РЕФЕРАТ

на магістерську дисертацію

виконану на тему: Керування контентом в віртуальній реальності за допомогою жестів.

студентом: Куцаченко Нікітою Геннадієвичем

Робота виконана на 84 сторінках, містить 47 ілюстрацій, 22 таблиці. При підготовці використовувалася література з 25 різних джерел.

Актуальність

Віртуальна реальність - одна з технологій нашого часу, що розвиваються найбільш стрімко. Вона дає можливість моделювати об'єкти і процеси реального середовища і поміщати користувача в умови і події, створені інформаційною системою.

На сьогоднішній день мобільна VR - це найдешевший для споживачів спосіб зануритися у віртуальну реальність; для неї існує безліч підтримуваних мобільних пристроїв з низькою ціною. Незважаючи на деякі проблеми сучасні і майбутні моделі мобільних пристроїв досить потужні, щоб забезпечувати переконливі відчуття від VR, комфортні для більшості користувачів.

Досі ще не існує уніфікованого методу вводу для віртуальної реальності на основі мобільних пристроїв. У всіх існуючих методів є одні або інші недоліки, до того ж більшість із них потребують придбання додаткового устаткування (контроллера та ін). І саме тому, актуальним було б створити такий спосіб вводу, для якого не потрібно було б купувати будь-яке обладнання та який допоміг би ще глибше відчути ефект присутності у віртуальному просторі.

Мета

Метою даної дипломної роботи є дослідження існуючих інструментів для розробки додатків, що підтримують технологію віртуальної реальності, а також дослідження методів для відстеження руху у відеопотоці та використання

одного з методів на мобільному пристрої, для керування контентом у додатку, що підтримує технологію віртуальної реальності.

Завдання

Для досягнення мети були поставлені наступні завдання:

- Провести огляд віртуальної реальності;
- Провести огляд існуючих методів детектування руху;
- Дослідити існуючі алгоритми оцінки оптичного потоку;
- Проаналізувати основні інструменти для розробки додатків з підтримкою віртуальної реальності;
- Розробити тестовий мобільний додаток з підтримкою віртуальної реальності із можливістю керування контентом за допомогою рухів.

Об’єкт дослідження

Об’єктом дослідження було обрано бібліотеку комп’ютерного зору OpenCV та інструменти для розробки VR-додатків для мобільних пристроїв на платформі Android.

Предмет дослідження

Предметом дослідження в даній роботі є функціональність фреймворків для створення ефекту віртуальної реальності на мобільних платформах, а також алгоритми відстеження руху бібліотеки OpenCV та їх ефективність.

Методи досліджень

Проведення аналізу фреймворків для написання додатків з підтримкою віртуальної реальності на платформі Android, а також методів та засобів бібліотеки комп’ютерного зору OpenCV для відстеження руху на мобільних пристроях та перевірка їх ефективності на прикладі тестового додатку.

Наукова новизна

Наукова новизна роботи полягає в:

1. Виробленні рекомендацій щодо вибору інструментів для реалізації додатків з підтримкою віртуальної реальності на мобільних пристроях на платформі Android
2. Виробленні рекомендацій щодо застосування технологій комп’ютерного зору для відстеження рухів у відеопотоці у сфері мобільної віртуальної реальності.

Практична цінність

В ході роботи був розроблений додаток для мобільних пристроїв з підтримкою технологій віртуальної реальності з використанням фреймворку Gear VR Framework. За допомогою бібліотеки комп’ютерного зору OpenCV було реалізовано модуль, який дозволяє користувачу керувати контентом за допомогою рухів. Результат даної роботи можна використовувати в подальшому для створення лабораторного практикуму на кафедрі СП на тему комп’ютерного зору та віртуальної реальності.

Ключові слова

Віртуальна реальність, комп’ютерний зір, OpenCV, Gear Vr Framework, Unity, Unreal Engine, Oculus Mobile SDK, Daydream.

РЕФЕРАТ

на магистерскую диссертацию

выполненную на тему : Управление контентом в виртуальной реальности с помощью жестов.

студентом: Куцаченко Никитой Геннадиевичем

Работа выполнена на 84 страницах, содержит 47 иллюстрации, 22 таблицы. При подготовке использовалась литература из 25 разных источников.

Актуальность

Виртуальная реальность - одна из технологий нашего времени, развиваются наиболее быстро. Она дает возможность моделировать объекты и процессы реальной среды и помещать пользователя в условия и события, созданные информационной системой.

На сегодняшний день мобильная VR - это самый дешевый для потребителей способ окунуться в виртуальную реальность; для нее существует множество поддерживаемых мобильных устройств низкой цены. Несмотря на некоторые проблемы современные и будущие модели мобильных устройств достаточно мощные, чтобы обеспечивать убедительные ощущения от VR, комфортные для большинства пользователей.

До сих пор еще не существует унифицированного метода ввода для виртуальной реальности на основе мобильных устройств. У всех существующих методов есть те или иные недостатки, к тому же большинство из них нуждаются в приобретении дополнительного оборудования(контроллера и пр). И именно поэтому, актуальным было бы создать такой способ ввода, для которого не нужно было бы покупать никакое оборудование и который помог бы еще глубже почувствовать эффект присутствия в виртуальном пространстве.

Цель

Целью данной работы является исследование существующих инструментов для разработки приложений, поддерживающих технологию виртуальной реальности, а также исследования методов для отслеживания

движения в видеопотоке и использование одного из методов на мобильном устройстве, для управления контентом в приложении, поддерживающем технологию виртуальной реальности.

Задание

Для достижения цели были поставлены следующие задачи:

- Сделать обзор виртуальной реальности;
- Сделать обзор существующих методов детектирования движения;
- Исследовать существующие алгоритмы оценки оптического потока;
- Проанализировать основные инструменты для разработки приложений с поддержкой виртуальной реальности;
- Разработать тестовое мобильное приложение с поддержкой виртуальной реальности с возможностью управления контентом с помощью движений;

Объект исследования

Объектом исследования было выбрано библиотеку компьютерного зрения OpenCV и инструменты для разработки VR-приложений для мобильных устройств на платформе Android.

Предмет исследования

Предметом исследования в данной работе является функциональность фреймворков для создания эффекта виртуальной реальности на мобильных платформах, а также алгоритмы отслеживания движения библиотеки OpenCV и их эффективность.

Методы исследований

Проведение анализа фреймворков для написания приложений с поддержкой виртуальной реальности на платформе Android, а также методов и средств библиотеки компьютерного зрения OpenCV для отслеживания движения в мобильных устройствах и проверка их эффективности на примере тестового приложения.

Научная новизна

Научная новизна работы заключается в:

1. Выработке рекомендаций по выбору инструментов для реализации приложений с поддержкой виртуальной реальности на мобильных устройствах на платформе Android
2. Выработке рекомендаций по применению технологий компьютерного зрения для отслеживания движений в видеопотоке в сфере мобильной виртуальной реальности.

Практическая ценность

В ходе работы было разработано приложение для мобильных устройств с поддержкой технологий виртуальной реальности с использованием фреймворка Gear VR Framework. С помощью библиотеки компьютерного зрения OpenCV было реализовано модуль, который позволяет пользователю управлять контентом с помощью движений. Результат данной работы можно использовать в дальнейшем для создания лабораторного практикума на кафедре СП на тему компьютерного зрения и виртуальной реальности.

Ключевые слова

Виртуальная реальность, компьютерное зрение, OpenCV, Gear Vr Framework, Unity, Unreal Engine, Oculus Mobile SDK, Daydream.

ABSTRACT

on master's thesis

on topic: Controlling content in virtual reality using gestures.

Student: Kutsachenko Nikita Gennadievich

Work carried out on 84 pages containing 47 figures, 22 tables. The paper was written with references to 25 different sources.

Relevance

Virtual reality is one of the technologies of our time, developing most quickly. It allows you to model the objects and processes of the real environment and put the user in the conditions and events created by the information system.

For today, VR for mobiles is the cheapest way for consumers to plunge into virtual reality; there are many supported mobile devices of low price for it. Despite some problems, modern and future models of mobile devices are powerful enough to provide convincing VR sensations that are comfortable for users.

Until now, there is still no unified input method for virtual reality based on mobile devices. All existing methods have some disadvantages, also, most of them need to purchase additional equipment (controller, etc.). And for this reason, it would be relevant to create such an input method, for which no equipment would have to be bought, and which would help to feel even deeper effect of presence in the virtual space.

Goal

The aim of this thesis is to study the existing tools for developing applications that support the technology of virtual reality, and research methods for tracking motion in video stream and usage one of the methods on mobile devices for content control in the application that supports virtual reality.

Task

To achieve the goal, the following tasks were set:

- Make an overview of virtual reality;

- Review current methods of motion detection;
- Consider existing algorithms for estimating the optical flow;
- Analyze the main tools for developing applications that support virtual reality;
- Develop a mobile application that supports virtual reality with the ability to manage content using motion;

Object of study

The object of the study was selected computer vision library OpenCV and tools for developing VR-applications for mobile devices on the Android platform.

Subject of study

The subject of study in this paper is frameworks functionality for creating the effect of virtual reality on mobile platforms and motion tracking algorithms of OpenCV library and their effectiveness.

Research methods

Conducting an analysis of the frameworks used for writing applications supporting virtual reality on the Android platform, as well as methods and tools of the computer vision library OpenCV for movement tracking in mobile devices and testing their effectiveness using the example of a test application.

Scientific novelty

The scientific novelty of the work:

1. Making recommendations on the selection of tools to implement applications with support of virtual reality on mobile devices on the Android platform.
2. Making recommendations on the use of computer vision technology to track movements in the video stream in mobile virtual reality.

Practical value

In the course of the work, an application was developed for mobile devices supporting virtual reality technologies using the Gear VR Framework. With the help of the OpenCV computer vision library, a module was implemented that allows the

user to manage content using movements. The result of this work can be used in the future to create a laboratory workshop at the Department of Computer Science on the subject of computer vision and virtual reality.

Keywords

Virtual reality, computer vision, OpenCV, Gear Vr Framework, Unity, Unreal Engine, Oculus Mobile SDK, Daydream.

ЗМІСТ

Перелік умовних позначень, символів, скорочень і термінів 12
ВСТУП 13
1 ВІРТУАЛЬНА РЕАЛЬНІСТЬ 15
Вступ 15
1.1 Властивості та застосування віртуальної реальності 15
1.2 Принципи роботи віртуальної реальності 18
Висновки 22
2 АНАЛІЗ ІНСТРУМЕНТІВ ДЛЯ РОЗРОБКИ ДОДАТКІВ З ПІДТРИМКОЮ ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ НА ПЛАТФОРМІ ANDROID 23
Вступ 23
2.1 Платформа Daydream 24
2.2 Платформа Oculus Mobile SDK 26
2.3 Візуалізатор Unity 28
2.4 Візуалізатор Unreal Engine 32
2.5 Gear VR Framework 37
Висновки 42
3 АНАЛІЗ ТЕХНОЛОГІЙ КОМП'ЮТЕРНОГО ЗОРУ ДЛЯ ВІДСТЕЖЕННЯ РУХУ 44
Вступ 44
3.1 Алгоритми відстеження руху 44
3.2 Бібліотека комп'ютерного зору з відкритим висхідним кодом OpenCV 47
3.3 Детектування руху за допомогою оцінки оптичного потоку засобами OpenCV 51
Висновки 54
4 РОЗРОБКА ДОДАТКУ З ПІДТРИМКОЮ ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ ТА МОЖЛИВІСТЮ КЕРУВАННЯ ЗА ДОПОМОГОЮ ЖЕСТІВ ДЛЯ ТЕСТУВАННЯ НАДІЙНОСТІ 56
4.1 Загальний алгоритм роботи додатку 56
4.2 Реалізація модулю, що відповідає за VR-середовище 59
4.3 Використання OpenCV у Android Studio 61

4.4 Реалізація модулю, що відповідає за відстеження руху за допомогою бібліотеки OpenCV 64
4.5 Аналіз споживання енергії 68
Висновки 70
5 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ 71
Вступ 71
5.1 Опис ідеї проекту 73
5.2 Технологічний аудит ідеї проекту 75
5.3 Аналіз ринкових можливостей запуску стартап-проекту 76
5.4 Розроблення ринкової стратегії проекту 85
5.5 Розроблення маркетингової програми стартап-проекту 95
Висновки 98
ВИСНОВКИ 100
ПЕРЕЛІК ПОСИЛАННЬ 103

Перелік умовних позначень, символів, скорочень і термінів

ПЗ – програмне забезпечення;

CV – Computer Vision;

ГР – гральний рушій;

ц.р. – цього року

VR – віртуальна реальність.

OLED – organic light-emitting diode.

OpenCV – бібліотека комп’ютерного зору з відкритим кодом.

Unity3D – це інструмент для розробки дво- і тривимірних додатків та ігор.

Unreal Engine – це інструмент для розробки дво- і тривимірних додатків та ігор.

OpenGL – програмний інтерфейс для написання додатків, що використовують двовимірну і тривимірну комп'ютерну графіку.

Стереоскопія – це техніка для створення ілюзії глибини картинки за допомогою стереоскопічних засобів для бінокулярного зору.

Оптичний потік – зображення видимого руху, що представляє собою зсув кожної точки між двома зображеннями.

Рендеринг – в комп'ютерній графіці — це процес отримання зображення за моделлю з допомогою комп’ютерної програми. Тут модель – це опис тривимірних об’єктів (3D) на визначеній мові програмування і у вигляді структури даних. Такий опис може містити геометричні дані, положення точки спостерігача, інформацію про освітлення. А зображення – це цифрове растрове зображення.

ВСТУП

Віртуальна реальність - одна з технологій нашого часу, що розвиваются найбільш стрімко. Це інтерактивний тривимірний простір, створений за допомогою мультимедіа технологій, що об'єднує в собі такі форми кодування інформації: вербальна, іконографічна, фонографічна. Ця технологія дає можливість моделювати об'єкти і процеси реального середовища і поміщати користувача в умови і події, створені інформаційною системою. Об'єкти віртуальної реальності зазвичай ведуть себе близько до поведінки аналогічних об'єктів матеріальної реальності. Користувач може впливати на ці об'єкти в згоді з реальними законами фізики (гравітація, властивості води, зіткнення з предметами, відображення і т.п.).

Останнім часом віртуальна реальність все більше поширюється в сфері споживчої електроніки, а також в обробній промисловості, охороні здоров'я, освіті і т.д. Крім іншого, віртуальну реальність використовують для навчання пілотів, машиністів і рятувальників.

На сьогоднішній день мобільна VR - це найдешевший для споживачів спосіб зануритися у віртуальну реальність; для неї існує безліч підтримуваних мобільних пристроїв з низькою ціною. Незважаючи на деякі проблеми сучасні і майбутні моделі мобільних пристроїв досить потужні, щоб забезпечувати переконливі відчуття від VR, комфортні для більшості користувачів.

Актуальність дипломної роботи полягає в тому, що досі ще не існує уніфікованого методу вводу для віртуальної реальності на основі мобільних пристроїв. У всіх існуючих методів є одні або інші недоліки, до того ж більшість із них потребують придбання додаткового устаткування (контроллера та ін). І саме тому, актуальним було б створити такий спосіб вводу, для якого не потрібно було б купувати будь-яке обладнання та який допоміг би ще глибше відчути ефект присутності у віртуальному просторі.

Мета дипломної роботи – дослідження існуючих інструментів для розробки додатків, що підтримують технологію віртуальної реальності, а також дослідження методів для відстеження руху у відеопотоці та використання

одного з методів на мобільному пристрої, для керування контентом у додатку, що підтримує технологію віртуальної реальності.

Для досягнення мети були поставлені наступні завдання:

- Провести огляд віртуальної реальності;
- Провести огляд існуючих методів детектування руху;
- Дослідити існуючі алгоритми оцінки оптичного потоку;
- Проаналізувати основні інструменти для розробки додатків з підтримкою віртуальної реальності;
- Розробити тестовий мобільний додаток з підтримкою віртуальної реальності із можливістю керування контентом за допомогою рухів.

Об’єкт дослідження - бібліотека комп’ютерного зору OpenCV та інструменти для розробки VR-додатків для мобільних пристроїв на платформі Android.

Предмет дослідження - функціональність фреймворків для створення ефекту віртуальної реальності на мобільних платформах, а також алгоритми відстеження руху бібліотеки OpenCV та їх ефективність.

1 ВІРТУАЛЬНА РЕАЛЬНІСТЬ

Вступ

Сьогодні ми є свідками розквіту віртуальної реальності (virtual reality, скорочено VR) - захоплюючої нової технології, яка обіцяє докорінно змінити нашу взаємодію з інформацією, друзями і світом в цілому.

Ідеї віртуальної реальності вже багато років, але тільки в останні кілька років світ підібрався достатньо близько до межі її побутового використання. Сам термін «віртуальна реальність» увійшов у вжиток лише в 1985 році, тридцять років тому. Перша технічна реалізація пристрою, який, за планом розробника Айвена Сазерленда, повинен був занурювати людей в вигаданий світ, побачила світ у 1968 році. Через величезні розміри і побічні ефекти його пристрій не знайшов свого користувача [1].

Але, вже у наші часи, у серпні 2012 року кампанія, запущена Палмером Лакі на Kickstarter, піднімає 250 000 доларів всього за кілька годин. У березні 2014 року Oculus VR за 2 мільярди доларів купує гігант Facebook.

І з того часу розпочинається активний розвиток віртуальної реальності, як і на стаціонарних, так і на мобільних платформах, для яких виходять такі гарнітури віртуальної реальності, як Google Cardboard, а пізніше і більш якісні варіанти, такі як Samsung GearVr та ін.

Якщо в минулому році прибуток від VR склав 1,8 мільярда доларів, то в поточному році прибуток прогнозується вже в розмірі 4,9 мільярда доларів. До 2020 ж році фахівці SuperData обіцяють зростання прибутку від віртуальної реальності до позначки в 37,7 мільярдів доларів. А отже, можна зробити висновок, що технології віртуальної реальності є досить актуальною сферою дослідження.

1.1 Властивості та застосування віртуальної реальності

Віртуальна реальність - це комп'ютерне моделювання 3D-середовища, яка

здається людині, що взаємодіє з нею, виключно реальною завдяки спеціальному електронному обладнанню. Розробники ставлять завдання створити у користувача відчуття присутності у віртуальному середовищі. Для перегляду стереоскопічних 3D-сцен використовується шолом-дисплей (наприклад, у вигляді окулярів).

Користувач може озирнутися навколо, повернувши голову, і переміститися в просторі за допомогою ручного управління або датчиків руху. Об'єкти віртуальної реальності зазвичай ведуть себе близько до поведінки аналогічних об'єктів матеріальної реальності. Користувач може впливати на ці об'єкти у відповідності з реальними законами фізики. Таким чином досягається ефект повної присутності[2].

Властивості VR, на які потрібно орієнтуватися при створенні додатку з підтримкою віртуальної реальності:

- Правдоподібна - підтримує у користувача відчуття реальності того, що відбувається.
- Інтерактивна - забезпечує взаємодію з середовищем.
- Машинно-згенерована - базується на потужному апаратному забезпеченні.
- Доступна для вивчення - надає можливість досліджувати великий деталізований світ.
- Створює ефект присутності - залучає до процесу як мозок, так і тіло користувача, впливаючи на максимально можливе число органів почуттів.

Типи VR:

1. VR з ефектом повного занурення.

   Цей тип передбачає наявність трьох факторів:

   - Правдоподібна симуляція світу з високим ступенем деталізації.
   - Високопродуктивний комп'ютер, здатний розпізнавати дії користувача і реагувати на них в режимі реального часу.
   - Спеціальне обладнання, з'єднаний з комп'ютером, яке

забезпечує ефект занурення в процесі дослідження середовища.

2. VR без занурення

Не кожному і не завжди потрібно повне занурення в альтернативну реальність. До типу «без занурення» відносяться симуляції з якісним зображенням, звуком і контролерами, в ідеалі трансльовані на широкоформатному екрані. Також в цю категорію потрапляють такі проекти, як археологічні 3D-реконструкції древніх поселень або моделі будівель, які архітектори створюють для демонстрації своєї роботи клієнту. Всі перераховані вище приклади не відповідають стандартам VR в повній мірі, але дозволяють відчути модельований світ на кілька рівнів глибше, ніж інші засоби мультимедіа, а тому зараховуються до віртуальної реальності.

3. VR на базі інтернет-технологій.

Фахівці в області комп'ютерних наук розробили спосіб створення віртуальних світів в Інтернеті, використовуючи технологію віртуальної реальності Markup Language, аналогічну HTML. Вона на якийсь час була обділена увагою і зараз вважається застарілою, але з огляду на зростаючий інтерес Facebook до VR, в майбутньому віртуальна реальність обіцяє ґрунтуватися не тільки на взаємодію, але і на інтернет-технології.

Області застосування VR:

1. Навчання

VR використовується для моделювання середовища тренувань в тих заняттях, в яких необхідна попередня підготовка: наприклад, керування літаком, стрибки з парашутом і навіть операції на мозку.

2. Наука

VR дозволяє поліпшити і прискорити дослідження молекулярного і атомного світу: занурюючись у віртуальному середовищі, вчений може поводитися з частинками так, ніби це кубики LEGO.

3. Медицина

Крім допомоги в навчанні хірургів, технологія VR виявляється корисною і на самих операціях: лікар, використовуючи спеціальне обладнання, може керувати рухами робота, отримуючи при цьому можливість краще контролювати процес.

4. Промисловий дизайн та архітектура

Замість того, щоб будувати дорогі моделі машин, літаки або будівлі, можна створити віртуальну модель, що дозволяє не тільки дослідити проект зсередини, але і проводити тестування його технічних характеристик.

5. Ігри та розваги

На даний момент це найвідоміша і найширша область використання VR: сюди входять як гри, так і кіно, віртуальний туризм і відвідування різних заходів.

1.2 Принципи роботи віртуальної реальності

Віртуальна реальність дозволяє користувачеві зануритися у віртуальний світ, на відміну від звичайних екранів перед користувачем, які не дозволяють отримати такий досвід. VR може містити в собі 4 з 5 почуттів, включаючи зір, слух, дотик і, можливо, навіть запах. З цією силою, VR може може помістити людину до віртуального світу досить легко.

Найочевиднішою різницею між віртуальною реальністю і традиційними дисплеями є спосіб представлення графіки користувачу. Тобто, графіка для віртуальної реальності не потребує автоматично бути кращою або гіршою, але саме представлення принципово відрізняється від плоского досвіду, на який ми покладалися на протязі десятиліть.

Стереоскопічний дисплей. Стереоскопічний дисплей є одним з компонентів, який відокремлює віртуальну реальність від більшості систем, відмінних від VR. Стереоскопічні дисплеї представляють інший вид віртуальної сцени для кожного ока, точно таким чином, як і стерео навушники програють різні звуки для кожного вуха. Стереоскопія - це потужний сигнал в мозок, що

деякі об'єкти знаходяться далі, ніж інші. У поєднанні з іншими сигналами глибини, такі як паралакс (здається, що об'єкти на великій відстані рухаються повільніше, ніж об'єкти, що знаходяться близько), лінії, що сходяться і затінення, стереоскопічний дисплей може бути використаний, щоб ефективно створити відчуття присутності [3].

