курсової роботи	28.05.2020	

Студентка Карлишева А. О.

Керівник Афонін А. О.

«_____» _____

Зміст

Анотація	5
Вступ.....	6
1. Основні скорочення	10
2. Доповнена реальність	11
2.1 Взаємодія з користувачем	11
2.2 Взаємодія з оточенням.....	14
3. Розробка додатку.....	16
3.1 Компоненти доповненої реальності	16
3.1.1 Камера	17
3.1.2 Дисплей	17
3.1.3 Місцеположення	18
3.2 Прив'язка віртуальних об'єктів до реальних координат	20
3.3 Механіка додатку	24
Висновки	26
Список використаної літератури	27

Анотація

Курсова робота присвячена роботі з доповненою реальністю та методам прив'язки віртуальних об'єктів до реальних координат.

У першому розділі було досліджено види існуючих систем доповненої реальності.

Другий розділ присвячено розробці додатку на основі ARCore для мобільного пристрою з операційною системою Android. Були коротко описані аспекти роботи з ARCore та компоненти додатку, а окрім цього, було детально розглянуто роботу з місцеположеннями.

Ключові слова: доповнена реальність, місцеположення, GPS, координати, AR.

Вступ

З кожним роком технології прямують все далі й далі у своєму розвитку. З'являються нові технології та розвиваються уже існуючі. Популярності поступово набуває Віртуальна Реальність (VR), а пліч-о-пліч з нею – Доповнена Реальність (AR).

Уже сьогодні можна опинитися в іншій реальності завдяки VR-клубу, а додаток на смартфоні дозволить побачити більше, ніж дозволяють очі людини. Маркет доповненої реальності стрімко розвивається.

На масове поширення AR технологій значно вплинув розвиток «заліза». Мобільні пристрої тепер досить потужні, щоб використовувати AR застосунки. Окрім того, розвиток характеристик камер мобільних пристроїв також дозволяє покращити роботу AR додатків. Безперечно, технологічний розвиток мобільних девайсів відіграв важливу роль, але не тільки це стало ключовим фактором поширення AR.

По-перше, контент таких застосунків тепер має більше смислове навантаження. AR технології використовуються не тільки у розважальних додатках, а й у дійсно корисних застосунках. Наприклад, за допомогою додатка ARuler можливо виміряти довжину будь-якої поверхні, що знаходиться поруч з вами. Одночасно, технології дозволяють досягнути досить високої точності.

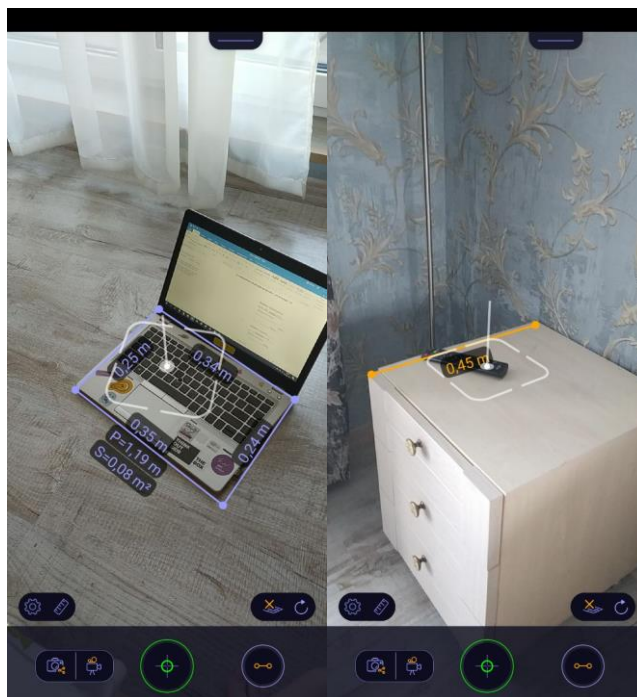


Рисунок – робота додатку ARRuler

По-друге, інтерактивність теж розширюється. Наприклад, у грі Pokemon GO користувач напряду взаємодіє з AR об'єктами через екран та камеру. Окрім цього, збільшується реалістичність додатків – в першу чергу через те, що збільшується потужність девайсів.



Рисунок– фото роботи Pokemon GO [3]

По-третє, на популярність досить сильно вплинуло те, що компанії-гіганти, такі як, наприклад, Facebook, інтегрували елементи AR у свої продукти. Зараз кожному користувачу смартфона доступні для використання так звані «маски» в Instagram. За допомогою інтерфейсу камери вбудованого у додаток, користувачі можуть використовувати уже існуючі маски для того,

щоб змінити риси обличчя, додати прикраси чи макіяж, або ж розмістити на своїй фотографії чи відео AR-анімацію. Окрім цього, Facebook надає відкритий API для створення власних масок.



