

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
Факультет електроніки та інформаційних технологій

Кафедра комп'ютерних наук

Кваліфікаційна робота бакалавра

**ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДОДАТКУ  
ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ «УСТАНОВКА ОСУШКИ  
ПРИРОДНОГО ГАЗУ».  
МОДУЛЬ ВЗАЄМОДІЇ ІЗ ВІРТУАЛЬНОЮ РЕАЛЬНІСТЮ»**

Здобувач освіти гр. ІН-81

Данила КУЗЬМУК

Науковий керівник,  
кандидат фізико-математичних наук,  
доцент кафедри комп'ютерних наук

Сергій ШАПОВАЛОВ

Завідувач кафедри  
доктор технічних наук, професор

Анатолій ДОВБИШ

Суми 2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

**Кафедра комп'ютерних наук**

Затверджую \_\_\_\_\_

Зав. кафедрою Довбиш А.С.

“ \_\_\_\_\_ ” \_\_\_\_\_ 2022 р.

**ЗАВДАННЯ  
до випускної роботи**

Студента четвертого курсу, групи ІН-81 спеціальності 122-Комп'ютерні науки, денної форми навчання Кузьмука Д.А.

**Тема: «Інформаційне та програмне забезпечення додатку віртуальної реальності "Установка осушки природного газу". Модуль взаємодії із віртуальною реальністю»**

Затверджена наказом по СумДУ

№ \_\_\_\_\_ от \_\_\_\_\_ 2022 р.

**Зміст пояснювальної записки:** 1) Інформаційний огляд. Огляд предметної області. Актуальність створеного додатку. Постановка задачі; 2) Вибір методу рішення. Вибір апаратної платформи. Вибір програмного супроводу; 3) Практична реалізація. Створення віртуального простору. Налаштування елементів взаємодії з віртуальною реальністю.

Дата видачі завдання “ \_\_\_\_\_ ” \_\_\_\_\_ 2022 р.

Керівник випускної роботи \_\_\_\_\_ Шаповалов С.П.

Завдання прийняв до виконання \_\_\_\_\_ Кузьмук Д.А.

## РЕФЕРАТ

**Записка:** 43 стор., 19 рис., 1 табл., 1 додаток, 15 джерел.

**Об'єкт дослідження** — Застосування технологій VR в дистанційному освітньому середовищі..

**Мета роботи** — створення віртуального простору для процесу навчання операторів станції осушки природного газу.

**Методи дослідження** — дослідження аудиторії та порівняння сервісів аналогів.

**Результати** — розроблено програмне забезпечення, що відтворює у віртуальному просторі досвід роботи оператора станції осушки природного газу. Використання технології віртуальної реальності дозволило максимально перенести відчуття та емоції, які отримують спеціалісти обраної галузі під час своєї роботи. Модуль віртуальної реальності розроблений у середовищі рушія Unity та з використанням плагінів Open VR та VRTK.

ВІРТУАЛЬНА РЕАЛЬНІСТЬ, VR НАВЧАННЯ,  
НАВЧАЛЬНИЙ ПРОЦЕС, МОДЕЛЮВАННЯ,  
ПОСТ ОБРОБКА, UNITY, OPEN VR, VRTK

## ЗМІСТ

ВСТУП .....	6
1 ІНФОРМАЦІЙНИЙ ОГЛЯД .....	7
1.1 Огляд предметної області .....	7
1.2 Огляд подібних рішень.....	9
1.3 Актуальність створеного додатку .....	12
1.4 Постановка задачі .....	14
2 ВИБІР МЕТОДУ РІШЕННЯ.....	15
2.1 Вибір апаратної платформи .....	15
2.2 Вибір програмного супроводу .....	19
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ .....	22
3.1 Створення віртуального простору .....	22
3.2 Налаштування елементів взаємодії з віртуальною реальністю .....	31
ВИСНОВКИ .....	35
СПИСОК ЛІТЕРАТУРИ .....	36
ДОДАТОК А.....	38

## ВСТУП

Завдяки інформатизації і глобалізації соціальних процесів виник новий простір – віртуальний. Наразі, завдяки створенню глобальної комп'ютерної Інтернет-мережі, мільйони користувачів світу отримали доступ до численних інформаційних ресурсів і послуг. Мобільні додатки і віртуальна реальність стали частиною повсякденного життя сучасного людства. Технології віртуальної реальності широко застосовуються у найрізноманітніших сферах. Це і проектування та дизайн, будівництво і видобуток корисних копалин, військові технології, тренажери і симулятори, реклама, індустрія розваг тощо.