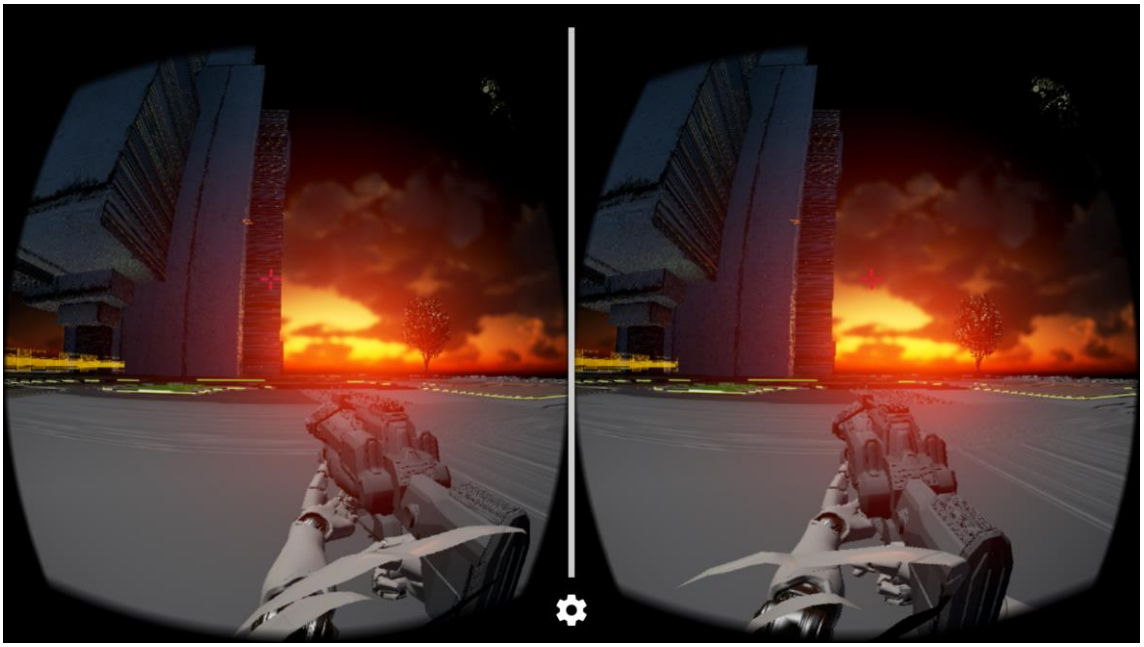

Рисунок 1.1 – Приклад зображення на стереоскопічному дисплеї

Два пласких прямокутника - це ще не всі технології, що потрібні для відображення VR. Проекція світу, в математичному сенсі, принципово відрізняється, коли екран кріпиться до голови користувача. При використанні з лінзами що вбудовані у шолом віртуальної реальності (HMD), бочковидні викривлення дають величезне поле зору (FOV), яке надає відчуття, ніби воно розтягнуте навколо голови користувача. До того ж, лінзи дозволяють розміщувати дисплей ближче до ока.

Рендеринг та людське око. Нерви, які вистилають сітківку ока не викладені в рівномірно рознесену сітку прямокутників. А отже, вони не схожі на пікселі на звичайному екрані монітора комп'ютера. До деякої міри вони є логарифмічно-полярними. Іншими словами, вони розташовані по спіралі навколо центру сітківки, де вони щільно упаковані і стають все більш і більш розрідженими у міру того, як вони рухаються далі від центру. При розгортанні цієї спіралі, пласке зображення стає таким як показано на рисунку 1.2.

Рисунок 1.2 – Приклад сприйняття зображення людським оком

Крім усього іншого це означає, що, люди дійсно добре бачать деталі на яких вони фокусуються, в той час у периферичному зорі не вистачає значних деталей. В майбутньому технології, які будуть рендерити картинку з найвищою чіткістю у центрі, яка буде поступово зменшуватися у відповідності до відстані від нього, дозволить надавати графіці ультра-високу чіткість, де це необхідно, дозволяючи при цьому комп'ютеру, витрачати менше ресурсів там, де така якість непотрібна [4].

Типи дисплеїв для віртуальної реальності. Кілька компаній вже використовують пласкі екрани всередині їх шоломів віртуальної реальності. Oculus VR, компанія, що нещодавно створила Oculus Rift, використовує пару OLED-панелей в їх гарнітурі. OLED-панелі являють собою світлодіодні екрани з плівкою, що виготовлена з органічних сполук. Хоча OLED технологія використовується в таких пристроях, як телефони та комп'ютерні монітори з початку 2000-х років, екрани, що використовуються в гарнітурах віртуальної реальності, таких як Oculus Rift, мають більш суворі вимоги.

Щоб використовувати панель OLED в гарнітурі віртуальної реальності, частота оновлення (як часто нові зображення малюються на екран, щоб

імітувати рух) повинна бути набагато більше, ніж у телевізора, монітора або екрану телефона [5].

Особливості дисплеїв для віртуальної реальності:

- Інші дисплеї оновлюють зображення близько 60 разів на секунду. Дисплей у VR гарнітурі повинен оновити зображення 90 або більше разів на секунду.
- Щоб уникнути розмиття руху, час відгуку (час, за який пікселі змінюють один колір на інший) також має бути швидшим.
- Всі пікселі на екрані мають змінювати колір відразу, замість того, щоб оновлюватися зверху вниз, як це відбувається на екрані телефону або монітора.

Відстеження положення. Віртуальна реальність просто не була б віртуальною реальністю без можливості озирнутися. Відстеження положення голови користувача, під час її руху, є дуже важливим фактором для підтримки ілюзії 360-градусного світу.

Більшість гарнітур віртуальної реальності мають принаймні три ступені свободи: ви можете кивати головою, нахиляючи вниз і вгору, нахиляти голову від плеча до плеча, або повертати голову з боку в бік. Деякі навіть дозволяють переміщатися по фізичній кімнаті в трьох вимірах, що в цілому дає шість ступенів свободи.

Пристрій для запису такого типу інформації, яка необхідна для відстеження голови називається блоком інерціального вимірювання (IMU - inertial measurement unit).

IMU використовує акселерометр, пристрій для вимірювання сили в трьох вимірах, щоб записувати лінійний рух і силу тяжіння. З акселерометром також використовується гіроскоп для вимірювання кутового переміщення. Магнітометр - як тривимірний компас - дає орієнтацію IMU по відношенню до землі. При об'єднанні вимірювань різних пристроїв IMU може оцінити, як пристрій обертається або переміщається з плином часу.

Більшість смартфонів мають IMU, він використовуються, щоб повертати

контент на екрані, коли телефон повертають на бік, або щоб телефон міг використовуватися як компас і рівень в крайньому випадку. Гарнітури використовують IMU (або більш ніж один, в деяких випадках), щоб оцінити не тільки положення пристрою, але і положення голови користувача у віртуальному світі.

IMU мають кілька недоліків. Без фіксованої точки відліку, вони мають тенденцію до «дрейфу». Це означає, що реальний світ залишається на місці, але у віртуальному світі поступово змінюється орієнтація, через те, що невеликі помилки відстеження сумуються.

Деякі пристрої, в тому числі Oculus Rift, значно підвищують їх точність за допомогою комп'ютерного зору для відстеження гарнітури від камерами. Маркери, які легко розрізняються комп’ютером, так звані реперні маркери, можуть використовувати інфрачервоне світло або відбивачі, щоб дати абсолютне положення користувача в системі віртуальної реальності [6].

Висновки

В даному розділі було проаналізовано віртуальну реальність у цілому, її властивості та області застосування.

Огляд існуючих додатків показав основні тенденції в розвитку технології віртуальної реальності. Ця технологія в наш час дуже бурхливо розвивається і може знайти застосування в багатьох областях.

Як зазначено в цьому розділі, застосування технологій віртуальної реальності не обмежується лише іграми. Ці технології використовуються у багатьох важливих сферах. Істотна увага в таких додатках приділяється їх можливостям при взаємодії з навколишнім середовищем і користувачем. Тому було зроблено висновки про актуальність розробки модулю із використанням даних технологій.

Також було розглянуто принципи роботи технологій віртуальної реальності для подальшої реалізації програмного продукту.

2 АНАЛІЗ ІНСТРУМЕНТІВ ДЛЯ РОЗРОБКИ ДОДАТКІВ З ПІДТРИМКОЮ ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ НА ПЛАТФОРМІ ANDROID

Вступ

В даний час існують два основних напрямки VR-розробки - tethered і untethered, тобто на основі комп'ютерів і на основі мобільних пристроїв. Прив'язаний до комп'ютера tethered VR використовує потужні графічні карти настільних комп'ютерів для створення картинки в таких висококласних VR-пристроях, як Oculus Rift і HTC Vive. Для мобільного untethered VR використовують мобільний пристрій користувача, що вставлений в VR-пристрій з пластмаси, картону або пеноматеріалу; при цьому використовуються обчислювальні ресурси тільки користувацького пристрою, характеристики якого можуть змінюватися в широкому діапазоні.

Віртуальна реальність на базі мобільних пристроїв має одну велику перевагу відносно віртуальної реальності на основі комп’ютерів – дешевизну, а отже доступність для кінцевого користувача. Саме тому платформою для розробки модуля було обрано untethered напрям.

Асортимент окулярів для віртуальної реальності дуже широкий, від дешевих Google Cardboard v2 і більш міцних варіацій, до Samsung Gear VR і інших варіантів, які будуть випущені найближчим часом. Великий вибір пристроїв і варіацій - це добре, але для розробки додатків з підтримкою VR в ідеалі потрібен точний набір вивчених апаратних обмежень і можливостей, щоб забезпечити кінцевому користувачеві надійний, плавний і зручний робочий процес. Тому при виборі інструментів для розробки було проаналізовано кращі на сьогоднішній день в своєму класі платформи: Samsung Gear VR і Google Daydream VR, а також такі фреймворки для розробки додатків з підтримкою технологій віртуальної реальності, як Unity, Unreal Engine та Gear VR Framework.

2.1 Платформа Daydream

На конференції Google I / O 2016 18 травня 2016 року, яку регулярно проводить компанія, була анонсована нова система Daydream. Daydream - це гарнітура VR, комплектний контролер і відповідне ПО, засноване на адаптованій для даного сегмента версії Android N.

Система Daydream, в якій використувується унікальний софт і залізо, стала просунутим спадкоємцем Cardboard, картонних окулярів, випущених пошуковим гігантом три роки тому і задасть стандарти якості для обладнання віртуальної реальності. Система передбачає наявність пристрою для занурення в VR, що працює з смартфонами на базі Android N, створеними за стандартами Google. [7]

Рисунок 2.1 – VR гарнітура Daydream

Якщо картонні окуляри могли працювати практично з будь-яким смартфоном, лише б йому вистачало продуктивності і якості екрану, то Daydream буде обмежений певною лінійкою пристроїв, так як тільки вони будуть обладнані необхідними сенсорами і екранами. На даний момент немає ніяких ознак того, що пошуковий гігант буде використовувати щось на зразок свого проекту Google Tango зі створенням 3D карти навколишнього простору або прийоми доповненої реальності. Але компоненти системи передбачають

якісну графіку без зависань, швидкий відгук і інші переваги, здатні зробити занурення у віртуальну реальність ще більш зручним і вражаючим і які неможливо отримати простим оновленням програмного забезпечення.

Для смартфонів, які зможуть працювати з Daydream, Google приготував особливий режим роботи, який отримав назву Android VR Mode. Він вбудовується в останні версії операційної системи. VR Mode включає в себе серію оптимізацій, призначених поліпшити швидкодію додатків. Це ціла екосистема, яка пропонує власникам смартфонів зручну навігацію в віртуальному просторі.

Так домашній екран Daydream дозволить вибирати програми та інший контент, не знімаючи шолома і не виймаючи з нього смартфона. На даний момент використовується нізкополігональних фон тропічного лісу, що нагадує те, що використовував пошуковий гігант в своїх додатках для Cardboard. Тут користувачеві будуть запропоновані спеціальні VR-версії YouTube, Play Store, Play Movies, Street View і Google Photos [8].

Рисунок 2.2 – Домашній екран Daydream

Також відомо, що Google привернув кілька сторонніх компаній, щоб вони перенесли додатки в екосистему Daydream, в тому числі платформи для стрімінга відео, такі як Netflix, і продукти ігрових компаній, в тому числі Ubisoft і Electronic Arts.

Google відзначає, що SDK покликаний істотно полегшити виконання основних завдань при розробці додатків для смартфонів і гарнітур віртуальної реальності, здатних працювати з Daydream. З цією метою, зокрема, реалізована нативна інтеграція Unity і Unreal Engine 4.

2.2 Платформа Oculus Mobile SDK

Oculus Mobile SDK включає в себе бібліотеку, інструменти та ресурси для нативної розробки під гарнітуру віртуальної реальності Gear VR.

Gear VR може з'єднуватися з телефонами Samsung Galaxy S6 та пізнішими моделями. Не зважаючи на те, що інші Android смартфони не можуть бути використані, на даний момент Gear VR має більше моделей сумісних телефонів, ніж Google Daydream View.

Рисунок 2.3 – Gear VR

Oculus Mobile SDK містить:

- VrApi для інтеграції движка третьою стороною (не потрібно для Unity або Unreal).
- Native фреймворк для створення високопродуктивних додатків VR з нуля.

- Додаткові бібліотеки, що забезпечує підтримку GUI, локалізації і інших функцій.
- Приклади нативних додатків та вихідні коди для забезпечення еталонної моделі при створенні власних додатків для VR.
- Інструменти і ресурси для надання допомоги при нативній розробці.

Native SDK містить чотири основних нативні бібліотеки:

- VrApi: мінімальний API для VR;
- VrAppFramework: фреймворк, що використовується нативними додатками;
- VrAppSupport: підтримка GUI, звуку і т.д.;
- LibOVRKernel: низькорівнева Oculus бібліотека для контейнерів, математичних операцій і т.д.

VrApi забезпечує мінімально необхідний API для візуалізації сцен в VR. Додатки можуть відправляти запити до VrApi для отримання даних орієнтації, і передавати текстури, щоб застосовувати викривлення, поєднання даних з датчиків і композицію. VrApi Input API, дозволяє розробникам робити запити щодо стану підключених пристроїв, таких як Gear VR Controller. Розробники, що працюють з іншими візуалізаторами, крім Unity або Unreal будуть використовувати VrApi щоб інтегрувати мобільний SDK.

VrAppFramework підтримє інтеграцію VrApi і надає обгортку навколо Android activity, яка управляє життєвим циклом Android-додатку. VrAppFramework є основою для декількох перших сторонніх додатків, в тому числі Oculus video і Oculus photo 360.

VrAppSupport і LibOVRKernel забезпечують мінімальну функціональність для додатків, що використовують VrAppFramework, таких як GUI, звук і т.д.

LibOVRKernel і VrAppFramework поставляються з повним вихідним кодом, а також у вигляді готових попередньо збудованих бібліотек і .aar файлів. VrApi поставляється у вигляді набору загальних include-файлів і попередньо збудованих динамічних бібліотек. Представлення VrApi в окремій динамічній бібліотеці дозволяє реалізації VrApi бути оновленою після того, як додаток

було випущено, що полегшує застосування необхідних виправлень, впровадження необхідних оптимізацій, а також додавання підтримки нових пристроїв, що не вимагає ще одну компіляцію додатку з новим SDK [9].

Крім нативної розробки Oculus Mobile SDK підтримує також інтеграцію з найбільш відомими візуалізаторами, такими як Unity та Unreal Engine.

Безкоштовна і професійна версії редактора Unity 5.1 або більш пізні версії підтримують розробку під Rift та Gear VR з коробки. Підтримка VR вмикається за допомогою простого відмічання прапорця в налаштуваннях.

Для підтримки Unity Oculus пропонує опціональні утиліти для пакета Unity 5, які включають в себе сценарії, префаби і інші ресурси. Пакет включає в себе інтерфейс для управління поведінкою VR камери, префаб для підтримки виду від першої особи, єдиний вхідний інтерфейс для контролерів, розширені можливості візуалізації, засоби для дебаггінгу і багато іншого.

Коли підтримка VR включена Unity автоматично застосовує відстеження положення і орієнтації, стереоскопічний рендерінг і виправлення викривлень для основної камери,.

Unreal поширюється з плагінами Oculus, які дозволяють спростити розробку для Oculus Rift і Gear VR.

Додатки, орієнтовані на пристрої Oculus автоматично застосовують стереоскопічний рендеринг, відстеження орієнтації (Rift і Gear VR) і позиції (тільки Rift). Плагін OculusPlatform надає онлайн підсистему, яка взаємодіє з функціями Oculus Platform. Плагін OculusInput забезпечує підтримку контролерів [10].

2.3 Візуалізатор Unity

Unity — багатоплатформовий інструмент для розробки дво- та тривімирних додатків та ігор, що працює на операційних системах Windows і OS X. Створені за допомогою Unity застосування працюють під системами Windows, OS X, Android, Apple iOS, Linux, а також на гральних консолях Wii, PlayStation 3 і XBox 360 [11].

Є можливість створювати інтернет-додатки за допомогою спеціального під'єднуваного модуля для браузера Unity, а також за допомогою експериментальної реалізації в межах модуля Adobe Flash Player. Застосування, створені за допомогою Unity, підтримують DirectX та OpenGL.

Редактор Unity має простий Drag & Drop інтерфейс, який легко налаштовувати, що складається з різних вікон, завдяки чому можна проводити налагодження гри прямо в редакторі. Рушій підтримує три сценарних мови: C #, JavaScript (модифікація). Проект в Unity ділиться на сцени (рівні) - окремі файли, що містять свої ігрові світи зі своїм набором об'єктів, сценаріїв, і налаштувань. Сцени можуть містити в собі як, об'єкти (моделі), так і порожні ігрові об'єкти – тобто ті які не мають моделі. Об'єкти, в свою чергу містять набори компонентів, з якими і взаємодіють скрипти. Також у них є назва (в Unity допускається наявність двох і більше об'єктів з однаковими назвами), може бути тег (мітка) і шар, на якому він повинен відображатися. Так, у будь-якого предмета на сцені обов'язково присутній компонент Transform - він зберігає в собі координати місця розташування, повороту і розмірів по всіх трьох осях. У об'єктів з видимою геометрією також за замовчуванням присутній компонент Mesh Renderer, що робить модель видимою.

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

При імпорті текстури в рушій можна згенерувати alpha-канал, mip-рівні, normal-map, light-map, карту відображень, проте безпосередньо на модель текстуру прикріпити не можна - буде створено матеріал, з яким буде призначений шейдер, і потім матеріал прикріпиться до моделі. Редактор Unity підтримує написання і редагування шейдерів. Крім того він містить компонент для створення анімації, яку також можна створити попередньо в 3D-редакторі та імпортувати разом з моделлю, а потім розбити на файли.

Графічний рушій використовує DirectX (Windows), OpenGL (Mac, Windows, Linux), OpenGL ES (Android, iOS), та спеціальне власне API для Wii. Також підтримуються bump mapping, reflection mapping, parallax mapping, screen space ambient occlusion (SSAO), динамічні тіні з використанням shadow maps, render-to-texture та повноекранні ефекти post-processing [12].

Unity підтримує файли 3ds Max, Maya, Softimage, Blender, modo, ZBrush, Cinema 4D, Cheetah3D, Adobe Photoshop, Adobe Fireworks та Allegorithmic Substance. В ігровий проект Unity можна імпортувати об'єкти цих програм та робити налаштовування за допомогою графічного інтерфейсу.

Для написання шейдерів використовується ShaderLab, що підтримує шейдерні програми написані на GLSL або Cg. Шейдер може включати декілька варіантів реалізації, що дозволяє Unity визначати найкращий варіант для конкретної відеокарти. Unity також має вбудовану підтримку фізичного рушія Nvidia PhysX (колишнього Ageia), підтримку симуляції одягу в системі реального часу на довільній та прив'язаній полігональній сітці (починаючи з Unity 3.0), підтримку системи ray casts та шарів зіткнення.

Скриптова система ігрового рушія зроблена на Mono — вільний відкритий проект з реалізації .NET Framework. Програмісти можуть використовувати UnityScript (власна скриптова мова, подібна до JavaScript та ECMAScript), C# або Boo (мова програмування, подібна до Python). Починаючи з версії 3.0, до Unity входить перероблена версія MonoDevelop для зневадження скриптів.

Із виходом версії 5.2 предбачається вбудована можливість редагувати скрипти у середовищі Visual Studio

Unity VR дозволяє розробляти додатки з підтримкою віртуальної реальності безпосередньо за допомогою фреймворка Unity, без будь-яких зовнішніх плагінів в проектах. Він забезпечує базовий API і набір функцій з підтримкою сумісності для декількох пристроїв. Unity VR був розроблений, щоб забезпечити сумісність з майбутніми пристроями і програмним забезпеченням. VR API поки ще має мінімальний дизайн, але буде

вдосконалюватися в міру того, як інтерес до VR продовжує зростати.

Використовуючи вбудовану підтримку VR в Unity, ви отримуєте:

- Стабільні версії для кожного пристрою VR
- Єдиний інтерфейс API, щоб взаємодіяти з різними пристроями VR
- Чиста папка проекту без зовнішнього плагіна для кожного пристрою
- Можливість включати і перемикатися між кількома пристроями в додатках
- Підвищення продуктивності (низькорівнева оптимізація фреймворку Unity можлива для нативних пристроїв)

Коли VR включений в Unity, кілька речей відбувається автоматично:

1. Автоматичний рендеринг на дисплеї гарнітури віртуальної реальності.

Всі камери в сцені спроможні здійснювати рендер на дисплей VR-гарнітури (HMD). Матриці projection та view автоматично коригуються з урахуванням трекінгу положення голови, позиції та поля зору.

Можна відключити рендеринг на HMD, використовуючи властивість stereoTargetEye компонента камери. Крім того, ви можете встановити камери, для того щоб рендерити до Render Texture, використовуючи властивість Target Texture.

Можна використовувати властивість stereoTargetEye для того, щоб камера рендерила тільки одне око до HMD. Це корисно для спеціальних ефектів, таких як снайперський приціл або стереоскопічне відео. Для досягнення цієї мети, треба додати дві камери на сцену: одну орієнтовану на ліве око, а іншу на праве. При цьому треба установити маски шарів, щоб налаштувати те, що надсилається на кожне око.

2. Автоматичне введення трекінгу голови

Трекінг голови і відповідне поле зору (FOV) автоматично застосовуються до камери, якщо пристрій знаходиться на голові. Фреймворк надає змогу вручну встановити певне значення FOV, але не можна безпосередньо встановити значення, що трансформують камеру.

Відстеження голови та позиції застосовується автоматично, так що

положення і орієнтація у віртуальному просторі найбільш точно відповідають позиції і орієнтації користувача до рендерінгу кадру. Це дає хороший досвід VR, і допомогає знизити ризик виникнення нудоти у користувача.

3. Розуміння камери

Camera Transform перекривається відстеженою позою голови. Для того, щоб перемістити або повернути камеру треба прикріпити її як дочірній об’єкт іншого GameObject. Таким чином, всі Transform перетворення батьківського об’єкта камери впливають на саму камеру. Це також відноситься і до переміщення або повороту камери за допомогою скрипту.

Існує відмінність між досвідом VR під час сидіння та під час використання усього простору кімнати: якщо пристрій підтримує використання усього простору кімнати, стартова позиція камери є центром ігрового простору користувача. Якщо ж передбачається, що користувач буде сидіти, розробник може скинути камеру в нейтральне положення за допомогою VR.InputTracking.Recenter () [13].

2.4 Візуалізатор Unreal Engine

Unreal Engine — ігровий рушій, розроблюваний і підтримуваний компанією Epic Games.

Перша гра, створена на цьому рушію — Unreal — з'явилася 1998 року. З тих пір різні версії цього ігрового рушія використали в більш ніж сотні ігор, серед яких Deus Ex, Lineage II, Thief: Deadly Shadows, Postal 2, серіях ігор Brothers in Arms, серія ігор Splinter Cell, Tom Clancy's Rainbow Six, а також у відомих ігрових серіях Unreal і Unreal Tournament від самих Epic Games. Пристосований у першу чергу для шутерів від першої особи, рушій використовувався й при створенні ігор інших жанрів.

Написаний мовою C++, рушій дозволяє створювати ігри для більшості операційних систем і платформ: Microsoft Windows, Linux, Mac OS і Mac OS X, консолей Xbox, Xbox 360, PlayStation 2, PlayStation Portable, PlayStation 3, Wii, Dreamcast і Nintendo GameCube. У грудні Марк Рейн продемонстрував роботу

рушія Unreal Engine 3 на iPod Touch і iPhone 3GS. У березні 2010 робота рушія була продемонстрована на комунікаторі Palm Pre, що базується на мобільній платформі webOS.

Для спрощення портування рушій використовує модульну систему залежних компонентів: підтримує різні системи рендерингу (Direct3D, OpenGL, Pixomatic; раніше підтримувалися Glide API, S3 Metal, PowerVR SGL), відтворення звуку (EAX, OpenAL, DirectSound3D; раніше підтримувалися A3D), засоби голосового відтворення тексту, розпізнавання мовлення (тільки для Xbox360, PlayStation 3, Nintendo Wii і Microsoft Windows, також планувалося для Linux і Mac), модулі для роботи з мережею й підтримка різних пристроїв вводу.

Для гри у мережі підтримуються технології Windows Live, Xbox Live, і GameSpy, включаючи до 64 гравців (клієнтів) одночасно. Попри те, що офіційно засоби розробки не містять у собі підтримки великої кількості клієнтів на одному сервері, рушій використовувався для створення MMORPG-ігор. Один з найвідоміших представників жанру, Lineage II, використовує рушій Unreal Engine [14].