Рисунок – AR маска з інстаграму

Окрім багатьом відомої доповненої реальності, що базується на розміщенні AR-об'єктів відносно безпосереднього оточення користувача, базуючись на даних та маркерах, отриманих через камеру та орієнтації пристрою у просторі, можливо створювати системи AR, що базуються також і на геопозиціях, так звані безмаркерні або координатно-орієнтовані системи доповненої реальності. Такі системи більше направлені на визначення не тільки орієнтації пристрою та користувача у просторі, а ще й їхнього географічного місцезнаходження, а розміщення об'єктів подібної доповненої реальності буде значним чином залежати від обрахованих даних.

В системі даного типу відображення об'єктів базується на координатах та орієнтації користувача. В залежності від місцезнаходження, система по-різному відображатиме дані, наприклад, додаток, що визначає назви вершин гір, що видно через камеру пристрою та виводить на екран відповідне зображення з камери, але з підписаними вершинами. Застосунок працює у реальному часі.

Відповідно, користувачі, що знаходяться на різних координатах, будуть бачити різні вершини, а отже й додаток відображатиме різну інформацію для кожного з них. Якщо ж користувачі знаходяться у одній і тій самій точці простору, то отримані з системи доповненої реальності дані та їхнє відображення будуть ідентичними.



Рисунок– фото роботи ViewRanger [4]

Для реалізації подібної системи, необхідна не тільки доповнена реальність, а й безпосередньо робота з координатами користувача. Детальніше це буде розглянуто нижче.

1. Основні скорочення

Доповнена реальність (ДР, Augmented Reality, AR) – система, у якій віртуальні об'єкти проєктуються у реальний світ. Наприклад, за допомогою камери, що в реальному часі переносить отримане зображення, на яке накладаються віртуальні об'єкти, на екран.

Global Positioning System (GPS) – сервіс, що дозволяє отримати місцеположення пристрою за допомогою супутників. Має високий рівень інтеграції у мобільні пристрої.

Окрім того, GPS може використовуватися разом із технологією Wi-Fi. Завдяки аналізу MAC-адрес доступних Wi-Fi мереж підвищується точність визначення координат.

Геолокація – відрізняється від GPS тим, що визначає місцеположення за конкретною адресою, замість координат.

Plain Old Java Object (POJO) – простий Java об'єкт, що не наслідується від якогось специфічного об'єкта.

SLAM, simultaneous localization and mapping, одночасна локалізація та картографування – метод, що використовується для побудови та оновлення мапи невідомого простору.

2. Доповнена реальність

2.1 Взаємодія з користувачем

Системи доповненої реальності мають різні ступені взаємодії з користувачем. У деяких система участь користувача мінімальна, навіть пасивна. У деяких система реагує тільки на зовнішні фактори – наприклад, розміщення об'єктів в реальному світі. Інші системи потребують прямої участі користувача – активної взаємодії з об'єктами доповненої реальності або ж введення деяких даних. Бувають системи що працюють безпосередньо в залежності від користувача – наприклад, якщо системою можливо керувати за допомогою жестів.

В залежності від рівня втручання користувача системи поділяються на два типи: автономні та інтерактивні.

Автономні системи не вимагають участі користувача, їхнє основне завдання – відображення інформації. Прикладом подібної системи може бути додаток, що генерує трьохвимірний об'єкт на місці певного QR-коду або іншого маркерного зображення, що відображає певну інформацію, пов'язану з цим маркером.

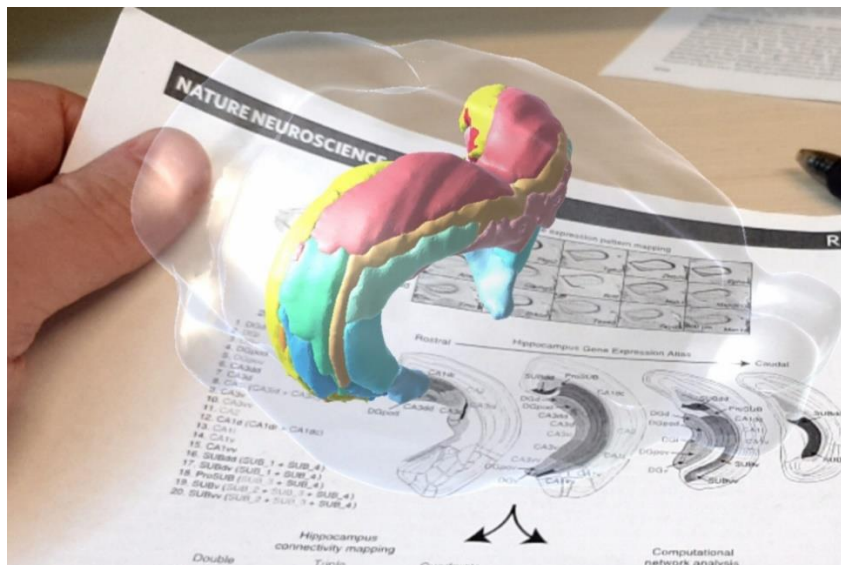


Рисунок – відображення інформації за допомогою AR [5]

Інтерактивні системи вимагають участі користувача для функціонування. Користувач може, наприклад, визначати розміщення об'єктів віртуальної реальності або безпосередньо взаємодіяти з ними. Приміром, додаток Pokemon GO дозволяє користувачам обрати опцію відображення так званих покемонів за допомогою доповненої реальності, а потім користувач може «ловити» їх через екран мобільного пристрою.