Усі елементи ігрового рушія представлені у вигляді об'єктів, що мають набір характеристик, і клас, який визначає доступні характеристики. У свою чергу будь-який клас є «дочірнім» класом object. Серед основних класів і об'єктів можна виділити наступні:

- Актор (actor) — базовий клас, що містить усі об'єкти, які мають відношення до ігрового процесу й мають просторові координати.
- Павн, пішак (pawn) — фізична модель гравця або об'єкта, керованого штучним інтелектом. Метод керування описаний спеціальним об'єктом, такий об'єкт називається контролером. Контролер штучного інтелекту описує лише загальну поведінку пішака під час ігрового процесу, а такі параметри як «здоров'я» (кількість пошкоджень, після яких пішак перестає функціонувати) або, наприклад, відстань, на якій пішак звертає увагу на звуки, задаються для кожного об'єкта окремо.

- Світ, рівень (world, game level) — об'єкт, що характеризує загальні властивості «простору», наприклад, силу тяжіння й туман, у якому розташовуються всі актори. Також може містити в собі параметри ігрового процесу, як, наприклад, ігровий режим, для якого призначений рівень.

Для роботи із простими й, як правило, нерухомими елементами ігрового простору (наприклад, стіни) використовується бінарна розбивка простору — увесь простір ділиться на «заповнене» і «порожнє». В «порожній» частині простору розташовуються всі об'єкти а також тільки в ній може перебувати «точка спостереження» при відмальовці сцени. Можливість повного або часткового поміщення об'єктів в «заповнену» частину простору не виключається, однак може привести до неправильної обробки таких об'єктів (наприклад, розрахунки фізичної взаємодії) або неправильної відмальовки у випадку поміщення туди «точки спостереження» (наприклад, ефект «залу дзеркал»). Усі пішаки, що потрапляють в «заповнену» частину простору, відразу «гинуть».

Поверхня (surface) є основним елементом бінарного дерева простору. Ці елементи створюються на грані перетинання між «заповненою» і «порожньою» частинами простору. Група елементів бінарного дерева простору називається нодом (node, укр. вузол). Цей термін, як правило, уживається в контексті node count — кількість нодів на екрані або в ігровому просторі взагалі. Кількість нодів, одночасно видимих на екрані впливає на продуктивність при промальовуванні сцени. Якщо якийсь нод не потрапляє на екран або перекривається цілком іншими нодами, він не обраховуються — це служить для підвищення продуктивності, особливо в закритих просторах. Розбивка всього простору на групи нодів називається зонуванням. Для цього іноді використовуються портали — невидимі поверхні, які служать для того щоб вручну розділити великий нод на два менші. Крім порталів використовуються антипортали, які обмежують області відмальовки [15].

Опис «заповнених» і «порожніх» частин простору виконується за

допомогою набору замкнених тривимірних об'єктів, складених з не пересічних поверхонь — брашей (brush, укр. пе́нзель). Цей принцип побудови простору називається конструктивною суцільною геометрією. Геометрія може бути «аддитивною» (увесь простір початково «порожній») і «віднімальною» (початково заповнений матерією простір). Браши діляться на три типи:

- Суцільні (solid) — повноцінно беруть участь у бінарній розбивці простору.
    - Аддитивні (additive) — «заповнюють» бінарний простір.
    - Віднімальні (substractive) — «вирізують» об'єми у просторі.
- Напів-Суцільні (semi-solid) — не впливають напряму на бінарне дерево простору, однак впливають на її фізичну модель. Можуть тільки «заповнювати» простір. Слугують для створення «невидимих» перешкод, а також зниження числа полігонів і нодів.
- Порожні (non-solid) — тільки створюють поверхні, не впливають на бінарне дерево простору. Використовуються переважно для створення об'ємів (volume) — частина простору, яка має властивості, відмінні від властивостей ігрового світу. Об'єми мають пріоритет, властивості об'єму з великим пріоритетом застосовуються до акторів, що перебувають у ньому. Ігровий світ завжди має мінімальний пріоритет. За допомогою об'ємів можна змінити гравітацію, в'язкість, туман і таке інше. Об'єми, починаючи з версії рушія Unreal Engine 2, використовуються для створення води (але не водної поверхні).

Unreal поширюється з плагінами Oculus, які дозволяють легко розробляти для Oculus Rift і Gear VR.

Завдяки вбудованій підтримці Unreal Engine створювати матеріали для Gear VR або Oculus Rift зможуть як досвідчені розробники, так і початківці. Unreal пропонує зручний візуальний інтерфейс з можливістю швидкої інтеграції на різних платформах. Завдяки технологіям кешування від Unreal збірки Gear VR можна оновлювати за лічені секунди по Wi-Fi.

Додатки Unreal, запущені на платформі Oculus, автоматично

застосовують стереоскопічну обробку в основній камері. Крім того, вони забезпечують відстеження координат положення голови з малою затримкою для Rift або Gear VR. Для доступу до ключових функцій Oculus або C ++ та для виконання спеціалізованих завдань можна використовувати вузли Blueprint. Плагін OculusPlatform надає онлайн підсистему, яка взаємодіє з функціями Oculus Platform. Плагін OculusInput забезпечує підтримку для контролерів Oculus Touch і Oculus Remote [16].

Epic забезпечує поширення бінарних файлів UE4 через Launcher і поширення висхідного коду рушія з їх сховища GitHub. Обидва ці види поширення забезпечують підтримку Oculus, але версія SDK може бути на один або два релізи позаду актуального.

Oculus також поширює UE4 через Oculus репозиторій GitHub. Це поширення завжди містить актуальний Oculus SDKs. Oculus підтримує поточний реліз UE4, попередній реліз UE4 і будь-який анонс наступного релізу UE4.

Ця послідовність розробки показана на рисунку 2.4:

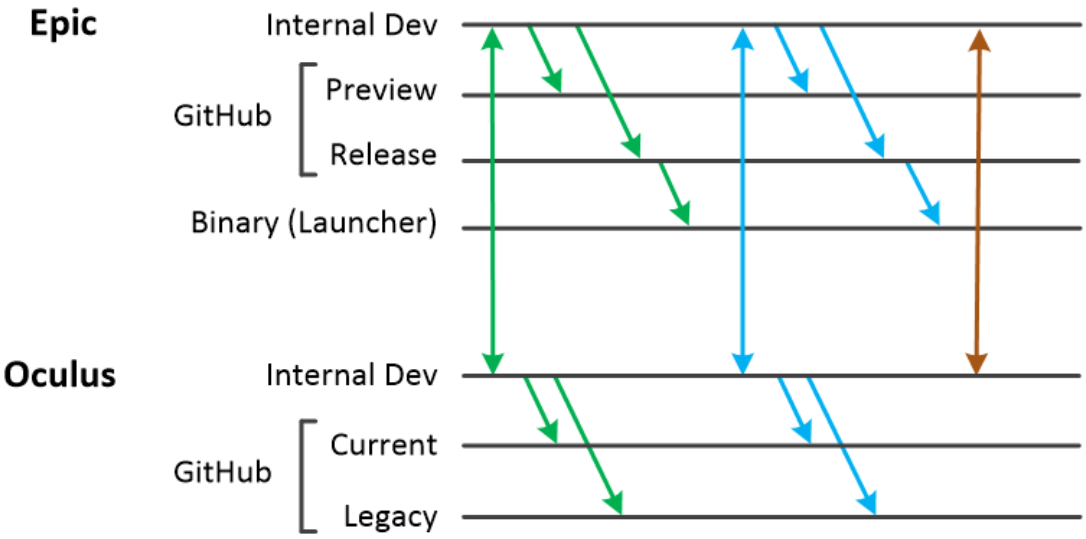

Рисунок 2.4 – Послідовність розробки Oculus SDK та Unreal Engine

Ресурси Oculus для Unreal:

- Platform SDK. Oculus Platformпідтримує функції, пов'язані з безпекою, ком’юніті, доходами та запрошеннями, такими як перевірка надання прав, створення мультиплеєрного ігрового сеансу, покупки у додатку, VoIP і

збереження даних у хмарах.

- Avatar SDK. Oculus Avatar SDK включає в себе Unreal пакет, для допомоги реалізації представлення рук у певному вигляді від першої особи для Rift і контролерів Touch. Він включає в себе аватар рук і тіла, що можуть переглядатися іншими користувачами в соціальних додатках для Rift і Gear VR. Моделі рук для першої особи і моделі рук та тіла для третьої особи, що підтримуються Avatar SDK автоматично підтягують вибір конфігурації аватара, який користувач зробив в Oculus Home, щоб забезпечити почуття ідентичності в різних додатках.
- Аудіо ресурси. Oculus Audio SDK включає в себе плагіни, які забезпечують моделювання положення джерела звуку і реверберації для інструментів редагування аудіо, зазвичай використовуваних з Unreal, в тому числі Audiokinetic Wwise і FMOD Studio.

2.5 Gear VR Framework

Gear VR Framework (GearVRf) Project представляє собою легкий, потужний, з відкритим вихідним кодом, візуалізатор з інтерфейсом Java для розробки мобільних ігор VR і додатків для Gear VR і Google Daydream View.

GearVRf має такі особливості:

- Простий - Java інтерфейс, Android Studio як середовище збірки і простий SDK дозволить вам швидко створювати прототипи. Поглиблених знань щодо рендерингу OpenGL і Oculus / Daydream не потрібно.
- Потужний - специфічні для VR оптимізації рендеринга і опціональний доступ до графічного конвеєра низького рівня дозволяють створювати і оптимізувати високопродуктивну графіку.
- Оптимізований для мобільних пристроїв - створений саме для виконання на мобільній платформі, GearVRf забезпечує легкий і орієнтований на продуктивність доступ до викликів на системному рівні Android OS.
- Відкритий вихідний код - це означає, що немає ліцензійних зборів або ліцензійних платежів. Заохочується внесок активних розробників

співтовариства.

- Ефективний - шар інтерфейсу GearVRf абстрагується від цільової мобільної платформи VR SDK. Ви можете написати код один раз і побудувати його для і Gear VR, і для Daydream / Cardboard. Параметри збірки за замовчуванням створюють єдиний APK, який працює для обох систем, за допомогою виконання перевірки потоку для обслуговування Oculus, що поверне до Daydream, якщо немає в наявності Oculus. Підтримкується навіть Google Cardboard, бекенд Daydream повертає до служби Google Cardboard якщо мобільний пристрій не підтримує Daydream.

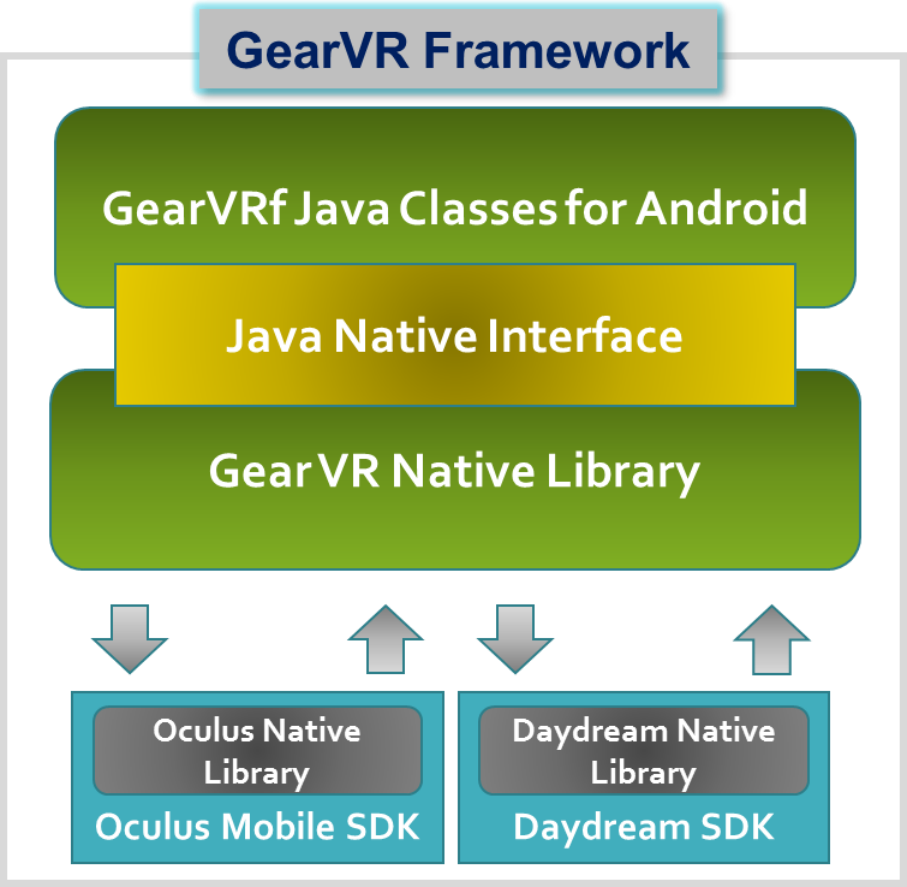

Рисунок 2.5 – Структура GearVR Framework

GearVRf надає інструменти для прискорення розробки вдосконалених функцій в додатках VR високої якості. Доступні розширення EGL (включаючи подвійне сканування, передній буфер, MSAA, і рендеринг плитки) дозволяють поліпшити якість рендерингу.

GearVRf є нативним 3D движком для рендеринга з інтерфейсом Android бібліотеки. Він дозволяє створити нетривіальний контент, використовуючи тільки вбудовані об'єкти. Він надає змогу додавати нові об'єкти (такі як об'єкти сцени з або без GL шейдерів), отриманих з класів або перекриття деяких методів - GearVRf піклується про все на апаратному рівні. Ви можете робити майже все в Java - весь вихідний код публікується, так що ви можете легко додати або змінити нативний код [17].

GearVRf є фреймворком, який керує тим, як і коли виконується код. Перевизначення об'єктів GearVRf дозволяє додати свій власний код. Ви також можете відслідковувати події GearVRf і забезпечувати зворотні виклики, які відповідають на них.

3D-сцена представлена у вигляді ієрархії об'єктів GearVRf сцени. Кожен видимий об'єкт має трикутну сітку, яка описує форму, матеріал, що описує його колірні властивості і матрицю перетворення, що контролює його положення в 3D-світі. Ви явно не викликаєте OpenGL при використанні GearVRf. Замість цього, фреймворк GearVRfуправляє всім рендерингом, забезпечуючи більш високий рівень абстракції для графіки.

При побудові Android програми, треба створити підклас класу Activity. Так само, при побудові програми GearVRf треба створити підклас GVRActivity, надаючи код ініціалізації для створення GVRMain, щоб створити початкову 3D сцену і обробляти вхідні події.

Під час ініціалізації GVRActivity створює GVRViewManager, який робить всю важку роботу. Цей клас відповідає за планування задач, 3D-рендеринг, анімації та завантаження ассетів.

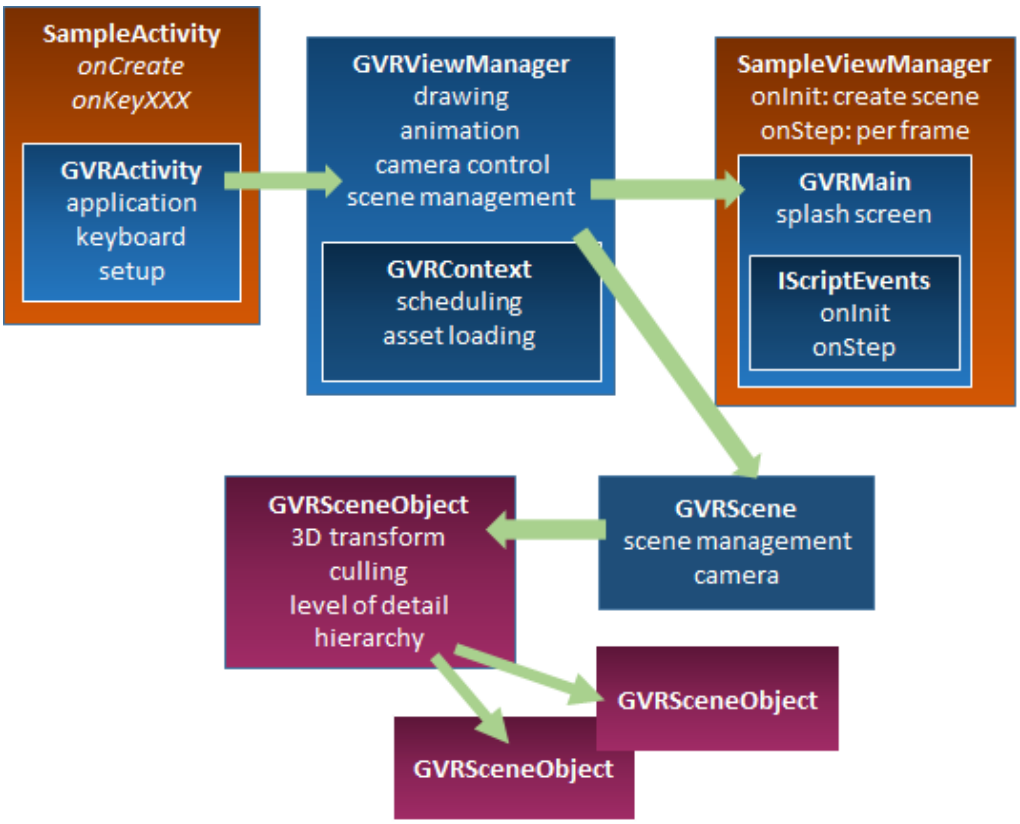

Рисунок 2.6 – Структура роботи додатку, релізованого з використанням GearVR Framework

Одним з ключових обмежень програмування вбудованого GPU є те, що є може існувати тільки один контекст GL. Тобто, всі команди GPU повинні виходити з одного й того самого потоку - потоку GL. GPU завжди повинен бути зайнятий; отже, GL потік не може бути основним GUI потоком.

При запуску GearVRf, Android додаток створює GL потік, встановлює телефон в стереоскопічний режим і підтримує пару методів зворотного виклику, які запускають код старту додатку і код, що повинен виконуватися в кожному кадрі в GL потоці. GearVRf надає методи для будь-якого потоку, за допомогою яких можна запланувати виклики потоку GL. Всі ці функції зворотного виклику означають, що програмування за допомогою GearVRf є подієво-орієнтованим в GL потоці, так само, як і Android програмування є подієво-орієнтованим в GUI потоці. Запуск двох незалежних систем в двох незалежних потоках означає, що треба мати на увазі IPC, коли Android Activity код в GUI потоці взаємодіє з кодом GearVRf в GL потоці. Операції в обох потоках створюють ще одну величезну частину додатку, яка може отримати

перевагу з атомарності подій. Тобто, зворотні виклики - це виклики методів з основного циклу - ні GUI потік, ні GL потік ніколи не може виконувати більше одного зворотного виклику одночасно, і кожен зворотний виклик повинен виконатися до кінця, перш ніж будь-який інший зворотній виклик зможе почати працювати у цьому потоці. Розробнику не потрібно писати додатковий код в зворотних викликах GL, для того щоб запобігти від доступу в частково оновлені структури даних інших зворотних викликів.

Код запуску будує граф сцени, що складаються з об'єктів сцени, а потім інший код покадрово маніпулює графом сцени в реальному часі. Кожен об'єкт сцени має матрицю 4x4, яка описує його стан, орієнтацію і масштабування щодо свого батьківского об’єкту. Кожен об'єкт сцени бути батьківським об’єктом інших об'єктів сцени, так що складні об'єкти можуть складатися з декількох невеликих об'єктів, кожен зі своєю власною формою і зовнішнім виглядом, а всі зміни будуть відбуватися синхронно. Кожен об'єкт сцени має методи для зміни його компонентів за допомогою ледачого шаблону оновлення, а це значить, що вартість декількох оновлень за один раз бути вартувати не набагато більше, ніж одне оновлення.

Об'єкт сцени можна зробити видимим, додавши геометрію поверхні і скін. Геометрія являє собою сітку з 3D трикутників. GearVRf надає методи для створення простих прямокутних чотирикутників та для завантаження більш складних мешів з файлів, створених за допомогою 3D редакторів.

Кожен клас матеріалу містить тип шейдера, GL ідентифікатор шейдера, значення для всіх параметрів шейдера, текстури та ін. Кожен шейдер має дві частини: вершинний шейдер і фрагментний шейдер. Вершинний шейдер викликається для кожної вершини кожного видимого трикутника і може обчислювати конкретні значення трикутника, які передаються в фрагментний шейдер, який малює кожен піксель кожного видимого трикутника. GearVRf містить стандартні шейдери, які забезпечують методи, такі як звичайна дискретизація текстури (растрового зображення в пам'яті GPU), без застосування будь-яких ефектів освітлення. Розробник може легко створювати

власні шейдери.

Граф сцени описує просторові відносини між об'єктами в сцені. Кожен об'єкт сцени має матрицю перетворення 4x4 для того, щоб можна було розташувати і орієнтувати його локально. Об'єкти сцени можуть бути вкладеними, так що перетворення батьківських вузлів успадковується дочірніми вузлами. Це дозволяє легко встановлювати і анімувати об’єкти відносно один одного [18].

На рисунку 2.7 можна побачити граф сцени для метелика з тілом і двома крилами. Кожен об'єкт сцени має положення і орієнтацію. Ліві і праві крила можуть спільно використовувати один і той же меш, але він розташований і орієнтований по-різному для кожного крила. Початкове зміщення тіла успадковується крилами.

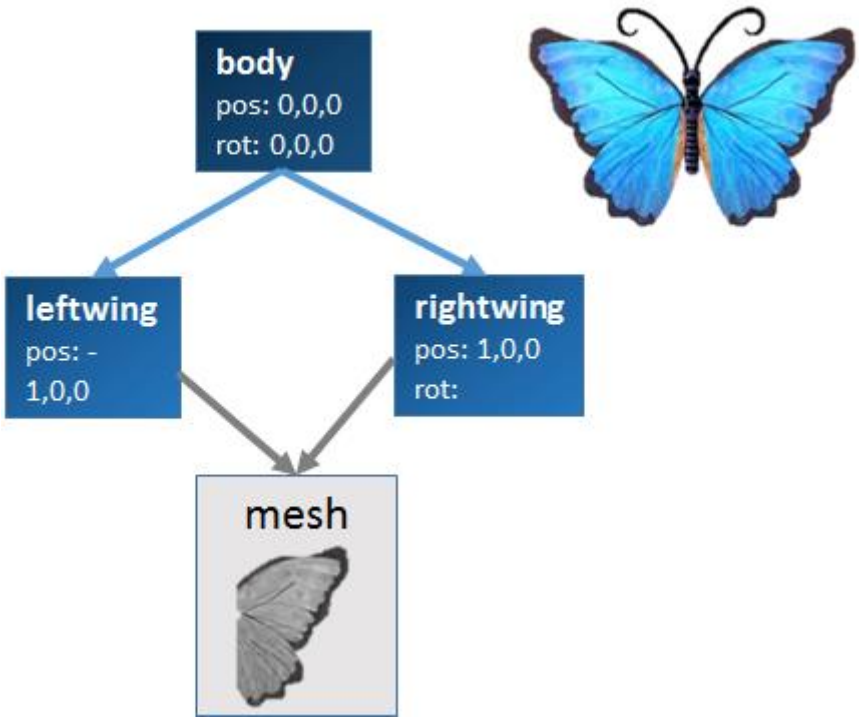

Рисунок 2.7 – Приклад графу сцени для метелика

Висновки

Багато чого залежить від вибору платформи та фреймворку для розробки додатку на мобільні платформи з підтримкою віртуальної реальності. Саме тому у даному розділі були проаналізовані та обрані для розробки модулю, що

відповідає за віртуальне середовище, наявні інструменти для розробки.

З огляду на особливості різних підходів при розробці програмного забезпечення з підтримкою VR вибір візуалізатора необхідно засновувати на таких факторах: вимоги до продуктивності ПЗ, оптимальний обсяг споживаних ресурсів, складність проекту і кваліфікованість програміста.

Доцільним є використання Unreal Engine, якщо продуктивність і якість картинки критичні для ПО. Також варто використовувати Unreal Engine, якщо буде існувати необхідність модифікації вихідного коду визуализатора. Якщо ж у проекту немає істотних вимог до якості візуальної частини та більш важливим є легкість розробки і розгортання, то слід подумати про використання Unity.

У випадку, коли не потрібна надмірна перевантаженість функціями та інтерфейсом, або не важливим є наявність редактора сцени слід звернути увагу на Gear Vr Framework. Розробка програмного продукту за допомогою цього фреймворка ведеться у Android Studio, що дозволяє самостійно додавати сторонні бібліотеки та модулі при необхідності, не потребуючи зайвих плагінів.

3 АНАЛІЗ ТЕХНОЛОГІЙ КОМП'ЮТЕРНОГО ЗОРУ ДЛЯ ВІДСТЕЖЕННЯ РУХУ

Вступ

Визначення руху і стеження за об'єктом є актуальним завданням для систем комп'ютерного зору. На сьогоднішній день для вирішення завдання стеження запропоновано багато підходів.