Рисунок – приклад використання інтерактивної системи доповненої реальності у серіалі «The Good Doctor»

Користувач може взаємодіяти з інтерфейсом програми та камерою, але не мусить мати доступ до внутрішніх складових, таких як набір інструментів доповненої реальності чи GPS.

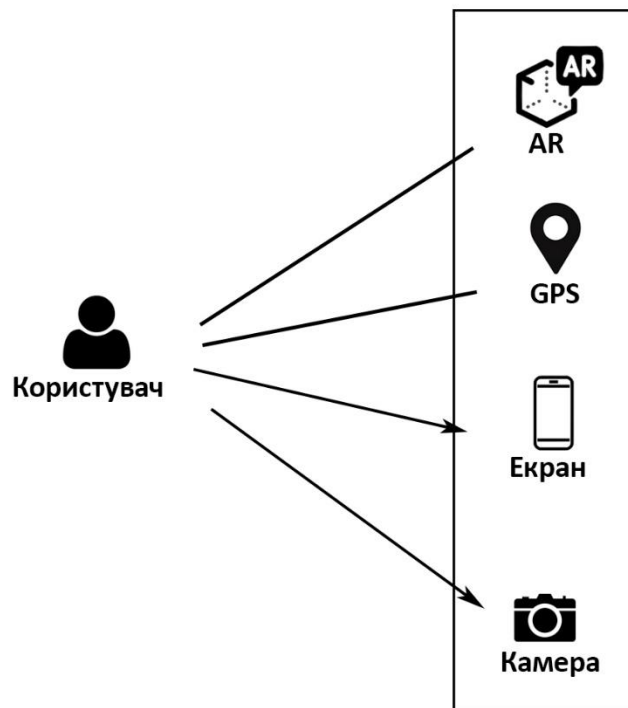


Рисунок – доступ користувача до компонентів додатку

Відповідно, аугментація та подібні процеси відбуваються без втручання користувача. Окремо слід зазначити, що місцезположення користувача теж має оброблюватися засобами додатку.



Рисунок – приклад взаємодії користувача та доповненої реальності. Кадр з серіалу «The Good Doctor»

2.2 Взаємодія з оточенням

Основним інструментом розробки додатку було обрано ARCore. Отже, пошук необхідних площин, на яких можна розмістити віртуальні об'єкти та відслідковування отриманих маркерів виконуються засобами цієї бібліотеки.

Для трекінгу переміщень мобільного девайсу ARCore використовує процес, що називається SLAM (Simultaneous localization and mapping), що означає одночасну локалізацію та картографування. За допомогою характерних точок на отриманому з камери зображенні скомбінованих з внутрішніми показниками, такими як, наприклад, позиція та орієнтація мобільного девайсу, розраховується зміна в місцеположенні мобільного пристрою та камери відносно реального світу.

ARCore шукає кластери характерних точок, що належать горизонтальній чи вертикальній поверхні, наприклад, стіл, підлога, стіни, та робить ці поверхні доступними для додатку як площини (planes).



Рисунок – візуальне відображення площин у додатку

Через те, що бібліотека орієнтується на характерні, особливі точки для визначення площин, рівні поверхні без помітної текстури можуть бути не визначені належним чином. Приміром, можуть бути проблеми з визначення білої стіни.

Окрім цього, ARCore може відслідковувати інформацію щодо освітлення, щоб віртуальні об'єкти були відображені з такими самими умовами, що й оточуючі об'єкти реального світу.