Візуальне відстеження полягає в послідовному визначенні місця розташування цільового об'єкта на кожному кадрі відеопотоку. Основною метою алгоритму відстеження є побудова траєкторії руху об'єкта, але також він повинен надавати інформацію про область зображення, яку займає об'єкт в кожен момент часу (тобто на кожному кадрі відеозапису).

Завдання розпізнавання об'єкта і зіставлення виявлених об'єктів на сусідніх кадрах можуть вирішуватися як окремо, так і спільно. У першому випадку можливі області, в яких може перебувати об'єкт, визначаються за допомогою алгоритму розпізнавання, і завдання трекера полягає в побудові відповідності виявлених областей з результатами з попереднього кадру. Такий підхід неформально визначається як відстеження на основі розпізнавання. У другому випадку область об'єкта і встановлення відповідності обчислюються спільно шляхом ітеративного поновлення положення об'єкта та інформації про область яку він займав на основі даних, отриманих на попередніх кадрах. Такий підхід називається розпізнаванням на основі відстеження. Він надає більш широкі можливості в порівнянні з першим підходом, оскільки дозволяє враховувати всю інформацію, отриману в ході відстеження, а не тільки дані з попереднього кадру.

3.1 Алгоритми відстеження руху

Відстеження об'єкта в часі зазвичай включає знаходження його області інтересів Obt (Xt , Yt) на послідовності кадрів з використанням таких характерних ознак, як точки, лінії або області.

Алгоритми відстеження областей застосовуються для відстеження об'єктів у відповідності до змін областей зображення, відповідних рухомим об'єктам. У таких алгоритмах зображення заднього фону динамічно оновлюється, а області руху зазвичай виділяються шляхом віднімання фону з поточного зображення [19].

Хоча вони успішно аналізують сцени, які містять невелику кількість об'єктів,алгоритми відстеження областей не можуть надійно обробляти випадки взаємного перекриття об'єктів. Більш того, оскільки ці алгоритми дають результати відстеження нарівні областей і по суті є процедурами виявлення руху, вони не дозволяють отримати обриси або тривимірне положення об'єкта(тривимірне положення об'єкта включає позицію і орієнтацію об'єкта в просторі). Відповідно, ці алгоритми не застосовні в умовах динамічно мінливого фону і великої кількості об'єктів, що рухаються.

Алгоритми відстеження по активному контуру відстежують об'єкти шляхом подання їх обрисів у вигляді обмежуючих контурів та динамічного оновлення цих контурів на наступних кадрах. Ці алгоритми мають своєю метою пряме вилучення форми об'єктів і дають більш повний опис об'єктів у порівнянні з алгоритмами відстеження областей.

На відміну від алгоритмів відстеження областей, алгоритми відстеження по активному контуру описують об'єкти більш просто і ефективно, а також скорочують обчислювальну складність. Вони можуть безперервно відстежувати об'єкти навіть в умовах спотворень і часткового перекриття об'єктів. Однак точність відстеження обмежується на рівні контуру. Відновлення тривимірного положення об'єкта з його контурів на плоскому зображенні є дуже складним завданням. Крім того, алгоритми відстеження по активному контуру дуже чутливі до точності початкової ініціалізації відстеження, що ускладнює запуск процесу відстеження в автоматичному режимі.

Алгоритми відстеження за характерними ознаками виконують розпізнавання і відстеження об'єктів шляхом вилучення елементів зображення, їх об'єднання в характерні ознаки більш високого рівня і наступного

зіставлення з характерними ознаками інших зображень. Алгоритми відстеження за характерними ознаками можна розділити на три класи відповідно до природи використовуваних ознак: алгоритми на основі глобальних ознак, алгоритми на основі локальних ознак, алгоритми на основі графів залежностей. Ознаки, що використовуються в алгоритмах на основі глобальних ознак, включають центр тяжіння, периметр, площа, колірні розподілення. Ознаки, які використовуються в алгоритмах на основі локальних ознак, включають прямолінійні і криволінійні сегменти, вершини кутів. Ознаки, що використовуються в алгоритмах на основі графів залежностей, включають набір відстаней і геометричних відносин між характерними ознаками.

У загальному випадку, оскільки вони застосовуються до двовимірних зображень, алгоритми відстеження за характерними ознаками можна легко і швидко адаптувати для обробки і відстеження декількох об'єктів в реальному часі, що необхідно для моніторингу перевантажених автострад. Такі алгоритми можуть справлятися з частковим перекриттям об'єктів, використовуючи інформацію про рух об'єкта, локальних ознаках і графах залежностей. Однак ці алгоритми мають ряд серйозних обмежень. По-перше, точність розпізнавання об'єктів на основі двовимірних ознак зображень є невисокою через нелінійних спотворень при проектуванні і варіації зображення при зміні точки огляду. По-друге, ці алгоритми в загальному випадку не дозволяють відновити тривимірне положення об'єктів.

Алгоритми відстеження по моделі відстежують об'єкти шляхом зіставлення ділянок зображення проекцій тривимірних моделей цих об'єктів, складених за апріорними даними. Моделі зазвичай будуються попередньо за допомогою ручних вимірювань, інструментів автоматизації проектних робіт або методів комп'ютерного зору. Для відстеження транспортних засобів зазвичай використовуються моделі у вигляді тривимірних «дротяних» каркасів.

У порівнянні з іншими алгоритмами відстеження алгоритми відстеження по моделі мають наступні основні переваги. По-перше, завдяки використанню апріорних знань про тривимірні контури або поверхні об'єктів, ці алгоритми є

початково стійкими до помилок. Вони можуть отримувати кращі результати навіть в умовах перекриття об'єктів і зближення кількох осередків руху на зображенні. По-друге, після встановлення геометричної відповідності між двовимірними координатами зображення і тривимірними просторовими координатами шляхом калібрування камери алгоритм природним чином отримує тривимірне положення об'єктів. По-третє, алгоритми відстеження по моделі застосовні навіть у випадках, коли об'єкти сильно змінюють свою орієнтацію в процесі руху. До недоліків алгоритмів цього класу можна віднести необхідність побудови моделей об'єктів, високу обчислювальну складність [20].

3.2 Бібліотека комп’ютерного зору з відкритим висхідним кодом OpenCV

OpenCV - de facto найпопулярніша бібліотека комп'ютерного зору. Вона написана на C / C ++, її вихідний код відкритий. бібліотека включає більше 1000 функцій і алгоритмів. Вона розробляється с 1998 року, спочатку в компанії Інтел, тепер в Itseez за активної участі спільноти. Про високу популярність бібліотеки свідчить кількість завантажень, їх більше 6000000 (без урахування svn / git трафіку).

Бібліотека розповсюджується за ліцензією BSD, що означає, що її можна вільно і безкоштовно використовувати як у відкритих проектах з відкритим кодом, так і в закритих, комерційних проектах. Бібліотеку не обов'язково копіювати цілком в свій проект, можна використовувати шматки коду. Єдина вимога ліцензії - наявність в супроводжуючих матеріалах копії ліцензії OpenCV [21].

Через ліберальну ліцензію бібліотека використовується багатьма компаніями, організаціями, університетами, наприклад, NVidia, Willow Garage, Intel, Google, Stanford University. Компанії NVidia і WillowGarage частково спонсують її розробку.

Невеликий перелік проектів, в яких використовується OpenCV:

- Система зору робота PR2, розробленого компанією WillowGarage. Проект

PR2 націлений на вирішення складної задачі - створення платформи для персональної робототехніки.

- Аудіо-візуальна інсталяція в Музеї сучасного мистецтва (Сан-Франциско).
- Контроль якості монет, виготовлених Центробанком Китаю.
- Курси комп'ютерного зору в Стенфорді.
- Панорами вулиць в картах Google.

Багатовимірна архітектура проекту представлена на рис. 3.1. Бібліотека складається з 16 модулів. Реалізовано близько 1000 алгоритмів. Функціональність доступна на різних мовах: C, C ++, Python, CUDA, Java. Підтримуються основні операційні системи: MS Windows, Linux, Mac, Android, iOS. Є можливість використання сторонніх бібліотек, наприклад, для роботи з пристроєм Kinect (OpenNI), розробки паралельних програм (TBB) і ін.

Проект використовує систему безперервної інтеграції. Щоночі останні робочі версії OpenCV викачуються зі сховищ і компілюються на 50 різних конфігураціях (різні операційні системи, різні платформи, різні параметри і т. Д.), Після чого запускаються регресивні тести. Наприклад, проводиться реальне тестування на Android-планшетах.

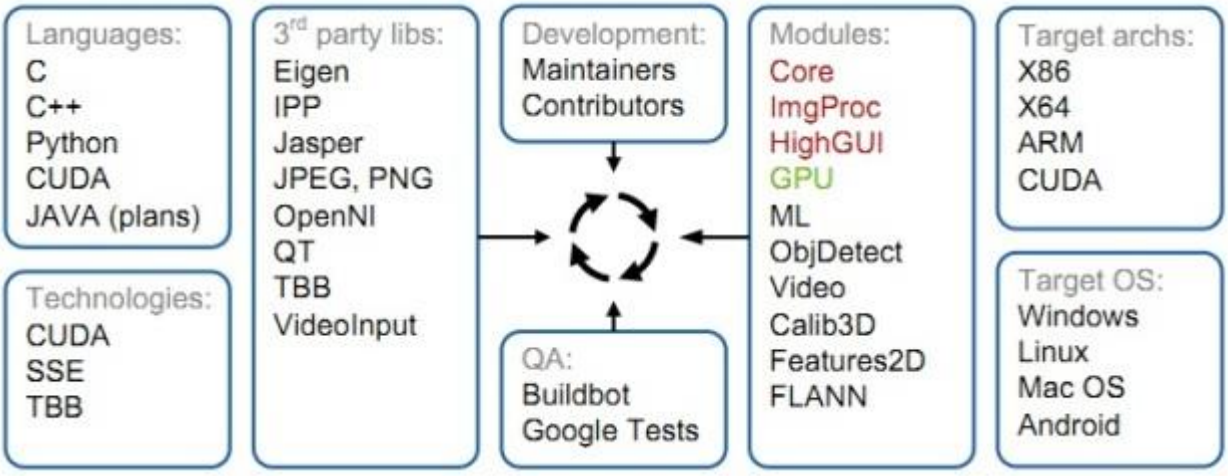

Рисунок 3.1 – Багатовимірна архітектура проекту

Основні модулі бібліотеки можна віднести до 4 групи (розділи):

- Модулі core, highgui, що реалізують базову функціональність (базові структури, математичні функції, генератори випадкових чисел, лінійна

алгебра, швидке перетворення Фур'є, введення / виведення зображень і відео, введення / виведення в форматах XML, YAML і ін.).

- Модулі imgproc, features2d для обробки зображень (фільтрація, геометричні перетворення, перетворення колірних просторів, сегментація, виявлення особливих точок і ребер, контурний аналіз і ін).
- Модулі video, objdetect, calib3d (калібрування камери, аналіз руху і відстеження об'єктів, обчислення координат в просторі, побудова карти глибини, детектування об'єктів, оптичний потік).
- Модуль ml, який реалізує алгоритми машинного навчання (метод найближчих сусідів, наївний байесовский класифікатор, дерева рішень, бустінг, градієнтний бустінг дерев рішень, випадковий ліс, машина опорних векторів, нейронні мережі та ін.).

Результати проміжних обчислень можна зберігати в xml-файлах, а потім їх прочитати, наприклад, в іншій програмі, що полегшує розробку алгоритму по частинах, роботу над алгоритмом цілою командою.

Бібліотека OpenCV реалізує, як правило, тільки базові операції, які використовуються в комп'ютерному зорі. Таким чином, її можна розглядати як в цілому низькорівневу бібліотеку комп'ютерного зору. Для вирішення серйозних завдань необхідно на основі наданих бібліотекою інструментів створювати свої складні додатки [22].

На рисунку 3.2 представлена загальна схема типового додатку, призначеного для вирішення того чи іншого завдання комп'ютерного зору:

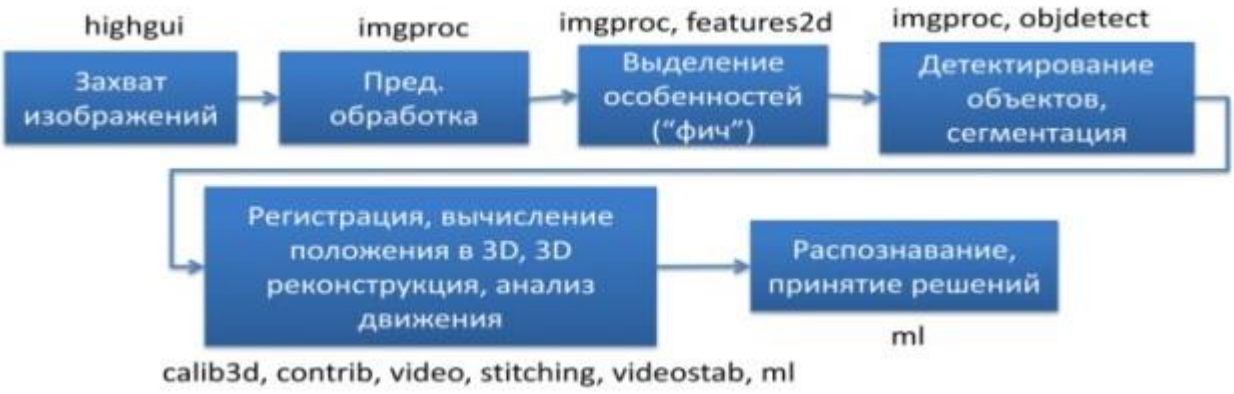

Рисунок 3.2 – Загальна схема типового додатку

Все починається з захоплення зображень (модуль highgui). Можна зчитати зображення з файлу або відео з мережевої камери через мережевий протокол.

Далі здійснюється попередня обробка (модуль imgproc), така, як усунення шуму, вирівнювання яскравості, контрасту, виділення і видалення відблисків, тіней. Наприклад, один і той же об'єкт при різному освітленні виглядає по-різному. У яскравому світлі червона машина, рух якої, необхідно відстежувати, буде яскраво-оранжевою. У похмуру погоду та ж машина буде виглядати червоно-рожевою. У цьому випадку на зображенні необхідно виконати вирівнювання кольорів. Передобробка може бути простою, але може містити в собі цілу складну технологію.

Наступний етап - виділення особливостей (модулі imgproc, features2d). Наприклад, в завданні стеження за об'єктом це може бути пошук спеціальних точок на об'єкті, за якими легко спостерігати; для завдання детектування (т. е. виявлення на зображенні) особи - обчислення опису кожного пікселя.

Далі відбувається детектування об'єктів, що нас цікавлять, виділення значущих частин, сегментація зображення (модулі imgproc, objdetect). Якщо, наприклад, камера нерухома, а зображення рухоме, можна використовувати алгоритми віднімання фону.

Після цього вирішується основне завдання, таке, як обчислення розташування об'єкта в 3d, реконструкцію 3d структури, аналіз структури, реєстрацію і т. п. (Модулі calib3d, contrib, video, stitching, videostab, ml).

В кінці відбувається розпізнавання і прийняття конкретних рішень (модуль ml). Наприклад, в системі відеоспостереження: з'явився небажаний об'єкт в кадрі чи ні. У задачі детектування тексту - детектувати текст, що саме за текст і т. д.

3.3 Детектування руху за допомогою оцінки оптичного потоку засобами OpenCV

Оптичний потік (ОП) - зображення видимого руху, що представляє собою зсув кожної точки між двома зображеннями. По суті, він являє собою поле швидкостей (так як зрушення з точністю до масштабу еквівалентне миттєвій швидкості). Суть ОП в тому, що для кожної точки зображення $I_1(x, y)$ знаходиться такий зсув (dx, dy), щоб вихідній точці відповідала точка на іншому зображенні $I_2(x + dx, y + dy)$. Як визначити відповідність точок - окреме питання. Для цього треба взяти якусь функцію точки, яка не змінюється в результаті зсуву. Зазвичай вважається, що у точки зберігається інтенсивність (тобто яскравість або колір для кольорових зображень), але можна вважати однаковими точки, у яких зберігається величина градієнта, гессіан, його величина або його визначник, лапласіан, інші характеристики. Очевидно, збереження інтенсивності дає збої, якщо змінюється освітленість або кут падіння світла. Проте, якщо мова йде про відеопотік, то, швидше за все, між двома кадрами освітлення сильно не зміниться, хоча б тому, що між ними проходить малий проміжок часу. Тому часто використовують інтенсивність в якості функції, що зберігається у точки.

Є два варіанти розрахунку оптичного потоку: щільний (dense) і вибірковий (sparse). Sparse потік розраховує зсув окремих заданих точок (наприклад, точок, виділених деяким feature detector'ом), dense потік рахує зсув всіх точок зображення. Звісно, вибірковий потік обчислюється швидше, однак для деяких алгоритмів різниця не така вже й велика, а для деяких завдань необхідним є знаходження потоку у всіх точках зображення.

Алгоритм Лукаса-Канаде. Розглянемо математичну модель оптичного потоку, вважаючи, що у точки в результаті зсуву не змінилася інтенсивність.

Нехай $I_1 = I(x, y, t_1)$ - інтенсивність в деякій точці (x, y) на першому зображенні (в момент часу t). На другому зображенні ця точка зсунулась на (dx, dy), при цьому пройшов час dt, тоді $I_2 = I(x + dx, y + dy, t_1 + dt) \approx$

$I_1 + I_x dx + I_y dy + I_t dt$ - це розкладання по Тейлору функції інтенсивності до першого члена, тут I_x, I_y, I_t - часткові похідні по координатах і часу, тобто по суті $I_t dt$ - зміна яскравості в точці (x, y) між двома кадрами.

Ми вважаємо, що у точки збереглася інтенсивність, а отже $I_x dx + I_y dy + I_t dt = 0$. Отримуємо одне рівняння з двома невідомими (dx і dy), тобто його недостатньо для вирішення.

Найпростіше рішення проблеми - алгоритм Лукаса-Канаде. На зображенні знаходяться об'єкти розміром більше 1 пікселя, а отже, швидше за все, в околиці поточної точки у інших точок будуть приблизно такі ж зсуви. Тому ми візьмемо вікно навколо цієї точки і мінімізуємо (по МНК) в ньому сумарну похибку з ваговими коефіцієнтами, розподіленими по Гаусу, тобто так, щоб найбільшу вагу мали пікселі, що знаходяться найближче до досліджуваного. Після найпростіших перетворень, отримуємо вже систему з 2 рівнянь з 2 невідомими:

$$\sum_{i,j} g(i,j) I_x(x_i, y_j)^2 dx + \sum_{i,j} g(i,j) I_x(x_i, y_j) I_y(x_i, y_j) dy = -\sum_{i,j} g(i,j) I_x(x_i, y_j) I_t(x_i, y_j) dt;$$

$$\sum_{i,j} g(i,j) I_x(x_i, y_j) I_y(x_i, y_j) dx + \sum_{i,j} g(i,j) I_y(x_i, y_j)^2 dy = -\sum_{i,j} g(i,j) I_y(x_i, y_j) I_t(x_i, y_j) dt.$$

Як відомо, ця система має єдине рішення не завжди (хоча і дуже часто): якщо детермінант системи дорівнює нулю, то рішень або немає, або нескінченне число. Ця проблема відома як aperture problem - неоднозначність зсуву при обмеженому полі зору для періодичних картинок. Вона відповідає випадку, коли в поле зору потрапляє фрагмент зображення, в якому присутня деяка циклічність. Проблема в тому, що через шуми в таких неоднозначних ситуаціях ми отримаємо не нульовий детермінант, а дуже маленький, який, швидше за все, призведе до дуже великих значень зсуву, що не корелює з дійсністю. Так що на певному етапі потрібно перевіряти, чи не є детермінант системи занадто маленьким, і у такому випадку не розглядати такі точки або відзначати їх як помилкові [23].

Звичайний Лукас-Канаде добре визначає маленькі зсуви, такі, в рамках яких зображення схоже на своє лінійне наближення.

Щоб визначати і великі зрушення слід скористається стандартним прийомом CV - multi-scaling'ом: побудувати «піраміду» зображень різного масштабу і пройти по ним оптичним потоком від меншого зображення до більшого, тоді детектований маленький зсув на маленькому зображенні буде відповідати великому зсуву на великому зображенні. На найменшому зображенні ми виявляємо зсув не більше 1-2 пікселів, а переходячи від меншого масштабу до більшого, ми користуємося результатом з попереднього кроку і уточнюємо значення зсуву.

Алгоритм Farneback'а. Farneback запропонував апроксимувати зміну інтенсивності в околиці за допомогою квадратичної форми: I = xAx + bx + c із симетричною матрицею A. Якщо зображення зсунулося в межах цієї околиці, то $I_2(x) = I_1(x - d)$, підставляємо в квадратичний розклад, розкриваємо дужки, отримуємо:

$$A_2 = A_1;$$
$$b_2 = b_1 - 2A_1 d;$$
$$c_2 = d^T A_1 d - b_1^T d + c_1$$

Тепер ми можемо обчислити значення A, b, c на обох зображеннях і тоді ця система стане надлишковою відносно d.

Доводиться вдаватися до наступної апроксимації:

$$A = \frac{1}{2}(A_1 + A_2)$$

Позначимо для простоти:

$$\Delta b = -\frac{1}{2}(d_2 - d_1)$$

Тоді отримаємо $Ad = \Delta b$.

Для компенсації шумів при обчисленні, знову звернемося до припущення, що в околиці досліджуваної точки у всіх точок більш-менш однаковий зсув. Тому знову ж проінтегруємо похибку $|Ad - \Delta b|^2$ по вікну з гауссовськими ваговими коефіцієнтами w, і знайдемо вектор d, що мінімізує цю сумарну похибку. Тоді ми отримаємо оптимальне значення $d = (\sum wA^T A)^{-1} \sum wA^T \Delta b$ і відповідну мінімальну помилку

$e = (\sum w\Delta b^T \Delta b) - d^T \sum w A^T \Delta b$. Тобто нам треба для кожної точки порахувати $A^T A, A^T \Delta b, \Delta b^T \Delta b$, усереднити по вікну, інвертувати матрицю і отримати результат. Відповідно ці добутки можна порахувати для всього зображення і використовувати заздалегідь розраховані значення для різних точок, тобто це як раз той випадок, коли має сенс рахувати dense потік [24].

У цього алгоритму є деяка кількість модифікацій і удосконалень, які в першу чергу дозволяють використовувати відому апріорну інформацію - задану початкову апроксимацію потоку і, знову ж таки, multi-scaling.

Метод SimpleFlow. Для всіх точок у вікні знаходиться функція «енергії», що відповідає (із зворотньою логарифмічною залежністю) за ймовірність переходу вихідної точки в цю точку: $e(x, y, dx, dy) = |F_1(x, y) - F_2(x + dx, y + dy)|^2$. Далі, рахується згортка цієї енергії з гауссових вікном:

$$E(x0, y0, dx, dy) = \sum (e(x, y, dx, dy) exp(-\frac{|(x, y) - (x_0, y_0)|^2}{2s_d}) exp(-\frac{|F_1(x, y) - F_1(x0, y0)|^2}{2s_c})$$

Знаходяться значення (dx, dy), які мінімізують цю функцію. Щоб отримати субпіксельну точність, розглядається мала околиця знайденої оптимальної точки (dx, dy) і в ній шукається пік функції енергії як пік параболоїда. І, як було згадано вище, ця процедура виконується для піраміди масштабування зображень [25].

Висновки

В даному розділі було проведено аналіз бібліотеки комп’ютерного зору OpenCV та загальної схеми типового додатку реалізованого за допомогою цієї бібліотеки.

Крім цього було проаналізовано існуючі алгоритми комп’ютерного зору для трекінгу руху об’єкта у відеопотоці. Якщо достатньо порахувати зсув деякого набору точок (sparse потік), то алгоритм Лукаса-Канаде працює добре, точно і досить швидко.

Для обчислення dense-потоку добре використовувати алгоритм

SimpleFlow, але він працює дуже повільно.

Якщо швидкодія є критичною, то для обчислення dense-потоку слід використовувати алгоритм Farneback’a, але необхідно упевнитися, що точність його роботи є достатньою.

4 РОЗРОБКА ДОДАТКУ З ПІДТРИМКОЮ ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ ТА МОЖЛИВІСТЮ КЕРУВАННЯ ЗА ДОПОМОГОЮ ЖЕСТІВ ДЛЯ ТЕСТУВАННЯ НАДІЙНОСТІ

4.1 Загальний алгоритм роботи додатку

Ідея програмного продукту полягала в тому, щоб розробити додаток для мобільних пристроїв на базі OS Android з підтримкою віртуальної реальності, який би дозволяв керувати якимось контентом за допомогою жестів, які отримувалися б з відеопотоку стандартної камери пристрою.

Додаток складається з двох модулів:

1. Модуль, який відповідає за графічну частину (віртуальне середовище).
2. Модуль комп’ютерного зору, який відповідає за виділення необхідних даних з відеопотоку.

На рисунку 4.1 можна побачити алгоритм роботи модуля віртуального середовища. Як видно, спочатку ініціалізується 3D-сцена, усі необхідні змінні, поля класу, тощо. Після цього відбувається завантаження об’єктів сцени, та необхідних для них текстур. І, врешті, починається цикл рендерингу, який дозволяє користувачу взаємодіяти з навколишнім середовищем, відмальовує кожний кадр під час зміни положення голови і т. д. Під час роботи цього циклу можна отримувати дані з модуля комп’ютерного зору і в залежності від них робити певні дії з контентом.

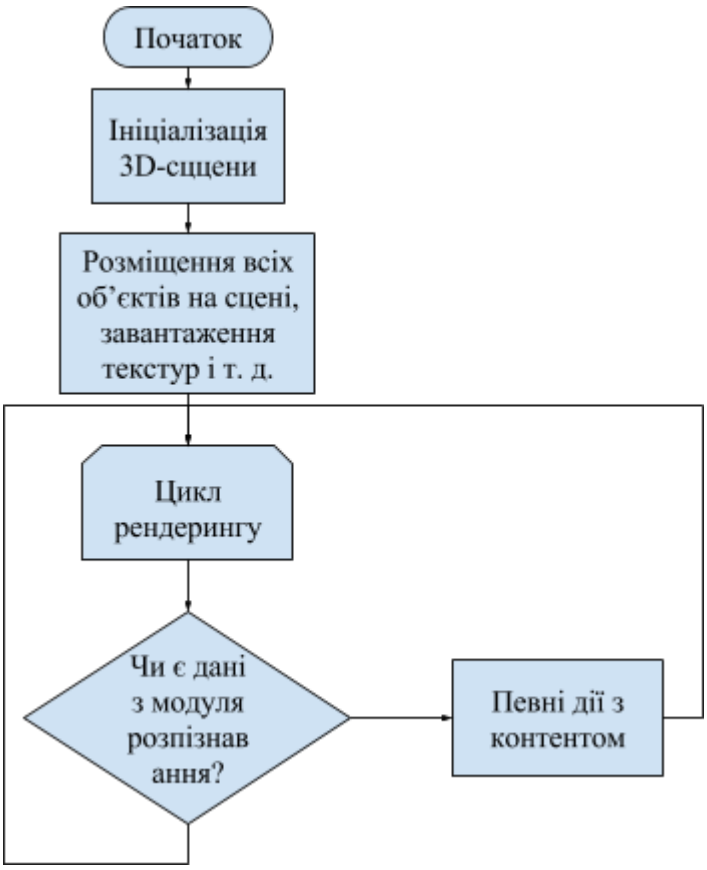

Рисунок 4.1 – Алгоритм роботи модулю віртуальної реальності

На рисунку 4.2 можна побачити алгоритм роботи модулю, що відповідає за комп'ютерний зір. Першим кроком відбувається ініціалізація камери та усі необхідні поля класу, що відповідає за роботу з розпізнаванням. Далі запускається цикл обробки кадрів з камери. В цьому циклі з стандартної камери мобільного пристрою отримується зображення у даний момент часу, після чого на ньому знаходяться ключові точки, які будуть використовуватися для вирахування зміщення на наступному кадрі.

Наступним кроком роботи алгоритму є отримання даних про зміщення усіх ключових точок та його напрямок за допомогою алгоритму Лукаса-Канаде. Після цього кроку відбувається усереднення результатів і знаходиться середній напрямок та величина зміщення. Величина зміщення також згладжується за допомогою метода рухомого середнього для зменшення ривків між цими послідовними величинами.

Після цього відбувається компенсація зміни положення голови користувача, адже вона може давати похибку або неправильні результати до

остаточного значення зміщення, отриманого за допомогою оцінки оптичного потоку.

І, нарешті, фінальні дані потрапляють до модуля який відповідає за віртуальне середовище.

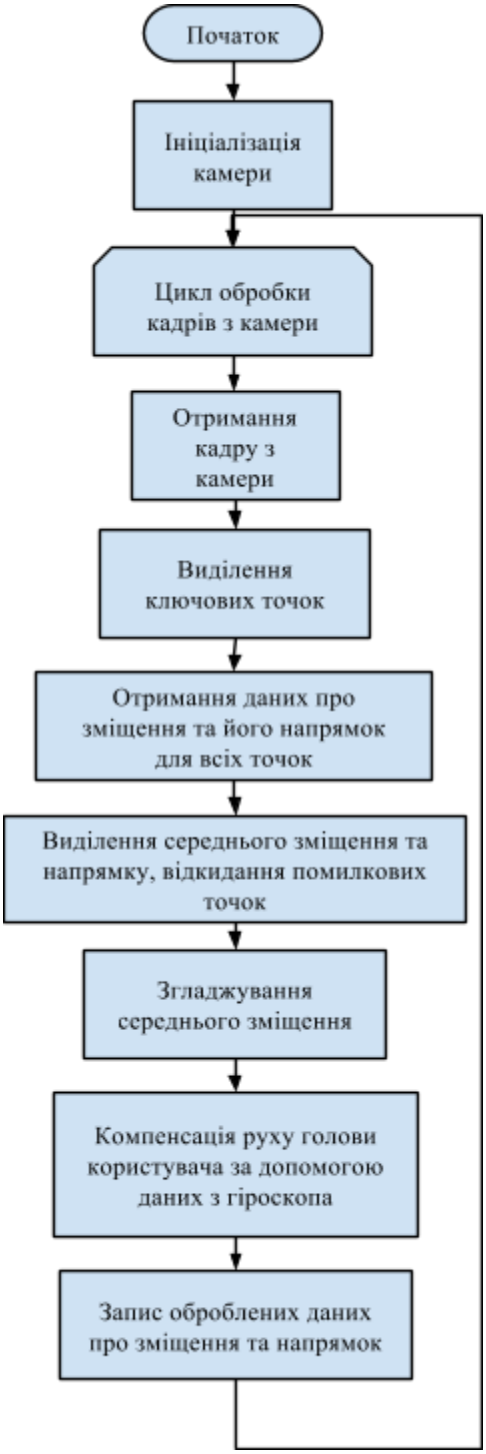

Рисунок 4.2 – Алгоритм роботи модулю комп’ютерного зору

4.2 Реалізація модулю, що відповідає за VR-середовище

Для реалізації модулю, що відповідає за графічну складову проекту було проведено аналіз усіх фреймворків, які дозволяють розробляти графічні додатки для плаформи Android з підтримкою технології віртуальної реальності.

На жаль, такі фреймворки, як Unity та Unreal Engine мають власні IDE для створення програмнного продукту і не підтримують сторонній Java-код, необхідний для реалізації модуля відстеження жестів. Також для них немає якісних безкоштовних плагінів, які б дозволили підключити OpenCV.

Саме тому фреймворком для реалізації графічної складової проекту було обрано Gear VR Framevork, який має вбудовану підтримку Oculus Mobile SDK.

Модуль з підтримкою віртуальної реальності було реалізовано на мові програмування Java в середовищі Android Studio.

Для початку роботи з Gear VR Framevork достатньо скачати бібліотеки з офіційного сайту, підключити їх до проекту і після цього можна приступати до розробки.

Для реалізації додатку з підтримкою віртуальної реальності було змінено Activity клас так, щоб він наслідував GVRActivity (замість android.app.Activity) і змінено метод OnCreate() так, щоб він викликав setMain() замість setContentView().

Далі було створено клас, який наслідує GVRMain. Цей клас відповідає за ініціалізацію 3D-сцени, створення усіх об’єктів, завантаження текстур, шейдерів, тощо.

Також у цьому класі знаходиться метод, який викликається для кожного кадру і потрібен для їх обробки.

В якості демонстрації роботи програмного продукту було створено віртуальне середовище, яке має вигляд космосу навколо користувача із планетою, що знаходиться перед ним. На рисунку 4.3 можна побачити віртуальне середовище очами користувача.

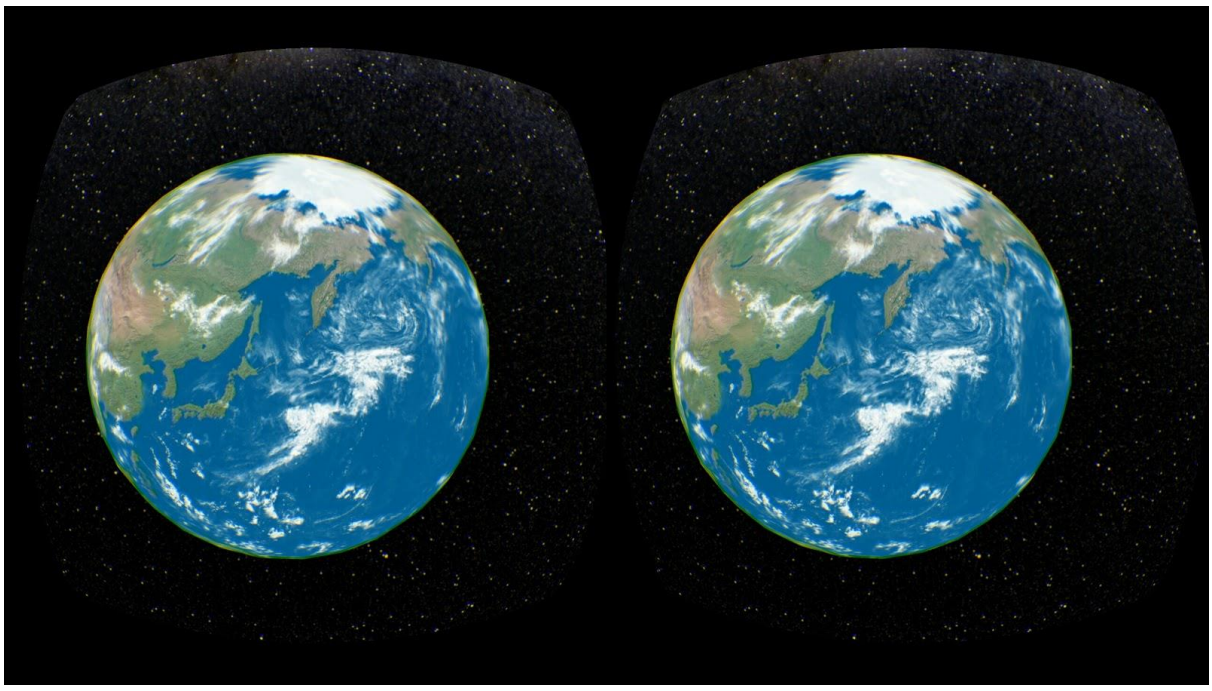

Рисунок 4.3 – Приклад віртуального середовища очами користувача

Об’єкт планети має інтерфейс, який дозволяє керувати його поворотом за допомогою даних, які отримуються при обробці відеопотоку.

Об’єкти космосу і планети є екземплярами одного і того ж класу, і являють собою видимі сфери з текстурами. Текстури представляють собою еквідистантну циліндричну проекцію. Проекція відображає меридіани на вертикальні лінії з однаковою відстанню, а паралелі на горизонтальні лінії з однаковою відстанню. Приклад такої проекції можна побачити на рисунку 4.4.

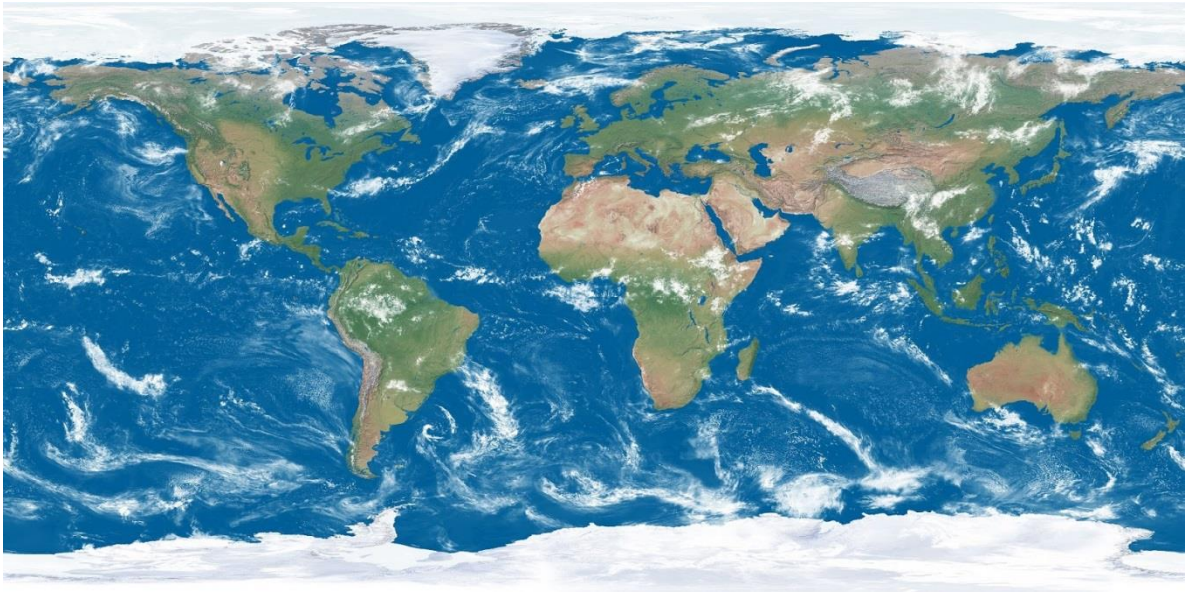

Рисунок 4.4 – Приклад еквідистантної циліндричної проекції

4.3 Використання OpenCV у Android Studio

Всім давно відомо, що мобільні пристрої все частіше комплектуються потужним апаратним забезпеченням. Процесори сучасних смартфонів майже зрівнялися з настільними аналогами, а проблеми з можливою нестачею оперативної і зовнішньої пам'яті йдуть на другий план. Сьогодні вже не зустрінеш телефони або планшети без значної програмної платформи на зразок Android, IOS або Windows, а крім того, всі мобільні пристрої мають ті чи інші вбудовані модулі, на кшталт камери.

Для того, щоб використати OpenCV в даному програмному продукті з метою реалізації модулю комп’ютерного зору, необхідно завантажити відповідний SDK. Зробити це можна на офіційному сайті (http://opencv.org).

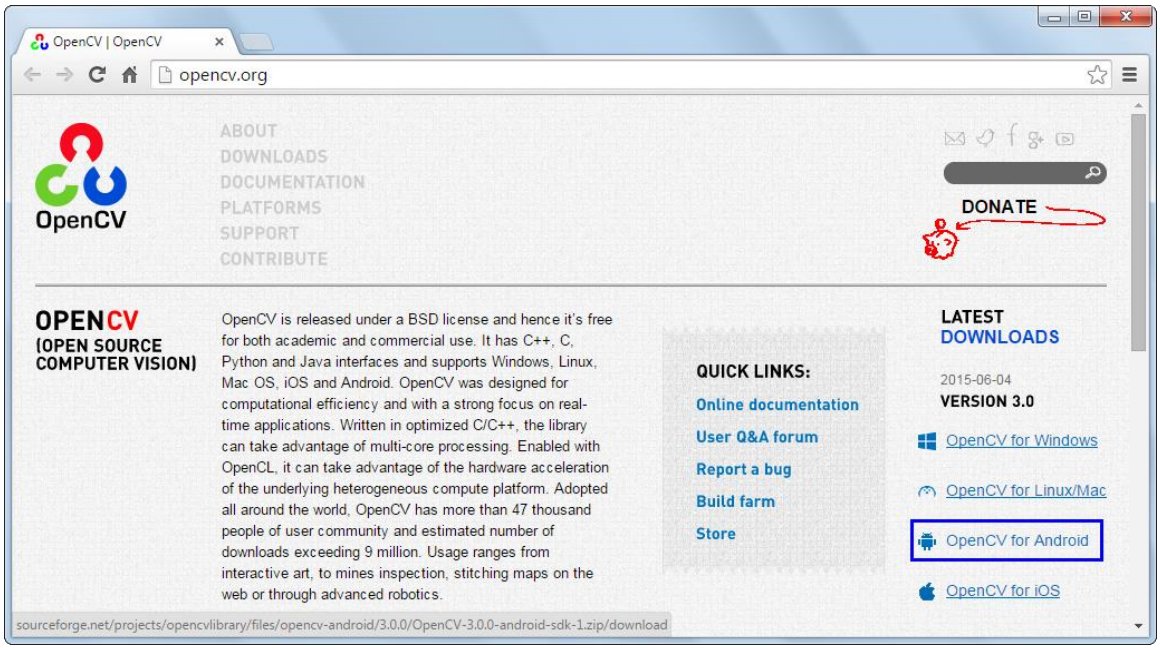

Рисунок 4.5 – Завантаження OpenCV

До архіву OpenCV SDK для Android входять виконавчі файли бібліотек під різні архітектури мікропроцесорів, обгортка на мові програмування Java, яка використовує Java Native Interface (JNI) для виконання функцій з платформо-залежних бібліотек, вихідні коди прикладів програм і файли .apk для установки OpenCV Manager. Наступний крок - додавання модулю OpenCV до проекту Android Studio.

Для імпорту модулю OpenCV в проект Android studio потрібно вибрати меню File -> New -> Import Module і вказати шлях sdk/java з розпакованого

архіву SDK. Всі інші налаштування імпорту міняти не обов'язково. На рисунку 4.6 показано вікно майстра із зазначенням шляху імпорту модуля і його нового імені в проекті.

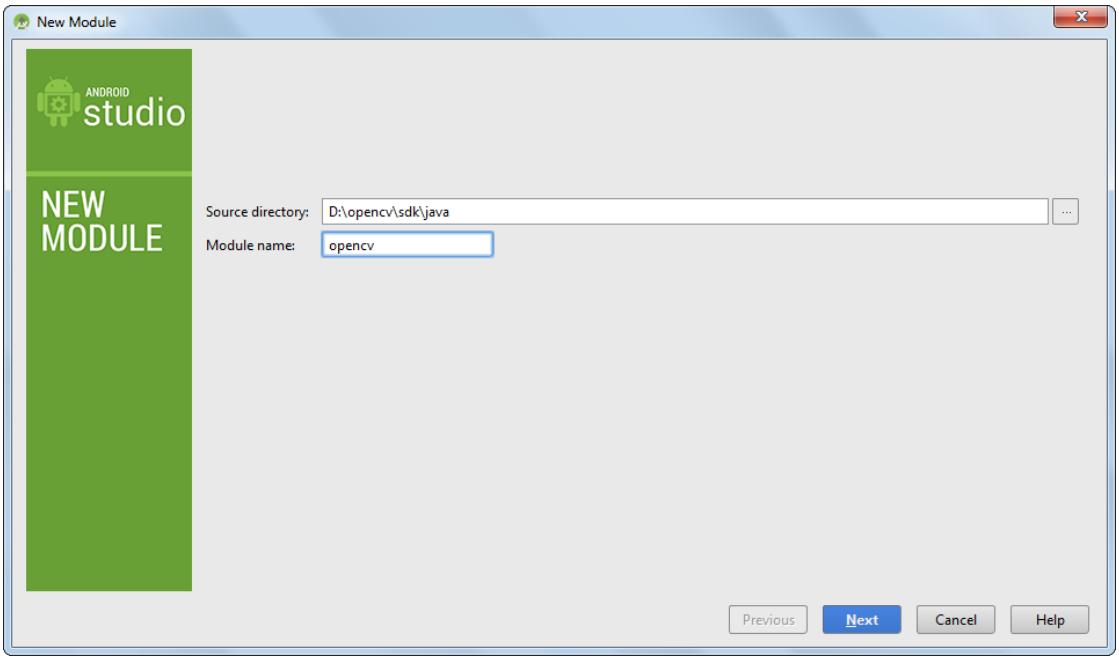

Рисунок 4.6 – Вікно майстра із зазначенням шляху імпорту модуля

Після імпорту модуля в його build.gradle потрібно обов'язково оновити параметри compileSdkVersion, buildToolsVersion, minSdkVersion і targetSdkVersion, щоб вони збігалися з відповідними параметрами з модуля програми. Після того, як модуль OpenCV був доданий, його потрібно приєднати в якості залежності до модуля програми. Для цього потрібно вибрати меню File -> Project Structure ... і для модуля app вказати залежність від OpenCV. На малюнку 4.7 показано вікно налаштуванння залежностей:

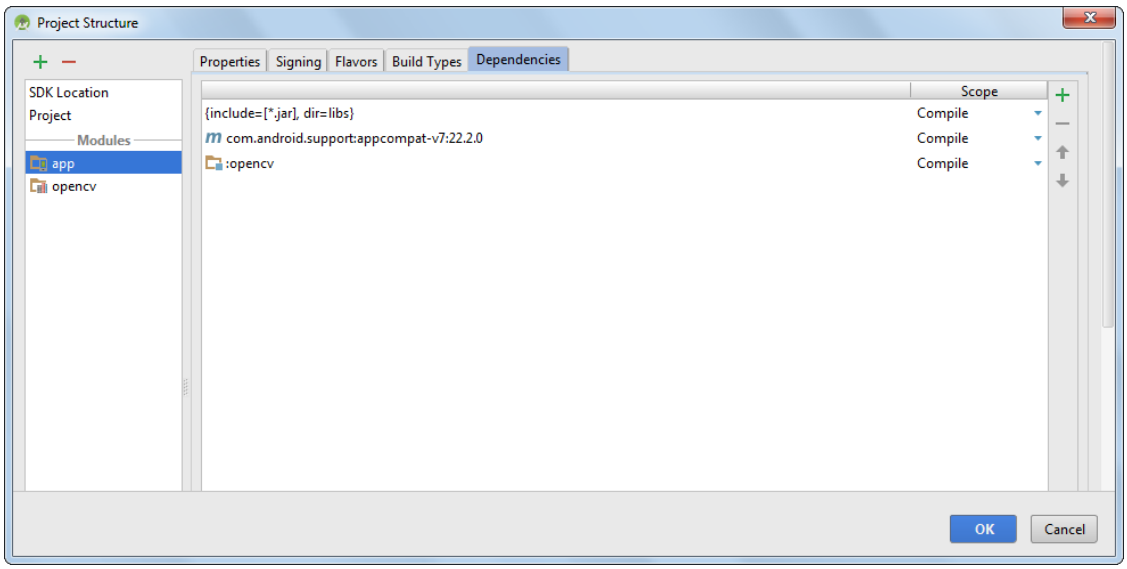

Рисунок 4.7 – Вікно налаштуваннь залежностей

Останнє, що залишилося зробити - додати виконавчі бібліотеки OpenCV в проект, щоб при виконанні збірки вони були упаковані в .apk додатка. Для цього потрібно скопіювати папку sdk/native/libs з OpenCV SDK в папку проекту по шляху app/src/main. Потім потрібно перейменувати папку libs в jniLibs. Слід зазначити, що не потрібно упаковувати статичні бібліотеки OpenCV (розширення .a) в .apk і зберігати їх в проекті. У створеній папці jniLibs для кожної архітектури можна залишити тільки файли з розширенням .so. На рисунку 4.8 показана отримана структура папок в Провіднику Windows.