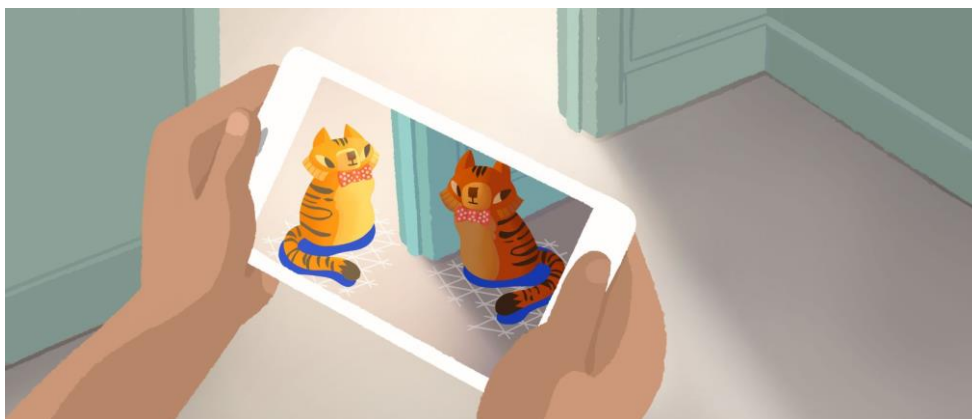


Рисунок – вплив реального освітлення на віртуальні об'єкти.

Використовуються спеціальні типи об'єктів, що називаються trackable (відслідковувані) для збереження даних щодо площин. Як натякає назва, ARCore відслідковує положення подібних об'єктів. Для розміщення віртуальних об'єктів створюється якір на відслідковуваній площині для забезпечення зв'язку між пристроєм та віртуальним об'єктом, навіть коли пристрій переміщується у просторі.

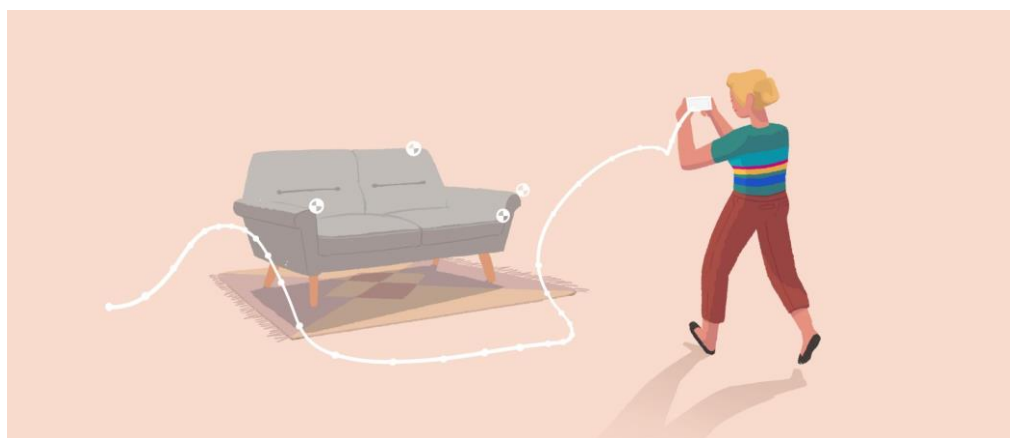


Рисунок – відслідковування характерних точок при переміщенні пристрою

3. Розробка додатку

Як основу для додатку було взято офіційний приклад AR застосунку з відкритого репозиторію ARCore. Весь код у репозиторії розповсюджується під ліцензією Apache-2.0, а отже, дозволяється використовувати цей код для модифікації та редистрибуції.

3.1 Компоненти доповненої реальності

Для роботи додатку необхідно, щоб декілька різних компонентів застосунку працювали злагоджено та одночасно. Серед них: пристрої вводу та виводу, пристрій, що відповідає за обробку даних. Серед пристроїв вводу можуть бути різноманітні датчики, наприклад, камера, GPS, гіроскоп, компас; у додатках, призначених для мобільних девайсів, може використовуватися ще й сенсорний дисплей. Окрім того, пристрої вводу можуть варіюватися в залежності від вимог системи доповненої реальності.

У якості пристроїв виводу у системах доповненої реальності може використовуватися, приміром, дисплей мобільного пристрою, або ж інші види дисплеїв. В окуляри може бути вбудовано так званий оптикопрозорий дисплей, на який проєктуються об'єкти доповненої реальності та, у такому випадку, немає необхідності накладати зображення на трансляцію відео.

На рисунку вище зображено зв'язок між компонентами застосунку під час його роботи та від моменту запуску.

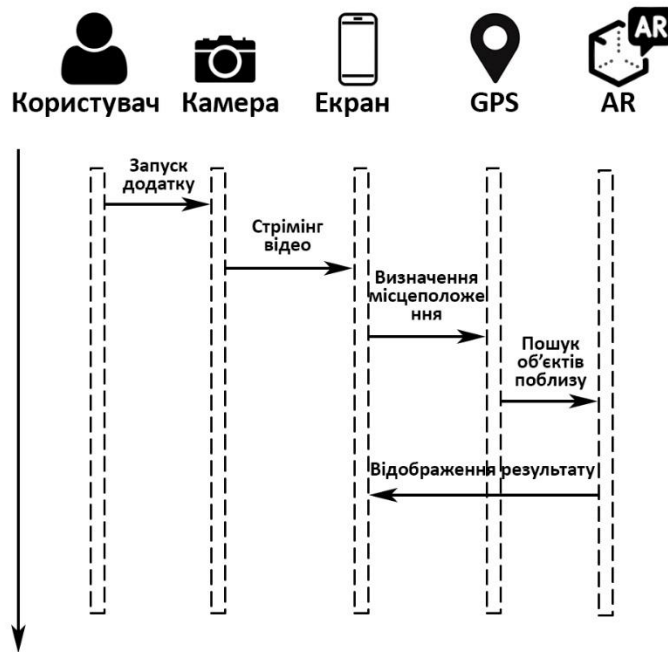


Рисунок – взаємодія компонентів додатку

Розглянемо детальніше роботу компонентів, що використовуються у роботі застосунку.

3.1.1 Камера

Для користувача, камера у додатку відповідає за збір інформації та представлення зображення реального світу на екрані мобільного пристрою. Необхідно, щоб був безперервний вивід потоку зображень, отриманих з камери, на екран.

Для роботи з камерою необхідно отримати дозвіл користувача. Механізм роботи з дозволом камери реалізовано у класі `CameraPermissionHelper.java`.

У випадку, якщо дозвіл не отримано, додаток не може продовжувати роботу, так як взаємодія з доповненою реальністю повністю базується на вхідних даних, отриманих з камери.

3.1.2 Дисплей

Дисплей слугує посередником між реальним світом та доповненою реальністю, накладаючи зображення віртуальних об'єктів на реальний світ. Фактично, екран відображає потік відео, на кожен кадр якого було додано зображення, що відповідає AR-об'єкту.

У додатку дисплей виконував роль пристрою виводу, без функції пристрою вводу, тому що користувач, конкретно у цьому додатку, не має можливості вводити дані до системи. Усі необхідні дані для роботи додатку збиралися та оброблювалися автоматично.

Окрім вищезазначеного, дисплей надає користувачеві повідомлення щодо роботи додатку, подібні до «ведеться пошук поверхні» чи «не вистачає освітлення».

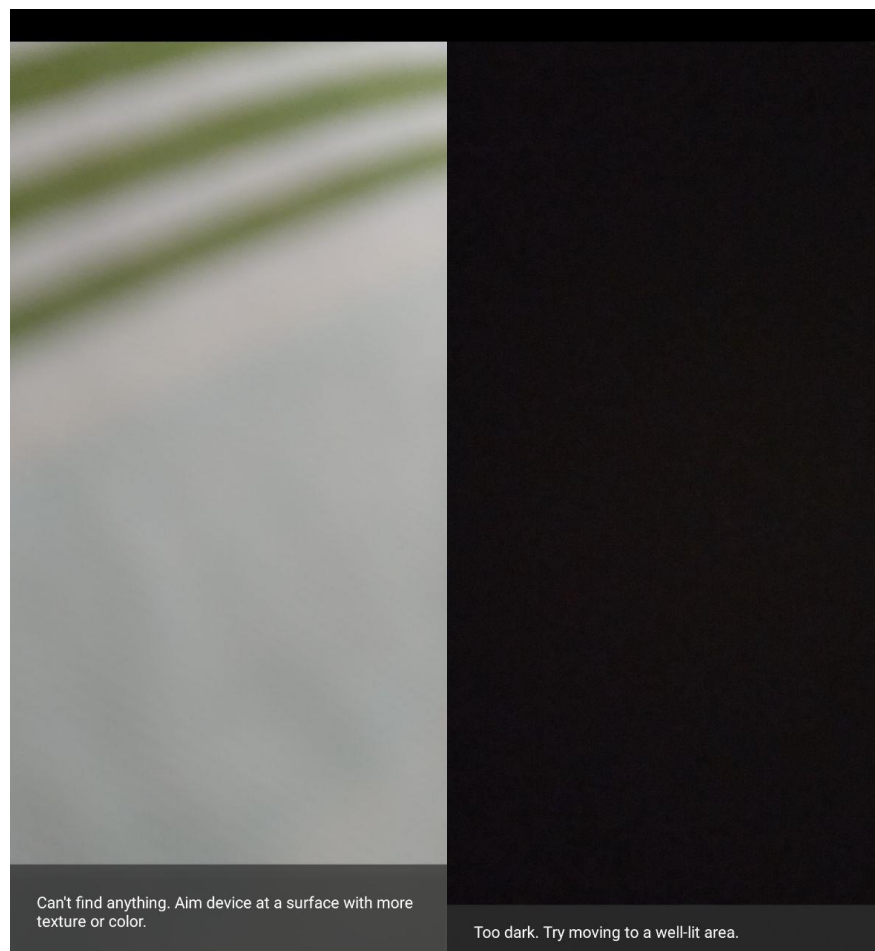


Рисунок – приклади повідомлень системи

3.1.3 Місцезаположення

Для правильної та безперебійної роботи додатку користувач має надати дозвіл на доступ до місцезаположення пристрою.