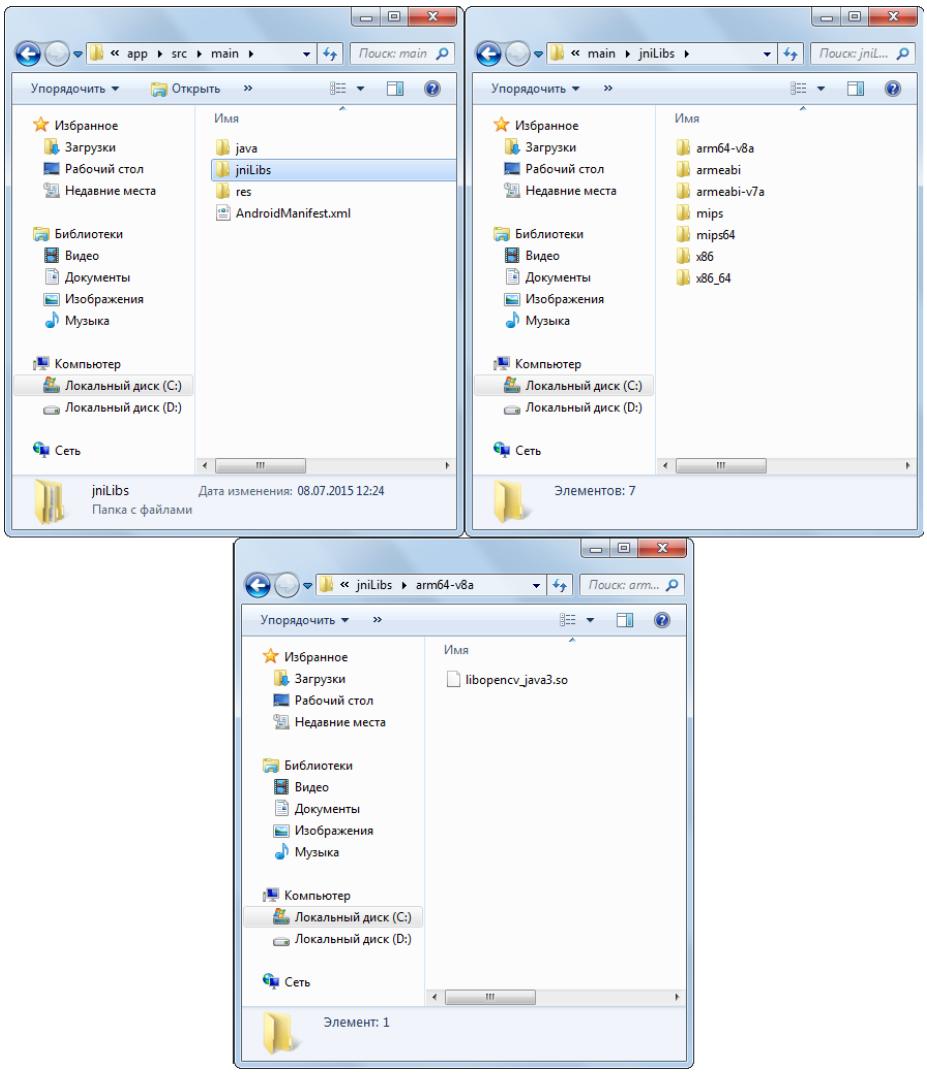

Рисунок 4.8 – Структура папок в Провіднику Windows

Якщо усі кроки були виконані, бібліотеку можна використовувати у серидовищі.

4.4 Реалізація модулю, що відповідає за відстеження руху за допомогою бібліотеки OpenCV

Модуль комп’ютерного зору було реалізовано на мові програмування Java в середовищі Android Studio. Під час його реалізації було випробувано декілька підходів до детекції руху у відеопотоці.

Для відстеження руху у sparse-потоці треба спочатку визначити точки на зображенні, для яких буде вираховуватися зміщення. Для цього можна скористатися або алгоритмами для виділення об’єкту на зображенні, або алгоритмами для детекції особливих точок зображення.

На жаль, як можна побачити на рисунку 4.9, при реалізації детекції руки на картинці був занадто низький FPS, а так як додаток повинен працювати у реальному часі, від цього методу довелося відмовитися.

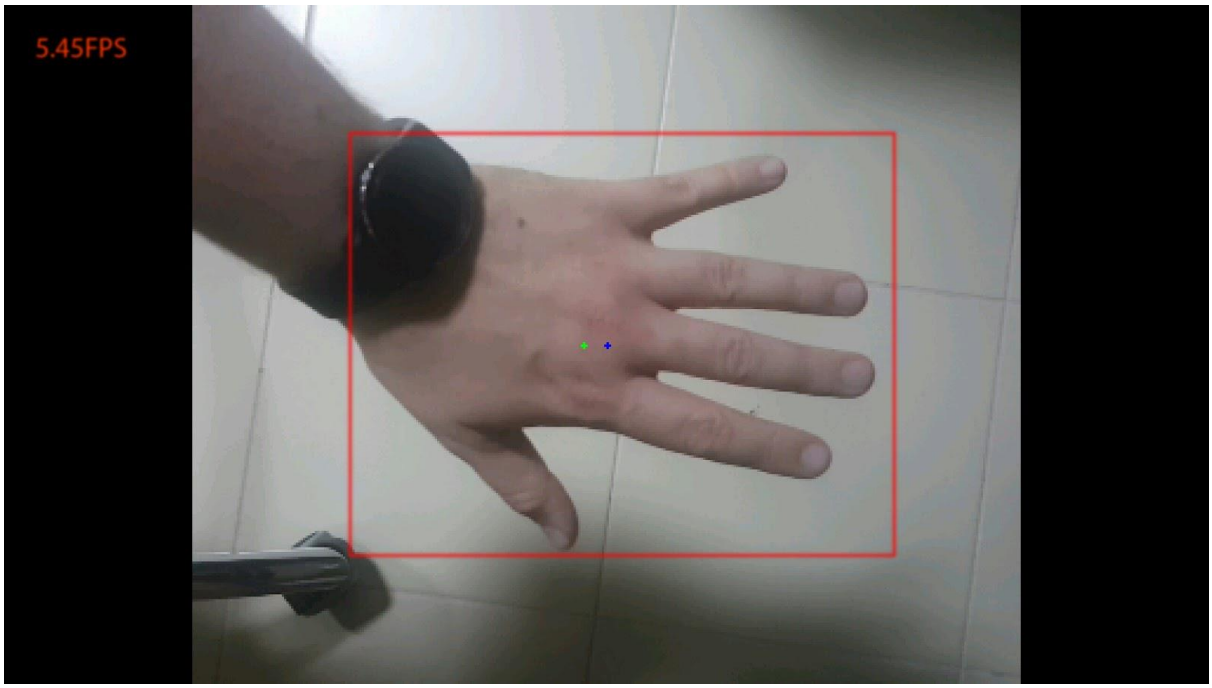

Рисунок 4.9 – Детекція руки

Замість цього було вирішено скористатися одним із алгоритмів визначення кутів. Кути (corners) - особливі точки, які формуються з двох або більше граней, і грані зазвичай визначають межу між різними об'єктами і / або частинами одного і того ж об'єкта. По-іншому можна сказати, що кути - це точки, у яких в околиці інтенсивність змінюється відносно центру (x, y). Кути визначаються по координатам і змінам яскравості навколишніх точок зображення. Головна властивість таких точок полягає в тому, що в області навколо кута у градієнта зображення переважають два домінуючих напрямки, що робить їх помітними. Градієнт - векторна величина, що показує напрямок найшвидшого зростання функції інтенсивності зображення I (x, y). Оскільки зображення дискретно, то вектор градієнта визначається через приватні похідні по осі x і y через зміни інтенсивностей сусідніх точок зображення.

Було використано алгоритм Shi-Tomasi - це модифікація алгоритму Harris'a. Міра відгуку в алгоритмі Harris'a описувалася наступною формулою:

$$R = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2$$

Замість цього в алгоритмі Shi-Tomasi було запропоновано:

$$R = \min(\lambda_1, \lambda_2)$$

Якщо це більше, ніж порогове значення, воно розглядається як кут. Якщо ми заносимо його в $\lambda_1 - \lambda_2$ простір, як ми робили в Harris Corner Detector, ми отримаємо зображення, як показано на рисунку 4.10:

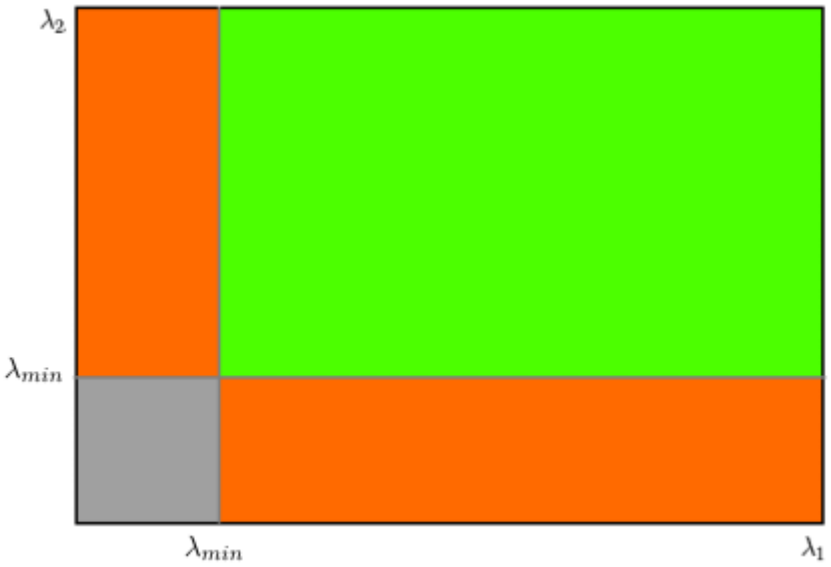

Рисунок 4.10 – Міра відгуку в алгоритмі Shi-Tomasi

З рисунку 4.10 можна побачити, що тільки тоді, коли λ_1 і λ_2 вище мінімального значення, λ_{min}, це вважається кутом (зелена область).

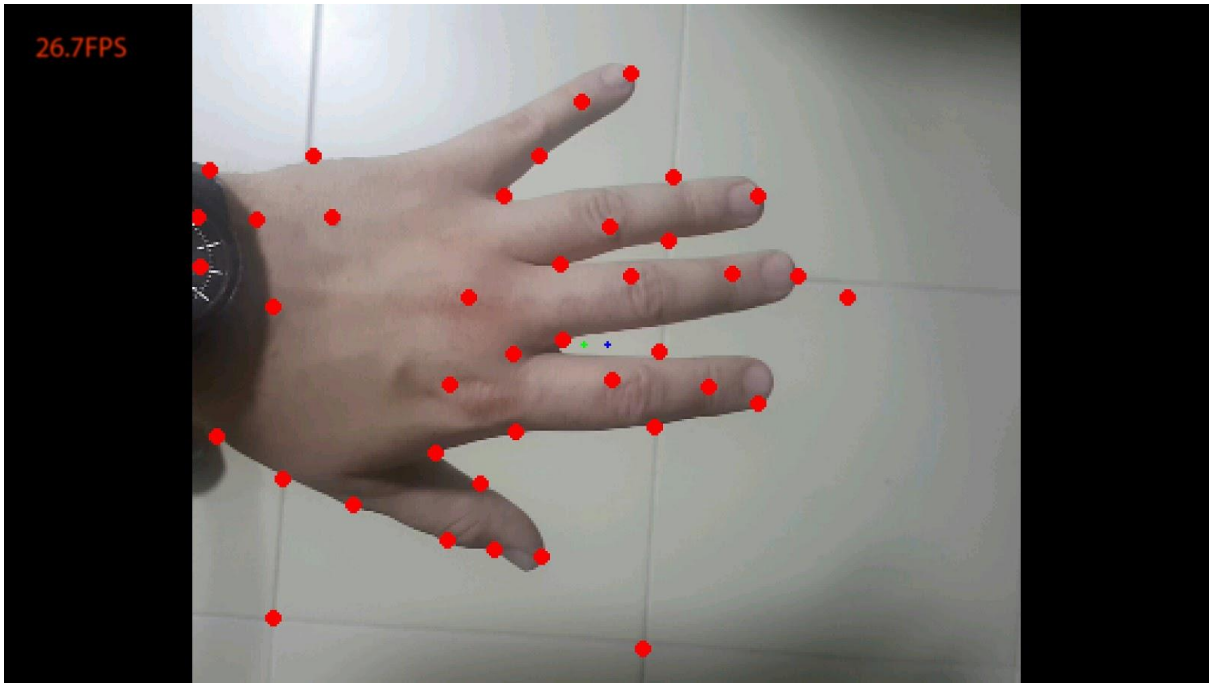

Рисунок 4.11 – Детекування кутів під час реалізації алгоритму Shi-Tomasi

На рисунку 4.11 можна побачити що програма досить добре виділяє особливі точки на межі пальців та фону, а також кути між пальцями та долонею. Також, з рисунку 4.11 видно, що FPS становить біля 27 кадрів на секунду, що є достатнім для роботи у реальному часі.

Для оцінки оптичного потоку було застосовано алгоритм Лукаса-Канаде. Основне рівняння оптичного потоку містить дві незалежні змінні і не може бути однозначно вирішеним. Алгоритм Лукаса-Канаде вирішує неоднозначність за рахунок використання інформації про сусідні пікселі в кожній точці. Метод заснований на припущенні, що в локальному оточенні кожного пікселя значення оптичного потоку однаково, таким чином можна записати рівняння оптичного потоку для всіх точок в околиці і вирішити систему рівнянь методом найменших квадратів. Результати алгоритму можна побачити на рисунку.

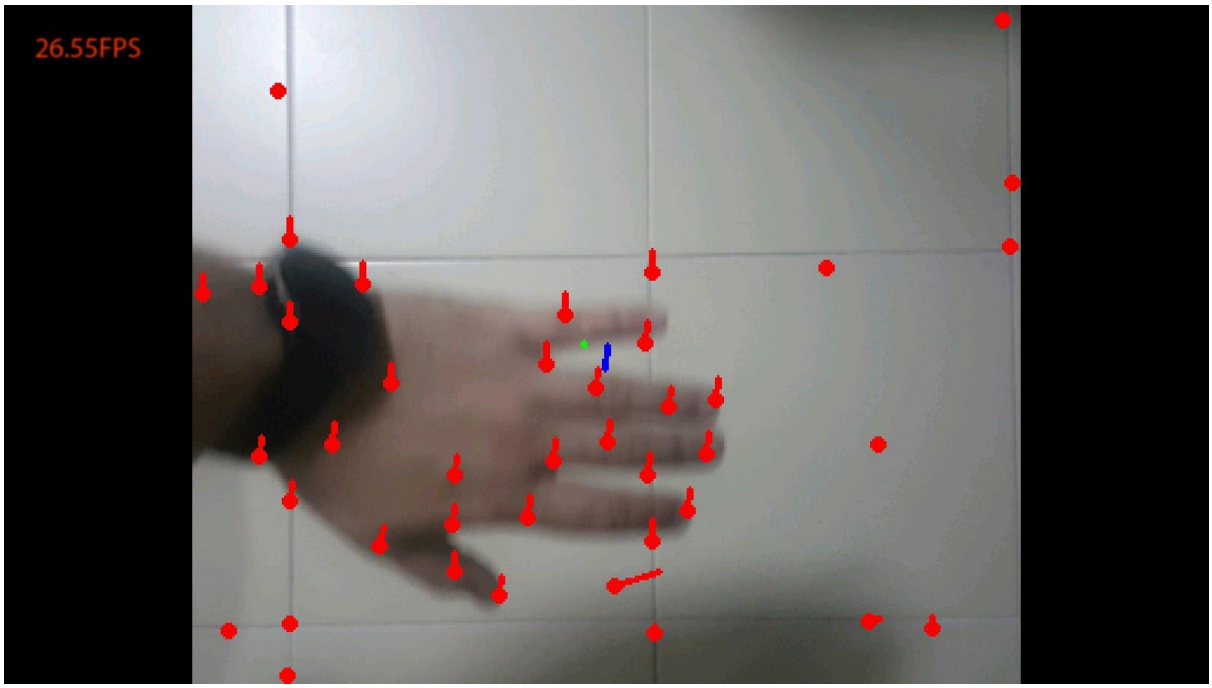

Рисунок 4.12 – Зміщення особливих точок

На рисунку 4.12 видно, що алгоритм отримує зміщення всіх особливих точок. Далі ці дані обробляються і знаходиться середнє зміщення, та середній напрямок зміщення. Також відфільтровуються та прибираються помилкові зміщення.

У процесі розробки було реалізовано в тому числі і відстеження руху у dense-потоці. Для цього було використано алгоритм Farneback'a. Результати роботи програми з даною модифікацією можна побачити на рисунку 4.13.

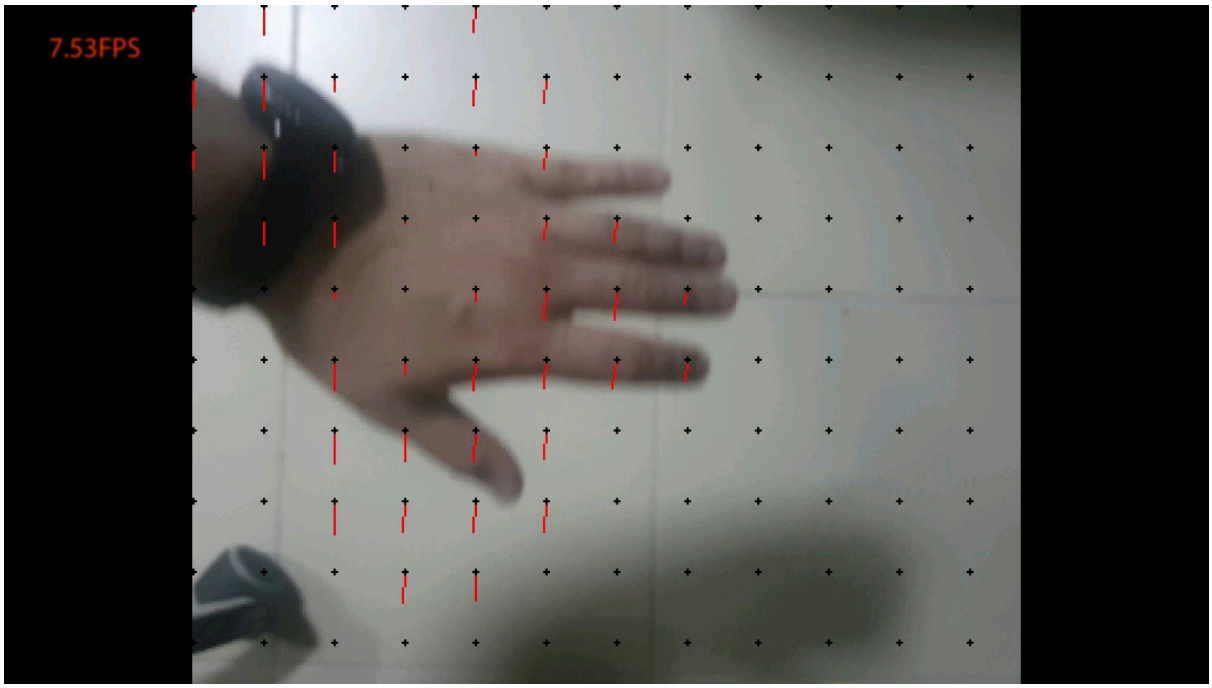

Рисунок 4.13 – Відстеження руху у dense-потоці

Як видно з рисунку 4.13, FPS занадто низький, щоб користуватися даним рішенням у реальному часі. Тому цей алгоритм не підійшов для реалізації програмного продукту.

При повороті голови також буде відбуватися відстеження зміщення особливих точок. Для запобігання цьому було розроблено модуль, який відповідає за те, щоб компенсувати значення середнього зміщення під час руху головою за допомогою даних про кутове прискорення, отриманих від гіроскопу.

4.5 Аналіз споживання енергії

Через те, що під час роботи додатку камера мобільного пристрою буде постійно активною актуальним стало питання споживання енергії програмним продуктом.

Тому було вирішено протестувати те, скільки енергії споживає смартфон за допомогою спеціального обладнання.

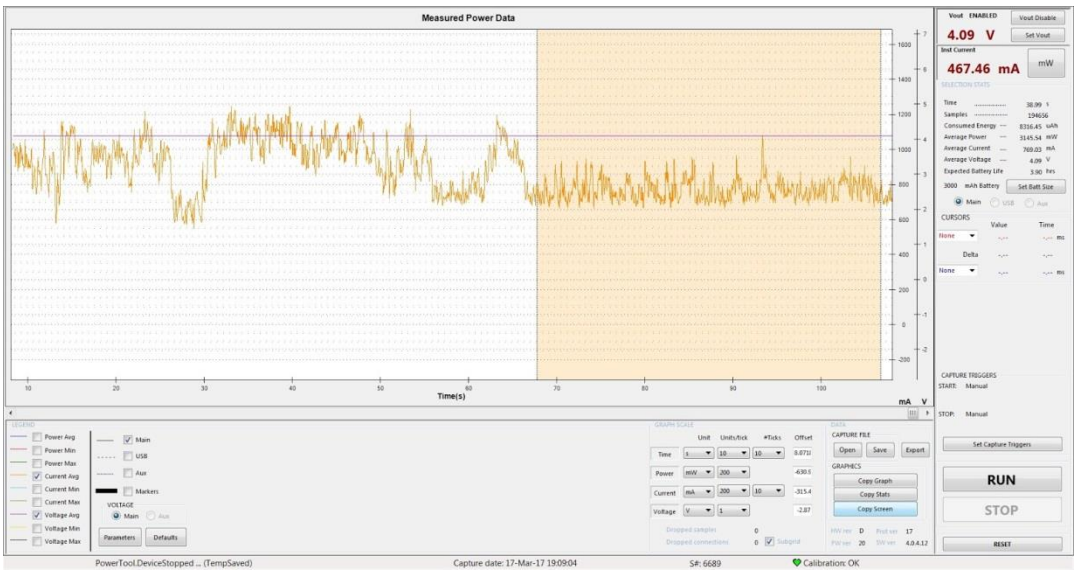

Рисунок 4.14 – Вимірювання споживання енергії без модуля комп’ютерного зору

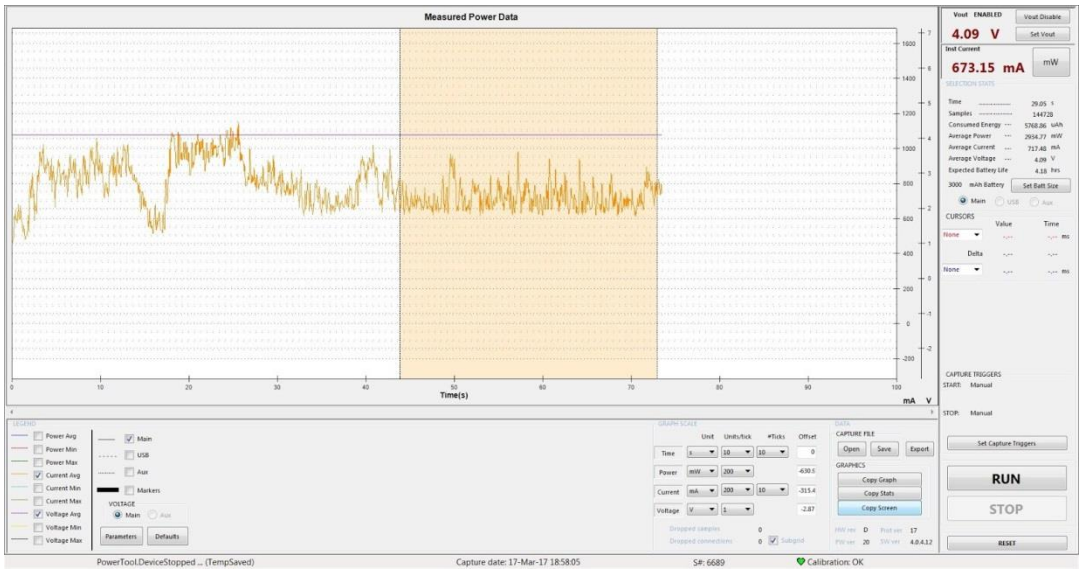

Рисунок 4.15 – Вимірювання споживання енергії з модулем комп’ютерного зору

Після проведення замірів було виявлено, що під час роботи додатку без модуля комп’ютерного зору він споживає 467.46 mA, а якщо модуль для відстеження руху пряцює, то 573.15 mA, що на становить 23% більше.

Цю величину можна зменшити, оптимізувавши частину програмного продукту, яка відповідає за роботу з камерою. Наприклад, можна відключити автофокус, тощо.

Висновки

В рамках даного розділу було розроблено програмний продукт, який дозволяє користувачу керувати контентом у віртуальній реальності за допомогою жестів. Додаток складається з двох модулів.

За допомогою фреймворку Gear Vr Framework, було створено модуль віртуальної реальності, який представляє собою віртуальне середовище відтворене у вигляді космосу навколо користувача із планетою перед ним, яку можна обертати використовуючи жести.

Також, було розроблено модуль для того, щоб відстежувати напрям і зміщення при русі на відеопотоці у реальному часі. При цьому, рух голови у гарнітурі віртуальної рельності компенсується за допомогою даних, що отримуються з гіроскопу.

При реалізації детекції руки на картинці був занадто низький FPS, а так як додаток повинен працювати у реальному часі, від цього методу довелося відмовитися. Замість цього було вирішено скористатися одним із алгоритмів визначення кутів. Після тестування було вирішено зупинитися на алгоритмі Shi-Tomasi, який показав оптимальну швидкодію та точність роботи.

Було протестовано декілька алгоритмів оцінки оптичного потоку і найкращий результат показав алгоритм Лукаса-Канаде. Основими перевагами даного алгоритму є те, що він добре працює з sparse-потоком, досить точний і швидкий.