Для цього, необхідно прописати у файлі `AndroidManifest.xml`, що відповідає за опис складових компонентів додатку, `uses permission`:

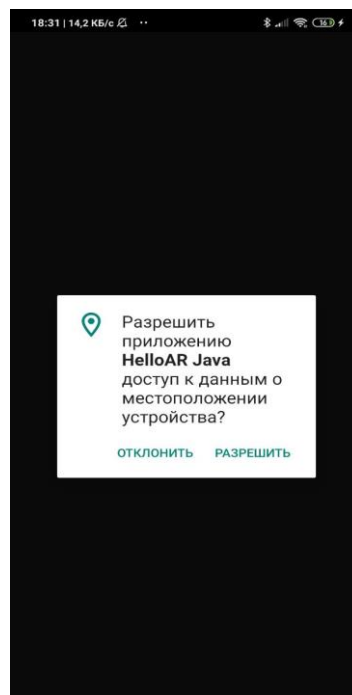
```
<uses-permission
```

```
android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

Після цього, під час запуску додатку, після створення й запуску основного вікна, необхідно виконати перевірку, чи має додаток зазначений вище дозвіл, і якщо його не надано, то запросити користувача надати доступ.

```
if (!hasPermission(Manifest.permission.ACCESS_FINE_LOCATION)){  
    requestPermissions(LOCATION_PERMS, LOCATION_REQUEST);  
}
```

Метод `requestPermissions` – вбудований. Приймає на вхід перелік необхідних дозволів (у даному випадку `LOCATION_PERMS` містить тільки `Manifest.permissions.ACCESS_FINE_LOCATION`) та код цього переліку, щоб потім можна було обробити, які саме дозволи були надані.



Рисунок– скрін запиту на доступ до місцеположення

Для реалізації роботи з координатами було використано відкриту бібліотеку `SimpleLocation` реалізовану під ліцензією `Apache-2.0`. Ця бібліотека додає рівень абстракції над вже існуючими інструментами `Android` для роботи з локаціями та координатами.

Для використання бібліотеки необхідно додати відомості про неї у кореневий та проектний файли `build.gradle`, що містять інформацію щодо збірки проекту.

У кореневий файл додаємо:

```
repositories{
maven { url "https://jitpack.io" }

}
```

Це підтягне бібліотеку із зовнішнього ресурсу, а код у проектному файлі

```
dependencies{
implementation 'com.github.delight-im:Android-SimpleLocation:v1.0.1'

}
```

Дозволяє використовувати бібліотеку у коді.

Для використання самої бібліотеки у методі створення основного вікна додатка створюється об'єкт `SimpleLocation`, що буде надалі відповідати за роботу з місцеположенням.

У додатку було використано лише деякі компоненти бібліотеки – власне, отримання координат користувача, збереження місцеположення віртуального об'єкту у класі `Point`, що належить до бібліотеки, та розрахування відстані між віртуальним об'єктом та користувачем.

Для прив'язки віртуальних об'єктів до координат було реалізовано допоміжний клас, що буде описано в підрозділі нижче.

3.2 Прив'язка віртуальних об'єктів до реальних координат

Досягти прив'язки можна за допомогою різних інструментів, використовуючи як вже існуючі в `Android SDK`, так і ті, що знаходяться у вільному доступі (так звані `open-source` бібліотеки).

Однак, вибір готових `open source` інструментів доволі обмежений, а часто, вони не відповідають поставленим вимогам. Приміром, `GART` (`Geo`

Augmented Reality Toolkit) призначений для розробки програмних застосунків лише для Windows Phone, а отже, не підходить для системи Android.

Тому використовувалися вбудовані засоби Java. Створений POJO об'єкт містить інформацію про назву файлів для рендерингу віртуального об'єкта, місцеположення – довжину та ширину – об'єкта та інші необхідні дані.

```
public class LocationObject {  
    private String object;  
    private String shade;  
    private String content="";  
    private SimpleLocation.Point location;  
    ...  
}
```

Клас також містить стандартні гетери, сетери, та кілька конструкторів з різною кількістю параметрів. Для збереження інформації про об'єкти, створено клас KnownLocationHelper.java, його роль – збереження списку об'єктів, що створюються прямо у коді, та кілька допоміжних методів для роботи з відстанню до цих об'єктів.

```
public class KnownPositionsHelper {  
    private ArrayList<LocationObject> locations;  
    public KnownPositionsHelper() {  
        locations = new ArrayList<>();  
        locations.add(new LocationObject("stop.obj", "stop_shade.png", new  
SimpleLocation.Point(50.392060, 30.470689)));  
        ...  
    }  
}
```

```
public boolean hasObjectNearby(SimpleLocation.Point location){  
    for (LocationObject lo : locations) {  
        if (SimpleLocation.calculateDistance(lo.getLocation(), location)<=10)
```

```

        return true;
    }
    return false;
}

public LocationObject getNearbyObject(SimpleLocation.Point location){
    for (LocationObject lo : locations) {
        if (SimpleLocation.calculateDistance(lo.getLocation(), location)<=10)
            return lo;
    }
    return null;
}
}

```

Фактично, цей клас замінює собою роботу з базою даних. Усі необхідні об'єкти внесені у код вручну та створюються під час запуску додатку. Координати для об'єктів були обрані таким чином, щоб вони знаходилися досить близько для тестування додатку, а згодом було додано ще декілька об'єктів розташованих у місті.

Логіка відображення необхідних об'єктів та безпосередня робота з координатами користувача була реалізована в основному робочому класі додатка MainActivity.java.

В методі onDrawFrame(), що викликається додатком кожного разу, коли змінюється зображення на екрані,

```

if (GeolocationPermissionHelper.hasLocationPermission(this)) {
    if (!knownPositionsHelper.hasObjectNearby(simpleLocation.getPosition())) {
        if (anchors.size()>0)
            anchors.clear();
    } else {
        if (anchors.size() != 0) {
            Plane tracked = null;

```