Завдяки достатній кількісті кадрів на секунду модуль працює у режимі реального часу.

Дане програмне забезпечення після невеликого допрацювання та оптимізації можна використовувати у будь-якому проекті з підтримкою технологій віртуальної реальності.

5 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

Вступ

Стартап як форма малого ризикового (венчурного) підприємництва впродовж останнього десятиліття набула широкого розповсюдження у світі через зниження бар'єрів входу в ринок (із появою Інтернету як інструменту комунікацій та збуту стало простіше знаходити споживачів та інвесторів, займатись пошуком ресурсів, перетинати кордони між ринками різних країн), і вважається однією із наріжних складових інноваційної економіки, оскільки за рахунок мобільності, гнучкості та великої кількості стартап-проектів загальна маса інноваційних ідей зростає.

Проте створення та ринкове впровадження стартап-проектів відзначається підвищеною мірою ризику, ринково успішними стає лише невелика частка, що за різними оцінками складає від 10% до 20%. Ідея стартап-проекту, взята окремо, не вартує майже нічого: головним завданням керівника проекту на початковому етапі його існування є перетворення ідеї проекту у працюючу бізнес-модель, що починається із формування концепції товару (послуги) для визначеної клієнтської групи за наявних ринкових умов.

Розроблення та виведення стартап-проекту на ринок передбачає здійснення низки кроків, в межах яких визначають ринкові перспективи проекту, графік та принципи організації виробництва, фінансовий аналіз та аналіз ризиків і заходи з просування пропозиції для інвесторів. Узагальнено етапи розроблення стартап-проекту можна подати таким чином:

1. Маркетинговий аналіз стартап-проекту

В межах цього етапу:

- розробляється опис самої ідеї проекту та визначаються загальні напрями використання потенційного товару чи послуги, а також їх відмінність від конкурентів;
- аналізуються ринкові можливості щодо його реалізації;
- на базі аналізу ринкового середовища розробляється стратегія ринкового впровадження потенційного товару в межах проекту.

2. Організація стартап-проекту

В межах цього етапу:

- складається календарний план-графік реалізації стартап-проекту;
- розраховується потреба в основних засобах та нематеріальних активах;
- визначається плановий обсяг виробництва потенційного товару, на основі чого формулюється потреба у матеріальних ресурсах та персоналі;
- розраховуються загальні початкові витрати на запуск проекту та планові загальногосподарські витрати, необхідні для реалізації проекту.

3. Фінансово-економічний аналіз та оцінка ризиків проекту

В межах цього етапу:

- визначається обсяг інвестиційних витрат;
- розраховуються основні фінансово-економічні показники проекту (обсяг виробництва продукції, собівартість виробництва, ціна реалізації, податкове навантаження та чистий прибуток) та визначаються показники інвестиційної привабливості проекту (запас фінансової міцності, рентабельність продажів та інвестицій, період окупності проекту);
- визначається рівень ризикованості проекту, визначаються основні ризики проекту та шляхи їх запобігання (реагування на ризики).

4. Заходи з комерціалізації проекту

Цей етап спрямовано на пошук інвесторів та просування інвести-ційної пропозиції (оферти). Він передбачає:

- визначення цільової групи інвесторів та опису їх ділових інтересів;
- складання інвест-пропозиції (оферти): стислої характеристики проекту для попереднього ознайомлення інвестора із проектом;
- планування заходів з просування оферти: визначення комунікаційних каналів та площадок та планування системи заходів

з просування в межах обраних каналів;

- планування ресурсів для реалізації заходів з просування оферти.\

Означені етапи, реалізовані послідовно та вчасно – створюють передумови для успішного ринкового старту. Проте фахівці зі створення та розвитку стартаппроектів окремо відзначають, що відсутність маркетингових знань та умінь, що уможливлюють розробку ринково затребуваного проекту із вихідної ідеї, є основною причиною високого рівня банкрутств стартап-компаній, і ця проблема може бути вирішена за рахунок навчання винахідників. Відповідно, основним призначенням даних Методичних рекомендацій є надання студентам знань щодо суті, основних принципів розроблення стратегії ринкового впровадження та маркетингового управління інноваційними стартаппроектами у промислових галузях економіки, використання ефективних маркетингових інструментів просування високотехнологічних продуктів виробництва та послуг.

5.1 Опис ідеї проекту

В межах підпункту було проаналізовано і подано у вигляді таблиць:

- зміст ідеї (що пропонується);
- можливі напрямки застосування;
- основні вигоди, що може отримати користувач товару (за кожним напрямком застосування);
- чим відрізняється від існуючих аналогів та замінників;

Перші три пункти подані у вигляді таблиці (таблиця 5.1) і дають цілісне уявлення про зміст ідеї та можливі базові потенційні ринки, в межах яких потрібно шукати групи потенційних клієнтів.

Таблиця 5.1 – Опис ідеї стартап-проекту

Зміст ідеї	*Напрямки застосування*	*Вигоди для користувача*
Модуль для додатків для мобільних телефонів на платформі Android з підтримкою віртуальної реальності, який буде дозволяти користувачу керувати контентом за допомогою жестів.	1. VR-ігри на Android-смартфони	1. Зникнення необхідності купувати додаткове обладнання, таке як контроллери
	2. Навчальні та розважальні програми з підтримкою віртуальної реальності.	2. Можливість значно глибше відчути ефект занурення.
		3. Інтуїтивний метод керування, без зайвих кнопок

Аналіз потенційних техніко-економічних переваг ідеї (чим відрізняється від існуючих аналогів та замінників) порівняно із пропозиціями конкурентів передбачає:

- визначення переліку техніко-економічних властивостей та характеристик ідеї (орієнтований можливий перелік властивостей та характеристик подано у додатку А);
- визначення попереднього кола конкурентів (проектів-конкурентів) або товарів-замінників чи товарів-аналогів, що вже існують на ринку, та проведення збір інформації щодо значень техніко-економічних показників для ідеї власного проекту та проектів-конкурентів відповідно до визначеного вище переліку;
- проведення порівняльного аналізу показників: для власної ідеї визначені показники, що мають а) гірші значення (W, слабкі); б) аналогічні (N, нейтральні) значення; в) кращі значення (S, сильні) (таблиця 5.2).

Таблиця 5.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	*Техніко-економічні характеристики ідеї*	*(потенційні) товари/концепції конкурентів*		*W (слабка сторона)*	*N (нейтральна сторона)*	*S (сильна сторона)*
		Мій проект	*Контроллер*			
1.	Кросплатформеність	Можливість використання з будь-якими видами гарнітур віртуальної реальності	Деякі контроллери підходять виключно до конкретних гарнітур віртуальної реальності			+
2.	Собівартість	Низька	Висока			+
3.	Зручність використання	Руки можуть втомлюватися з плином часу	Руки не втомлюються	+		
4.	Ефект занурення	Високий	Низький			+

Визначений перелік слабких, сильних та нейтральних характеристик та властивостей ідеї потенційного товару є підґрунтям для формування його конкурентоспроможності.

5.2 Технологічний аудит ідеї проекту

В межах даного підрозділу було проведено аудит технології, за допомогою якої можна реалізувати ідею проекту (технології створення товару).

Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (таблиця 5.3):

- за якою технологією буде виготовлено товар згідно ідеї проекту?
- чи існують такі технології, чи їх потрібно розробити/доробити?
- чи доступні такі технології авторам проекту?

Таблиця 5.3 – Технологічна здійсненність ідеї проекту

№ п/п	*Ідея проекту*	*Технології її реалізації*	*Наявність технологій*	*Доступність технологій*
	VR	Daydream	Є у наявності	Доступно на найновіших телефонах на базі ОС Android.
	VR	Oculus Mobile SDK	Є у наявності	Доступно тільки на телефонах Samsung.
	Графіка	Unity	Є у наявності	Доступно на усіх телефонах на базі ОС Android.
	Графіка	Unreal Engine	Є у наявності	Доступно на усіх телефонах на базі ОС Android.
	Графіка	Gear VR Framework	Є у наявності	Доступно на усіх телефонах на базі ОС Android.
	Розпізнавання	OpenCV	Є у наявності	Безкоштовна, доступна.
Обрана технологія реалізації ідеї проекту: Oculus Mobile SDK+ Gear VR Framework + OpenCV				

За результатами аналізу таблиці 5.3 зроблено висновок, про можливість реалізації проекту. Технологічним шляхом реалізації проекту було обрано такі технології, як Oculus Mobile SDK, Gear VR Framework та OpenCV через їх доступність та безкоштовність.

5.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час

ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Спочатку було проведено аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (таблиця 5.4).

Таблиця 5.4 – Попередня характеристика потенційного ринку стартап-проекту

№ п/п	*Показники стану ринку (найменування)*	*Характеристика*
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн/ум.од	2 400 000
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі , %	R = 34%

Середню норму рентабельності в галузі було порівняно із банківським відсотком на вкладення. Останній є меншим, тому є сенс вкладати гроші саме у цей проект.

За результатами аналізу таблиці 5.4 було зроблено висновок, що ринок є привабливим для входження.

Надалі були визначені потенційні групи клієнтів, їх характеристики, та зформовано орієнтовний перелік вимог до товару для кожної групи (таблиця 5. 5).

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	*Потреба, що формує ринок*	*Цільова аудиторія (цільові сегменти ринку)*	*Відмінності у поведінці різних потенційних цільових груп клієнтів*	*Вимоги споживачів до товару*
	Потреба у можливості керування контентом у додатках з підтримкою віртуальної реальності.	Власники смартфонів на базі ОС Android, які користуються додатками з підтримкою віртуальної реальності.	Необхідно мати Android-смартфон, гарнітуру віртуальної реальності та руки.	Рішення має бути крос-платформеним та інтуїтивно-зрозумілим

Після визначення потенційних груп клієнтів було проведено аналіз ринкового середовища: складено таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (таблицяі 5.6, 5.7).

Ринкові можливості - це сприятливі обставини, які підприємство може використовувати для отримання переваг. Як приклад ринкових можливостей можна привести погіршення позицій конкурентів, різке зростання по-питу, появу нових технологій виробництва продукції, зростання рівня доходів населення і т. п. Слід зазначити, що можливостями з погляду SWOT-аналізу є не всі можливості, які існують на ринку, а тільки ті, які можна використовувати

Ринкові загрози - події, настання яких може несприятливо вплинути на підприємство. Приклади ринкових загроз: вихід на ринок нових конкурентів, зростання податків, зміна смаків покупців, зниження народжуваності й т. п.

- чинники попиту (тут доцільно взяти до уваги місткість ринку, темпи його зростання або скорочення, структуру попиту на продукцію підприємства і т. ін.);
- чинники конкуренції (слід врахувати кількість основних конкурентів, наявність на ринку товарів-замінників, висоту бар'єрів входу на ринок і

виходу з нього, розподіл ринкових часток між основними учасниками ринку і т.ін.);

- чинники збуту (необхідно надати увагу кількості посередників, наявності сіток розподілу, умовам поставок матеріалів та комплектуючих і т. ін.);
- економічні чинники (враховується курс гривні (долара, євро), рівень інфляції, зміна рівня доходів населення, податкова політика держави і т.ін.);
- політичні і правові чинники (оцінюється рівень політичної стабільності в країні, рівень правової письменності населення, рівень законослухняності, рівень корумпованості влади і т.ін.);
- науково-технічні чинники (звичайно береться до уваги рівень розвитку науки, ступінь упровадження інновацій (нових товарів, технологій) в промислове виробництво, рівень державної підтримки розвитку науки і т. ін.);
- соціально-демографічні чинники (слід врахувати чисельність і статеву та вікову структури населення регіону, в якому працює підприємство, рівень народжуваності і смертності, рівень зайнятості населення і т.ін.);
- соціально-культурні чинники (звичайно враховуються традиції і система цінностей суспільства, існуюча культура споживання товарів і послуг, наявні стереотипи поведінки людей і т. ін.);
- природні і екологічні чинники (враховується кліматична зона, в якій працює підприємство, стан навколишнього середовища, відношення громадськості до захисту навколишнього середовища і т.ін.);
- міжнародні чинники (серед них враховується рівень стабільності в світі, наявність локальних конфліктів і т. ін.).

Таблиця 5.6 – Фактори загроз

№ п/п	*Фактор*	*Зміст загрози*	*Можлива реакція компанії*
	Конкуренція	Поява уніфікованого, дешевого та зручного пристрою для керування контентом у віртуальній реальності	1. Передбачити додаткові переваги власного проекту для того, щоб повідомити про них саме після виходу міжнародної компанії на ринок. 2. Обрати нову цільову аудиторію і зосередитися на ній
	Економічний	Подорожчання топових смартфонів.	Оптимізація програмного продукту, для можливості його запуску на більш бюджетних пристроях.

Таблиця 5.7 – Фактори можливостей

№ п/п	*Фактор*	*Зміст можливості*	*Можлива реакція компанії*
	Науково-технічний	Тенденція до випуску гарнітур віртуальної рельності з вбудованими камерами глибини, які дозволяють краще отримувати інформацію про навколишній простір та краще відстежувати положення рук.	Адаптація існуючого рішення і алгоритмів під нову технологію.
	Попит	Більш широке розповсюдження VR технологій.	Постійна підтримка продукту.

Надалі було проведено аналіз пропозиції: визначили загальні риси конкуренції на ринку (таблиця 5.8).

Таблиця 5.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	*В чому проявляється дана характеристика*	*Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)*
1. Вказати тип конкуренції: монополістична конкуренція.	Існує декілька фірм-конкурентів.	Підтримка якості продукту та постійні нововведення.
2. За рівнем конкурентної боротьби: Міжнародний.	Фірми-конкуренти знаходяться в інших країнах.	Адаптація продукту як для вітчизняних так і для зарубіжних клієнтів.
3. За галузевою ознакою: внутрішньогалузева.	Продукт використовується лише всередині даної галузі.	Постійне вдосконалення продукту.
4. Конкуренція за видами товарів: товарно-видова.	Види товарів однакові.	Створити продукт, враховуючи сильні і слабкі сторони конкурентів.
5. За характером конкурентних переваг: неціновa.	Вдосконалення технології створення контроллерів.	Зниження ціни на продукт та підтримка його якості.
6. За інтенсивністю: марочна.	Бренди існують і конкурують.	PR, реклама, просування бренду.

Було проведено аналіз конкуренції у галузі за моделлю М. Портера (таблиця 5.9). М. Портер вирізняє п’ять основних факторів, що впливають на привабливість вибору ринку з огляду на характер конкуренції. Це:

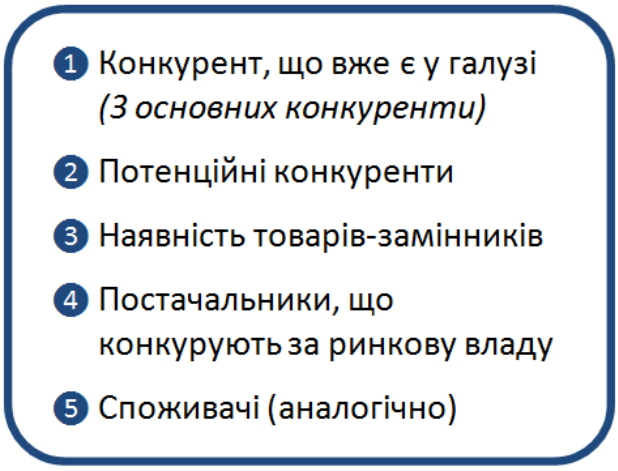

Рисунок 5.1 – Модель Портера

Сильні позиції компанії за кожним з факторів означають її можливості забезпечити необхідні темпи обороту капіталу та її здатність впливати на інших агентів ринку, диктуючі їм власні умови співпраці. Характеристики факторів моделі відрізняються для різних галузей та змінюються із часом. Сила кожного фактору є функцією від структури галузі та її техніко-економічних характеристик.

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	*Потенційні конкуренти*	*Постачальники*	*Клієнти*	*Товари-замінники*
Складові аналізу	Oculus, Daydream, Meizu	Наявність вже існуючих рішеннь	-	Контроль якості продукту	Поява уніфікованого та більш зручного методу вводу
Висновки:	Доволі інтенсивна конкурентна боротьба з вже закріпившимися на ринку гравцями.	Є можливості виходу на ринок, але є і конкуренти. Строки – 8 місяців.	-	Клієнти диктують усі умови роботи на ринку.	Перехід усіх додатків для віртуальної реальності на уніфікований метод управління.

За результатами аналізу таблиці 5.9 було зроблено висновок про

можливість роботи на ринку з огляду на конкурентну ситуацію. Також було зроблено висновок щодо характеристик, які повинен мати проект, щоб бути конкурентноспроможним на ринку. Цей висновок був врахований при формулюванні переліку факторів конкурентноспроможності у наступному пункті.

На основі аналізу конкуренції, проведеного в таблиці 5.9, а також із урахуванням характеристик ідеї проекту (таблиця 5.2), вимог споживачів до товару (таблиця 5.5) та факторів маркетингового середовища (таблиці 5.6, 5.7) визначається та обґрунтовується перелік факторів конкурентоспроможності. Аналіз оформлюється за таблицею 5.10

Таблиця 5.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	*Фактор конкурентоспроможності*	*Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)*
	Виконання програмного забезпечення у кросплатформеному вигляді	Можливість використання програмного забезпечення на будь-якій платформі.
	Ціна	Дане рішення не потребує використання додаткового обладнання та матеріалів, а достатньо лише програмного рішення.

За визначеними факторами конкурентоспроможності (таблиця 5.10) проведено аналіз сильних та слабких сторін стартап-проекту (таблиця 5.11).

Таблиця 5.11 – Порівняльний аналіз сильних та слабких сторін

№ п/п	*Фактор конкурентоспроможності*	*Бали 1-20*	*Рейтинг товарів-конкурентів у порівнянні*						
			–3	*–2*	*–1*	*0*	*+1*	*+2*	*+3*
1	Виконання програмного забезпечення у кросплатформеному вигляді	17			+				
2	Ціна	20		+					

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких

(Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (таблиця 5.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (таблиця 5.11).

Перелік ринкових загроз та ринкових можливостей було складено на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення. Наприклад: зниження доходів потенційних споживачів – фактор загрози, на основі якого можна зробити прогноз щодо посилення значущості цінового фактору при виборі товару та відповідно, – цінової конкуренції (а це вже – ринкова загроза).

Таблиця 5.12 – SWOT- аналіз стартап-проекту

Сильні сторони: ціна, кросплатформеність	Слабкі сторони: у деяких випадках можуть стомлюватися руки
Можливості: більш широке розповсюдження технологій з підтримкою віртуальної реальності, поява нових технологій моніторингу навколишнього середовища.	Загрози: видавлення з ринку конкурентами, зміна потреб користувачів

На основі SWOT-аналізу було розроблено альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок (див. таблицю 5.9, аналіз потенційних конкурентів).

Визначені альтернативи були проаналізовані з точки зору строків та ймовірності отримання ресурсів (таблиця 5.13).

Таблиця 5.13 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	*Альтернатива (орієнтовний комплекс заходів) ринкової поведінки*	*Ймовірність отримання ресурсів*	*Строки реалізації*
1	Розробка програмного продукту, PR, просування бренду	90%	9 місяців
2	Розробка програмного продукту, безкоштовне розповсюдження	60%	6 місяців

Після аналізу було обрано альтернативу №1.

5.4 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: було проведено опис цільових груп потенційних споживачів (таблиця 5.14).

Таблиця 5.14 – Вибір цільових груп потенційних споживачів

№ п/п	*Опис профілю цільової групи потенційних клієнтів*	*Готовність споживачів сприйняти продукт*	*Орієнтовний попит в межах цільової групи (сегменту)*	*Інтенсивність конкуренції в сегменті*	*Простота входу у сегмент*
1	Користувачі смартфонів з ОС Android віком 10 – 40	Висока	Високий	Висока	Середня
2	Користувачі смартфонів з ОС Android віком 40 – 99	Невисока	Невисокий	Невисока	Середня
Які цільові групи обрано: 1					

За результатами аналізу потенційних груп споживачів було обрано цільову групу, для якої буде запропоновано даний товар, та визначено стратегію охоплення ринку - стратегію концентрованого маркетингу(компанія зосереджується на одному сегменті).

Для роботи в обраних сегментах ринку сформовано базову стратегію розвитку (таблиця 5.15).

За М. Портером, існують три базові стратегії розвитку, що відрізняються за ступенем охоплення цільового ринку та типом конкурентної переваги, що має бути реалізована на ринку (за витратами або визначними якостями товару).

Стратегія лідерства по витратах передбачає, що компанія за рахунок

чинників внутрішнього і/або зовнішнього середовища може забезпечити більшу, ніж у конкурентів маржу між собівартістю товару і середньоринковою ціною (або ж ціною головного конкурента). Зокрема, ця стратегія припускає, що за рахунок великих можливостей по об'ємах збуту товарів (портфеля укладених контрактів на постачання) і продуктивності підприємство може добитися менших витрат. Ця стратегія зазвичай тісно пов'язана з можливістю досягнення ефекту масштабу і досвіду.

Компанії, що вибирають цю стратегію, проводять ретельний контроль за постійними витратами, знижують виробничі, збутові і рекламні витрати, проводять інвестиції, спрямовані на зменшення витрат, ретельне опрацювання конструкції нових товарів.

Переваги стратегії за Ж.-Ж. Ламбеном:

- фірма здатна протистояти своїм прямим конкурентам навіть у разі цінової війни і в змозі отримувати прибуток при ціні, мінімально допустимій для конкурентів;
- сильні клієнти не можуть добитися зниження ціни нижче рівня, прийнятного для найбільш сильного конкурента;
- низькі витрати забезпечують захист проти сильних постачальників, оскільки дають фірмі велику гнучкість у разі підвищення вхідних витрат;
- низькі витрати створюють бар'єр входу для нових конкурентів і одночасно хороший захист проти товарів-замінників.

В ході конкурентної боротьби з використанням цієї стратегії з ринку вимушені будуть піти фірми, менш ефективні з точки зору величини і структури витрат, нездібні до проведення технологічних новацій, спрямованих на зниження витрат.

Стратегія диференціації передбачає надання товару важливих з точки зору споживача відмітних властивостей, які роблять товар відмінним від товарів конкурентів. Така відмінність може базуватися на об'єктивних або суб'єктивних, відчутних і невідчутних властивостях товару(у ширшому

розумінні – комплексі маркетингу), бути реальною або уявною. Інструментом реалізації стратегії диференціації є ринкове позиціонування.

Переваги стратегії за Ж.-Ж. Ламбеном:

- по відношенню до прямих конкурентів диференціація знижує ступінь замінності товару, посилює прихильність марці, зменшує чутливість до ціни і тим самим підвищує рентабельність;
- прихильність клієнтів послабляє їх тиск на фірму і перешкоджає приходу на ринок нових конкурентів;
- підвищена рентабельність збільшує стійкість до можливого зростання витрат в результаті дій сильного постачальника;
- відмітні властивості товару і завойована прихильність клієнтів захищають фірму і від товарів-замінників.

Реалізація цієї стратегії вимагає, як правило, більш високих витрат. Проте успішна диференціація дозволяє компанії домогтись більшої рентабельності за рахунок того, що ринок готовий прийняти більш високу ціну (цінову премію бренду).

При веденні конкурентної боротьби з використанням цієї стратегії на ринку в першу чергу терплять фіаско фірми, що не здатні визначати потреби цільових ринків, оперативно реагувати на зміни в ринковому попиті, проводити ефективну політику маркетингових комунікацій, не мають необхідних навичок в області брендингу. Найважливішими здібностями, які повинна мати компанія, що приймає цю стратегію, є з генерування маркетингових ноу-хау, здійснення продуктових новацій.

Стратегія спеціалізації передбачає концентрацію на потребах одного цільового сегменту, без прагнення охопити увесь ринок. Мета тут полягає в задоволенні потреб вибраного цільового сегменту краще, ніж конкуренти. Така стратегія може спиратися як на диференціацію, так і на лідерство по витратах, або і на те, і на інше, але тільки у рамках цільового сегменту. Проте низька ринкова доля у разі невдалої реалізації стратегії може істотно підірвати конкурентоспроможність компанії.

Таблиця 5.15 – Визначення базової стратегії розвитку

№ п/п	*Обрана альтернатива розвитку проекту*	*Стратегія охоплення ринку*	*Ключові конкурентоспроможні позиції відповідно до обраної альтернативи*	*Базова стратегія розвитку*
	Розробка програмного продукту, PR, просування бренду	Масовий маркетинг	Екстрановий спосіб рухової взаємодії, який не потребує громіздких додаткових девайсів(контроллерів, які потрібно регулярно заряджати)	Стратегія диференціації

Наступним кроком обрано стратегію конкурентної поведінки (таблиця 5.16).