```

for (Plane plane : session.getAllTrackables(Plane.class)) {
    if (plane.getTrackingState() == TrackingState.TRACKING) {
        tracked = plane;
        break;
    }
}
if (hasTrackingPlane()) {
    anchors.add(new
ColoredAnchor(tracked.createAnchor(tracked.getCenterPose()),
knownPositionsHelper.getNearbyObject(simpleLocation.getPosition()), new
float[]{66.0f, 133.0f, 244.0f, 255.0f}));
}
}
}
}

```

У цьому коді перевіряється, чи є поруч віртуальний об'єкт. Якщо є – об'єкт додається до відображення доповненої реальності. Якщо поруч немає об'єктів, список об'єктів, які система відстежує спустошується.

3.3 Механіка додатку

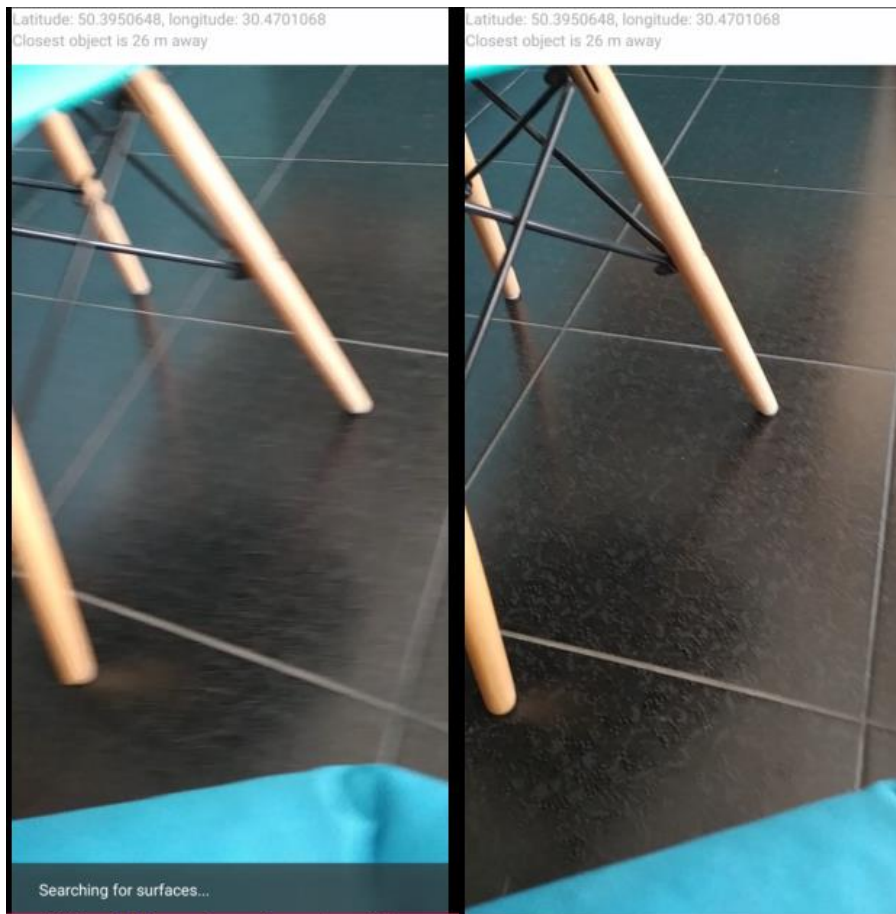


Рисунок – об’єкт не відображається, бо надто далеко

Для прив’язки віртуальних об’єктів до реального світу було реалізовано механізм, при якому за віртуальними об’єктами були закріплені конкретні координати. При зміні координат користувача додаток перевіряє, чи нема в 10 метрах від наявного місцеположення закріпленого об’єкту та, якщо є, починає шукати поверхню, на якій можна розмістити об’єкт. При знаходженні поверхні об’єкт додається до списку тих, що відображаються.

Для полегшення задачі, було визначено, що відстань між усіма віртуальними об’єктами не менша за 10 метрів. Подібні обмеження були встановлені для того, щоб одночасно можна було відображати не більше одного об’єкта.

Також, при зміні місцеположення пристрою, якщо пристрій віддаляється від віртуального об’єкту більш, ніж на 5 метрів, то об’єкт видаляється зі списку anchors і більше не відображається користувачеві.

Також було реалізовано клас `GeolocationPermissionHelper.java`, що відповідає за отримання дозволу для доступу до місцеположення.

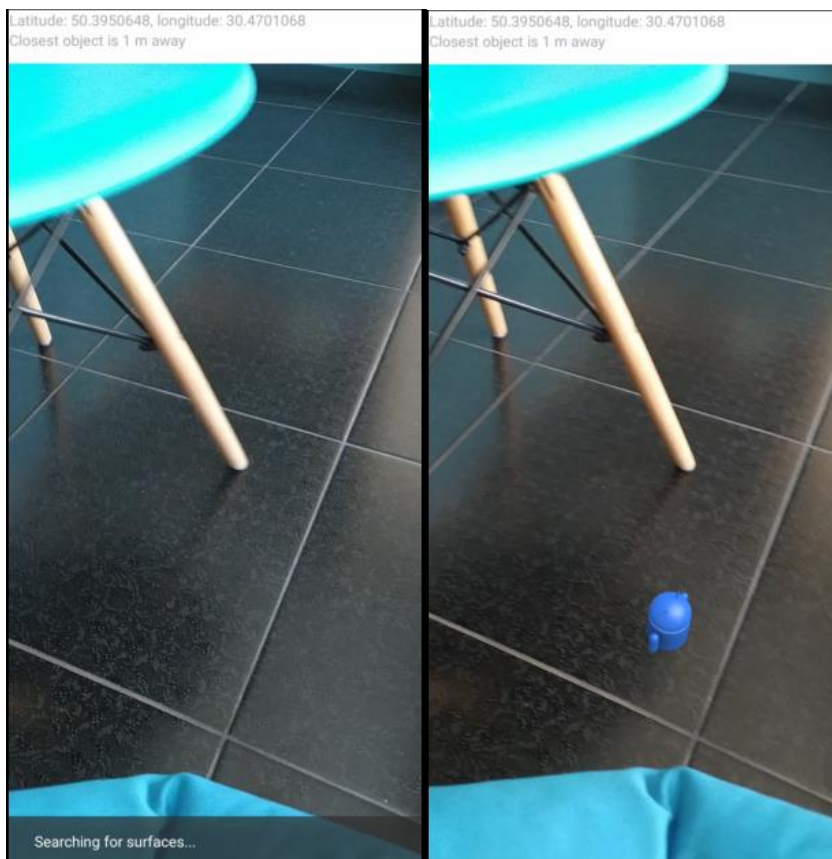


Рисунок – об’єкт достатньо близько, тому відображається

Як було описано в попередніх розділах, робота з доповненою реальністю відбувається повністю за допомогою інструментів open source бібліотеки ARCore.

Висновки

Було досліджено можливі способи прив'язки віртуальних об'єктів до реального світу під час побудови мобільного додатку з використанням ARCore. Також, було створено додаток, що використовує простий спосіб зв'язку віртуального та реального світів з використанням бібліотеки SimpleLocation та розробкою окремих компонентів та логіки.

Було заплановано генерування тексту, що супроводжує віртуальні об'єкти за допомогою Sceneform, але з невизначених причин плагін, необхідний для цього, викликав фатальні помилки при спробі запустити наявні середовища розробки – IntelliJIDEA та AndroidStudio. Фатальні помилки AndroidStudio вдалося вирішити видаленням плагіну, його перевстановлення призвела до повторення фатальних помилок, тому було прийняте рішення про припинення інтегрування плагіну в проект.

Список джерел

1. <https://developers.google.com/ar/discover/concepts>
2. Prakhar Saxena, Geo-location based augmented reality, International Journal of Research in Engineering and Technology, Volume 04 Issue 07 | July-2015, eISSN: 2319-1163 | pISSN: 2321-7308
3. <https://profile.ru/wp-content/uploads/2020/01/Pokemon-Go-782x440.jpg>
4. <http://www.appnews.co.uk/wp-content/uploads/2016/12/viewranger-skyline-app.jpg>
5. <https://wfneurology.org/2019-10-17-usc-news>