Стратегія лідера

Залежно від міри сформованості товарного(галузевого) ринку, характеру конкурентної боротьби компанії-лідери обирають одну з трьох стратегій: розширення первинного попиту, оборонну або наступальну стратегію або ж застосувати демаркетинг або диверсифікацію.

Стратегія розширення первинного попиту доцільна у разі, якщо фірмі-лідерові недоцільно розмінюватися на боротьбу з невеликими конкурентами, вона може отримати велику економічну віддачу від розширення первинного рівня попиту. В цьому випадку компанія займається реалізацією заходів по формуванню попиту(навчанню споживачів користуванню товаром, формування регулярного попиту, збільшення разового споживання), також пропаганду нових напрямів застосувань існуючих товарів, виявлень нових груш споживачів. Розширюючи таким чином ринковий попит, лідер надає допомогу усім підприємствам, що «йдуть за ним», несучи при цьому основні фінансові витрати, проводячи найбільш революційні НДДКР. Така стратегія можлива тільки на початкових стадіях життєвого циклу товару, коли попит ще є

розширюваним, а взаємний тиск конкурентів ще невеликий. Інакше фірмі лідерові необхідно приймати оборонну або наступальну стратегію.

У міру зростання ринку, його становлення позиції компанії-новатора починають атакувати конкуренти-імітатори. В цьому випадку, компанія може вибрати оборонну стратегію, метою якої є захист власної ринкової долі. Оборона може бути:

- інновації з метою постановки технологічних бар'єрів для входу в ринок нових конкурентів, подальшого збільшення відриву від них;
- ліквідація ніш для проникнення конкурентів за допомогою розширення товарного асортименту, цінових парасольок, захоплення каналів збуту;
- ведення цінової війни і/або проведення масованої рекламної атаки.

Наступальна стратегія припускає збільшення своєї частки ринку. При цьому переслідувана мета полягає в подальшому підвищенні прибутковості роботи компанії на ринку за рахунок максимального використання ефекту масштабу. Проте, існує межа, при перевищенні якої подальше зростання частки ринку стає невигідним. Це або чисто економічна недоцільність відвойовування добре захищених часток, що сильно захищаються, у дрібніших виробників або ж попадання під дію антимонопольного законодавства.

Наступальна стратегія припускає активну інноваційну політику компанії. Вона постійно атакує власні ж досягнення, збільшуючи розрив між собою і основними конкурентами. Постійні техніко-економічні вдосконалення, модифікація розміру і форми упаковки, використання event- маркетингу – типові складові арсеналу фірм-лідерів.

Якщо фірма потрапляє під дію антимонопольного законодавства, вона може удатися до стратегії демаркетинга, що припускає скорочення своєї частки ринку, зниження рівня попиту на деяких сегментах за рахунок підвищення ціни. При цьому ставиться завдання недопущення на ці сегменти конкурентів, а компенсація втрат прибутку через зменшення обсягів виробництва компенсується встановленням надвисоких цін.

Проте у більшості випадків найпривабливішою стратегією для компаній-лідерів є диверсифікація, що дозволяє використати переваги масштабу виробництва, know – how.

Стратегія виклику лідера

Стратегію виклику лідерові найчастіше вибирають компанії, які є другими, третіми на ринку, але бажають стати лідером ринку. Теоретично, ці компанії можуть прийняти два стратегічні рішення: атакувати лідера у боротьбі за частку ринку або ж йти за лідером.

Рішення атакувати лідера є досить ризикованим. Для його реалізації потрібні значні фінансові витрати, know – how, краще співвідношення «ціна-якість», переваги в системі розподілу і просування і т. д. У разі не реалізації цієї стратегії, компанія може бути відкинута на аутсайдерські позиції на досить довгий час. Тому реалізація цієї стратегії вимагає детального опрацювання по наступних напрямах:

- аналіз сильних і слабких сил своїх і фірми-лідера;
- виявлення можливих напрямів атаки;
- ревізія власних сил і ресурсів;
- аналіз можливих дій конкурентів і розробка методів захисту.

Залежно від цього компанія може вибрати одну з альтернативних стратегій: фронтальної або флангової атаки.

Фронтальна атака припускає атаку на сильні сторони конкурента. Така стратегія вимагає наявності у фірми значної переваги над тим, що атакує. У військовій стратегії це співвідношення зазвичай складає 3: 1. При цьому складно не лише стати першим, а мати можливості утримати першість в подальшому. Нового лідера атакуватиме не лише програвша фірма, але і треті, четверті в надії переділити ринок. В силу цього ця стратегія є найбільш ризиковою і у разі невдачі відбувається « виснаження» компанії, що може привести до значного відкидання підприємства. У разі ж успіху компанія стає лідером ринку з усіма перевагами цієї позиції.

Флангова атака передбачає атаку на слабкі сторони фірми-лідера,

наприклад, неосвоєні або погано відстежувані регіональні ринки або ринкові сегменти, ціну, значущий для споживача сервіс або показники якості продукції. Особливо б’є по лідерові цінова атака, оскільки, маючи велику ринкову частку, при зниженні ціни в абсолютному вираженні лідер терпить великі втрати, а недостатній приплив фінансових ресурсів відразу ж оголяє раніше приховані латентні слабкі місця компанії, може привести до системної кризи.

Стратегія наслідування лідеру

Компанії, що приймають слідування за лідером – це підприємства з невеликою часткою ринку, які вибирають адаптивну лінію поведінки на ринку, усвідомлюють своє місце на нім і йдуть у фарватері фірм-лідерів. Головна перевага такої стратегії – економія фінансових ресурсів, пов’язаних з необхідністю розширення товарного(галузевого) ринку, постійними інноваціями, витратами на утримання домінуючого положення.

Стратегія наслідування лідеру найчастіше має місце у випадку олігополії, коли кожен конкурент прагне уникнути боротьби, особливо цінової, а також у випадку, коли слабо виражений ефект масштабу, що не дозволяє отримати переваги від об’ємів продажів або ж він не грає істотної ролі. Стратегію наслідування лідеру приймають також фірми, які не змогли реалізувати стратегію виклику лідерові.

Компанії, що приймають таку стратегію, зазвичай випускають товари-імітатори, займаючи ринкову частку, яку з різних причин не можуть охопити фірми лідери. Вибір такої стратегії може також бути обумовлений також перевагою локалізації (краще знання ринку, налагоджені зв’язки з клієнтами тощо).

Для ефективної реалізації цієї стратегії компанії повинні задовольняти наступним основним умовам:

- систематичний аналіз сегментації ринку з метою виділення нових ринкових сегментів або таких, що незадовільно обслуговуються;
- ефективне використання НДДКР з метою вдосконалення технологічних процесів і незначних продуктових новацій;

- концентрація на прибутковості, а не на простому зростанні об’ємів продажів;
- постійний аналіз витрат на всіх стадіях виробництва і логістики;
- залишатися досить малим, щоб не бути досить цікавим для фірм-лідерів;
- сильний керівник, здатний не лише формулювати стратегію, але і тримати усю діяльність компанії під власним контролем.

Якщо врахувати, що лідерами ринку можуть бути лише декілька компаній, то ця стратегія є наймасовішою.

Стратегія заняття конкурентної ніші.

При прийнятті стратегії зайняття конкурентної ніші (інші назви – стратегія фахівця або нішера) компанія в якості цільового ринку вибирає один або декілька ринкових сегментів. Головна особливість – малий розмір сегментів/сегменту. Ця конкурентна стратегія являється похідною від такої базової стратегії компанії, як концентрація.

Ніша, для того, щоб вона була привабливою для компанії, повинна задовольняти таким умовам:

- бути досить прибутковою, щоб робити доцільним процес виробництва і обслуговування;
- залишатися стабільною упродовж тривалого проміжку часу;
- має бути добре захищеною, мати високі вхідні бар’єри;
- бути непривабливою для конкурентів;
- відповідати цілям і ресурсам компанії, її специфічним можливостям.

Головне завдання для компаній, що вибирають стратегію нішера або фахівця, – це постійна турбота про підтримку і розвиток своєї конкурентної переваги, формування лояльності і прихильності споживачів, підтримка вхідних бар’єрів.

Таблиця 5.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	*Чи є проект «першопрохідцем» на ринку?*	*Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?*	*Чи буде компанія копіювати основні характеристики товару конкурента, і які?*	*Стратегія конкурентної поведінки*
	Ні	Забирати існуючих	Ні	Стратегія наслідування лідеру

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту (див. таблицю 5.5), а також в залежності від обраної базової стратегії розвитку (таблиця 5.15) та стратегії конкурентної поведінки (таблиця 5.16) розроблено стратегію позиціонування (таблиця 5.17), що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Позиціювання — це маркетингове забезпечення товарові бажаного місця на ринку і у свідомості потенційних покупців (образ). Позиція компанії чи продукту показує чим він унікальний УТП (унікальну торговельну пропозицію), чим відрізняється від конкурентів (відстройка від конкурентів), чим корисний споживачу.

З точки зору маркетингу товар являє собою сукупність відчутних (розмір, колір, маса, швидкість і т. п.) і невловимих (престижність, модність, сучасність і т. п.) властивостей.

Дослідження свідчать, якщо позиціювання здійснюється більше ніж за трьома ознаками, то воно є неефективним, оскільки не відкладається у свідомості споживача.

Таблиця 5.17 – Визначення стратегії позиціонування

№ п/п	*Вимоги до товару цільової аудиторії*	*Базова стратегія розвитку*	*Ключові конкурентоспроможні позиції власного стартап-проекту*	*Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)*
1	Невисока ціна	позиціювання за показниками ціни	Відсутність контроллерів	Економічність, екологічність, ергономічність

Результатом виконання підрозділу стала узгоджена система рішень щодо ринкової поведінки стартап-компанії, яка визначає напрями роботи стартап-компанії на ринку.

5.5 Розроблення маркетингової програми стартап-проекту

Сформовано маркетингову концепцію товару, який отримає споживач. Для цього у таблиці 5.18 підсумовано результати попереднього аналізу конкурентоспроможності товару.

Концепція товару - письмовий опис фізичних та інших характеристик товару, які сприймаються споживачем, і набору вигод, які він обіцяє певній групі споживачів.

Таблиця 5.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	*Потреба*	*Вигода, яку пропонує товар*	*Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити*
	Кросплатформеність	Можливість використання на будь-якій платформі з підтримкою віртуальної реальності	Рішення є кроплатформеним
	Ціна	Низька ціна	Користувачу не потрібно платити зайві гроші.

Розроблено трирівневу маркетингову модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (таблиця 5.19).

1-й рівень

При формуванні задуму товару вирішується питання щодо того, засобом вирішення якої потреби і / або проблеми буде даний товар, яка його основна вигода. Дане питання безпосередньо пов'язаний з формуванням технічного завдання в процесі розробки конструкторської документації на виріб.

2-й рівень

Цей рівень являє рішення того, як буде реалізований товар в реальному/ включає в себе якість, властивості, дизайн, упаковку, ціну.

3-й рівень

Товар з підкріпленням (супроводом) - додаткові послуги та переваги для споживача, що створюються на основі товару за задумом і товару в реальному виконанні (гарантії якості , доставка, умови оплати та ін).

Таблиця 5.19 – Опис трьох рівнів моделі товару

<table>
<tr><th>Рівні товару</th><th colspan="3">Сутність та складові</th></tr>
<tr><td>I. Товар за задумом</td><td colspan="3">Товар дозволяє керувати контентом у додатках з підтримкою технологій віртуальної реальності.</td></tr>
<tr><td rowspan="5">II. Товар у реальному виконанні</td><td>Властивості/характеристики</td><td>М/Нм</td><td>Вр/Тх /Тл/Е/Ор</td></tr>
<tr><td>1. Якість
2. Простота у використанні
3. Низька ціна</td><td>-</td><td>-</td></tr>
<tr><td colspan="3">Якість: згідно до стандарту ISO 4444 буде проведено тестування</td></tr>
<tr><td colspan="3">Пакування нема</td></tr>
<tr><td colspan="3">Марка (власна): NK Studios + VR Gesture Control</td></tr>
<tr><td rowspan="2">III. Товар із підкріпленням</td><td colspan="3">До продажу: 1-місячна пробна безкоштовна версія</td></tr>
<tr><td colspan="3">Після продажу: Постійне додання нових жестів</td></tr>
<tr><td colspan="4">За рахунок чого потенційний товар буде захищено від копіювання: ноу-хау</td></tr>
</table>

Після формування маркетингової моделі товару слід відмітити, що проект буде захищено від копіювання за допомогою ноу-хау.

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (остаточне визначення ціни відбувається під час фінансово-економічного аналізу проекту), яке передбачає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів цільової групи споживачів (таблиця 5.20). Аналіз проведено експертним методом.

Таблиця 5.20 – Визначення меж встановлення ціни

№ п/п	*Рівень цін на товари-замінники*	*Рівень цін на товари-аналоги*	*Рівень доходів цільової групи споживачів*	*Верхня та нижня межі встановлення ціни на товар/послугу*
	12-40$	12-40$	3000$	1-8$

Наступним кроком є визначення оптимальної системи збуту, в межах якого було прийняте рішення (таблиця 5.21):

- проводити збут власними силами і залучати сторонніх посередників.
- користуватися однорівневим каналом збуту;

Таблиця 5.21 – Формування системи збуту

№ п/п	*Специфіка закупівельної поведінки цільових клієнтів*	*Функції збуту, які має виконувати постачальник товару*	*Глибина каналу збуту*	*Оптимальна система збуту*
	Одна одиниця на особу	Роздрібна торгівля	Однорівневий	Власні сили та через посередників

Останньою складової маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (таблиця 5.22).

Таблиця 5.22 – Концепція маркетингових комунікацій

№ п/п	*Специфіка поведінки цільових клієнтів*	*Канали комунікацій, якими користуютьс я цільові клієнти*	*Ключові позиції, обрані для позиціонуван ня*	*Завдання рекламного повідомлення*	*Концепція рекламного звернення*
	Клієнти обиратимуть зручніший товар з потрібними функціями.	Соціальні мережі, електронна пошта, мобільні телефони	Ціна, простота використання , кросплатфор меність, більш істотний ефект занурення	Показати переваги продукту, низьку ціну, занурення у віртуальне середовище за допомогою продукту.	Демо ролик з використання м, реклама.

Результатом підрозділу стала ринкова (маркетингова) програма, що включає в себе концепції товару, збуту, просування та попередній аналіз можливостей ціноутворення, спирається на цінності та потреби потенційних клієнтів, конкурентні переваги ідеї, стан та динаміку ринкового середовища, в межах якого впроваджено проект, та відповідну обрану альтернативу ринкової поведінки.

Висновки

В даному розділі було проведено аналіз програмного продукту у якості стартап проекту. Можна зазначити що у проекта є можливість комерціалізації, адже ринок технологій віртуальної реальності динамічно розвивається, створюються нові додатки які, в свою чергу, стимулють попит на різноманітні контролери для управління контентом у віртуальній реальності.

На ринку наявна монополістична конкуренція, існує декілька фірм-конкурентів, тому вихід на нього не буде легким. Проте проект є доволі конкурентноспроможним завдяки, в першу чергу, своїй низькій собівартості. Через те, що він є повністю програмним, його розробка не потребує витрат на різноманітні матеріали та обладнання, необхідні для виготовлення корпусу,

схем, тощо.

Для впровадження ринкової реалізації проекту слід обрати альтернативу, яка передбачає розробку програмного продукту, а потім якісну рекламу та PR, сконцентровану навколо позитивних характеристиках даного програмного продукту, таких як низька ціна, більш істотний ефект занурення, кросплатформеність, екологічність і т.д.

З огляду на проведений аналіз, можна чітко сказати, що подальша імплементація проекту є доцільною, адже він може знайти свою цільову аудиторію та зайняти місце на ринку.

ВИСНОВКИ

Комп’ютерний зір та віртуальна реальність має великий спектр застосувань у багатьох галузях. Широко застосовується у навчанні, науці, архітектурі, медицині, тощо.

Під час написання даної дипломної роботи було розглянуто два досить великих напрямки: розробка мобільних додатків під віртуальну реальність та трекінг руху у відеопотоці.

Аналіз сучасних платформ, що підтримують розробку додатків з віртуальною реальністю для операційної системи Android показав, що платформи Daydream та Oculus Mobile SDK мають приблизно однаковий функціонал. Але через те, що платформа Daydream доступна лише на найновіших мобильних пристроях, а Oculus Mobile SDK э в наявності на телефонах Samsung починаючи з Galaxy S6, для розробки додатку було обрано саме її.

Дослідження фреймворків, які дозволяють полегшити розробку програм з підтримкою технологій віртуальної реальності показало, що доцільним є використання Unreal Engine, якщо продуктивність і якість картинки критичні для ПЗ. Також варто використовувати Unreal Engine, якщо буде існувати необхідність модифікації вихідного коду фреймворку. Якщо ж у проекту немає істотних вимог до якості візуальної частини та більш важливим є легкість розробки і розгортання, то слід подумати про використання Unity.

У випадку, коли не потрібна надмірна перевантаженість функціями та інтерфейсом, або не важливим є наявність редактора сцени слід обрати Gear Vr Framework. Розробка програмного продукту за допомогою цього фреймворка ведеться у Android Studio, що дозволяє самостійно додавати сторонні бібліотеки та модулі при необхідності, не потребуючи зайвих плагінів.

Аналіз бібліотеки комп’ютерного зору OpenCV та існуючих алгоритмів комп’ютерного зору для трекінгу руху об’єкта у відеопотоці показав, що якщо достатньо порахувати зсув деякого набору точок (sparse потік), то алгоритм Лукаса-Канаде працює добре, точно і досить швидко. Для обчислення dense-

потоку добре використовувати алгоритм SimpleFlow, але він працює дуже повільно. Якщо швидкодія є критичною, то для обчислення dense-потоку слід використовувати алгоритм Farneback'а, але необхідно упевнитися, що точність його роботи є достатньою.

Використовуючи засоби та технології обрані за результатами аналізу було розроблено програмний продукт який дозволяє користувачу керувати об'єктом віртуального середовища за допомогою рухів.

Модуль віртуальної реальності у розробленому додатку, який представляє собою віртуальне середовище відтворене у вигляді космосу навколо користувача із планетою перед ним, яку можна обертати використовуючи рухи, було створено за допомогою фреймворку Gear Vr Framework, адже він дозволяє вести розробку також і сторонніх модулів для Android-проекту, що потрібно для реалізації частини, яка використовує OpenCV та відповідає за детекцію жестів, в той час як Unity та Unreal Engine не мають безкоштовних плагінів для роботи з OpenCV.

Крім цього, для реалізації модуля комп'ютерного зору було протестовано декілька алгоритмів та підходів і по результатам тестування обрано найкращі. Для детектування особливих точок застосовується алгоритм Shi-Tomasi, а для оцінки оптичного потоку найкраще підійшов алгоритм Лукаса-Канаде. При реалізації інших підходів занадто сильно знижувалася кількість кадрів на секунду, що унеможливлювало роботу додатку у режимі реального часу.

Після реалізації додатку були виявлені деякі місця для його покращення. Наприклад, для зменшення споживання енергії було б добре оптимізувати частину, яка працює з камерою.

Результатами цієї роботи є рекомендації, щодо застосування методів комп'ютерного зору для відстеження руху у відеопотоці на мобільних платформах, інструментів та платформ для розробки додатків з підтримкою технологій віртуальної реальності на мобільних платформах, а також розроблений мобільний додаток, який дав змогу протестувати методи у

реальних умовах.

Віртуальна реальності розвивається дуже стрімко і її синтез з технологіями комп’ютерного зору – логічний етап цього розвитку. Вже створюються нові технології, які дозволятимуть сканувати навколишнє середовище за допомогою вбудованих камер. А отже, алгоритми комп’ютерного зору будуть необхідні для обробки такої інформації.

ПЕРЕЛІК ПОСИЛАННЬ

1. Jason Jerald, Peter Giokaris, Danny Woodall, Arno Hartbolt, Anish Chandak, Sebastien Kuntz Developing virtual reality applications with Unity, Virtual Reality (VR), 2014 iEEE, Minneapolis, MN, USA.
2. Jonathan Schlueter, Holly Baiotto, Melynda Hoover, Vijay Kalivarapu. Gabriel Evans, Eliot Winer Best practices for cross-platform virtual reality development, SPIE 10197, Degraded Environments: Sensing, Processing, and Display 2017, 1019709 (May 5, 2017).
3. David TrenholmeShamus P. Smith Computer game engines for developing first-person virtual environments, S.P. Virtual Reality (2008) 12: 181.
4. Parth Rajesh Desai, Pooja Nikhil Desai, Komal Deepak Ajmera, Khushbu Mehta A Review Paper on Oculus Rift-A Virtual Reality Headset, https://arxiv.org/abs/1408.1173
5. Marie-Laure Ryan, Narrative as Virtual Reality: Immersion and Interactivity in Literature and Electronic Media, Johns Hopkins University Press, Baltimore, MD, 2001
6. Анна Юрьевна Скворцова, Олег Олегович Варламов, Георгий Сергеевич Сергушин, Ангелина Игоревна Белоусова Исследование возможностей практического применения технологий виртуальной реальности и угрозы ее развития, Автоматизация и контроль в технических системах, 2012
7. Seth, A., Vance, J.M. & Oliver, J.H., Virtual reality for assembly methods prototyping: a review, 2011, doi:10.1007/s10055-009-0153-y
8. Doug A. Bowman, Ryan P. McMahan Virtual Reality: How Much Immersion Is Enough?, Computer (Volume: 40, Issue: 7, July 2007)
9. F.P. Brooks What's real about virtual reality?, IEEE Computer Graphics and Applications (Volume: 19, Issue: 6, Nov-Dec 1999)
10. M. Mauve, J. Vogel, V. Hilt, W. Effelsberg Local-lag and timewarp: providing consistency for replicated continuous applications, IEEE Transactions on Multimedia (Volume: 6, Issue: 1, Feb. 2004)

11. S.R. Ellis What are virtual environments?, IEEE Computer Graphics and Applications (Volume: 14, Issue: 1, Jan. 1994)
12. Daydream.: [Електронний ресурс]. – Режим доступу: https://vr.google.com/daydream/. – Дата доступу : 01.05.2017.
13. Oculus SDK.: [Електронний ресурс]. – Режим доступу: https://developer.oculus.com/. – Дата доступу : 01.05.2017.
14. Gear Vr Framework.: [Електронний ресурс]. – Режим доступу: https://resources.samsungdevelopers.com/Gear_VR/020_GearVR_Framework_Project – Дата доступу : 01.05.2017.
15. Unreal Engine, Virtual Reality Best Practices: [Електронний ресурс]. – Режим доступу: https://docs.unrealengine.com/latest/INT/Platforms/VR/ContentSetup/index.html – Дата доступу : 01.05.2017.
16. Unity, VR Overview: [Електронний ресурс]. – Режим доступу: https://unity3d.com/ru/learn/tutorials/topics/virtual-reality/vr-overview?playlist=22946 – Дата доступу : 01.05.2017.
17. Куцаченко Н.Г. Сравнительный анализ визуализаторов для реализации проектов с поддержкой виртуальной реальности на платформе Android, Мультидисциплинарный научный журнал "Интернаука", Выпуск 17, 2017
18. Howard Rheingold, Virtual Reality: Exploring the Brave New Technologies, Simon & Schuster Adult Publishing Group, 1991
19. B. D. Lucas and T. Kanade. An iterative image registration technique with an application in stereo vision. In Seventh International Joint Conference on Artificial Intelligence, pages 674–679, Vancouver, 1981
20. G. Farneback. Two-Frame Motion Estimation Based on Polynomial Expansion. In SCIA’03 Proceedings of the 13th Scandinavian conference on Image analysis, pages 363-370, Halmstad, 2003
21. Michael W. Tao, Jiamin Bai, Pushmeet Kohli, and Sylvain Paris. SimpleFlow: A Non-iterative, Sublinear Optical Flow Algorithm. Computer Graphics Forum (Eurographics 2012), 2012.

22. OpenCV.: [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/OpenCV. – Дата доступу : 01.05.2017.
23. OpenCV – описание библиотеки компьютерного зрения.: [Електронний ресурс]. – Режим доступу: http://prog-master.com/opencv-opisanie/. – Дата доступу : 01.05.2017.
24. Deqing Sun, Stefan Roth, Michael J. Black Secrets of optical flow estimation and their principles, Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on
25. Куцаченко Н.Г. Сравнение алгоритмов трекинга движения при реализации средствами OpenCV, Системний аналіз та інформаційні технології: матеріали 19-ї Міжнародної науково-технічної конференції SAIT 2015, Київ 22-25 червня 2017 р.