Не оминула небачених раніше технологічних викликів й освітня сфера. Онлайн-заняття і використання VR технологій стали сьогодні невід'ємною частиною учбового процесу, чим докорінно змінили увесь процес навчання. Зокрема, VR технології дозволять демонструвати 3D візуалізації, заохочувати слухачів під час заочного навчання, використовувати онлайн презентації, проводити практичні заняття тощо.

Відповідно до зазначеного, метою дипломної роботи є створення додатку VR «Установка осушки природного газу», розробка його віртуального простору та інтерфейс взаємодії із ним.

Для досягнення поставленої мети необхідно виконати такі задачі:

- провести аналіз предметної області та визначити актуальність дослідження;
- розробити усі необхідні моделі для проекту;
- створити головну сцену, розмістити на ній створені моделі. Додати небо, корекцію кольору та інші ефекти пост обробки;
- розробити і налаштувати основні механіки додатку.

# 1 ІНФОРМАЦІЙНИЙ ОГЛЯД

## 1.1 Огляд предметної області

З кожним роком новітні технології стають все більш доступними для пересічного користувача. Якщо раніше говорили, що VR – це розробка майбутнього, то сьогодні з цією галуззю знайомі не лише програмісти та геймери, а також й пересічні люди.

Згідно зі статтею у відомому англomовному журналі Wired, який висвітлює теми науки та технологій, потужна популярність VR технологій розпочалась завдяки Oculus Rift – гарнітури, випущеної у 2012 році, яка підключалась до монітору користувача та занурювала його у трьохвимірну реальність. З появою таких гарнітур і можливостей ми переосмислили не тільки наше сприймання відеоігор, які перейшли у 3D простір, а й змінили розуміння усього цифрового простору в цілому [1].

Ця підвищена доступність зробила віртуальну реальність доступним варіантом не тільки для розважальної сфери (кіноіндустрія, реклама, туризм, ігри), а й для бізнесу (девелопмент та продаж/оренда нерухомості, маркетинг) державного (військова промисловість, медицина, освіта) і неурядового секторів тощо. Тож працювати та розробляти VR продукти, додатки є наразі не тільки прибутковим бізнесом, а й актуальним рішенням окремих проблем певних галузей, переосмисленням традиційних підходів у вищеперерахованих сферах діяльності.

Доволі широке використання, значний дослідницький та інноваційний потенціал отримали VR технології у навчальному процесі. Наразі перед освітньою сферою постає низка нагальних задач, з-поміж яких провідним

завданням виступає доступність навчання. Сьогодні учні та студенти в усьому світі мають унікальну можливість відвідувати заняття та навчатися дистанційно. Це надає освіті такої важливої характеристики, як зняття обмежень, зумовлених географічними кордонами, часовими поясами та фізичними факторами. Крім того, сучасні виклики (зокрема, поширення світом протягом останніх трьох років пандемії COVID-19, вторгнення у лютому 2022 року РФ до України та початок повномасштабної війни та ін.) вплинули на те, що людство загалом та учасники освітнього процесу зокрема втратили змогу спілкуватися, працювати і навчатися наживо. Саме зазначені події дали потужний поштовх і вплинули на подальший розвиток дистанційного навчання.

«Використання VR у класах, за прогнозами, різко зросте протягом наступних п'яти років» – зазначається у статті веб-сайту технічного спрямування Built In [2]. Використання віртуальної дійсності дозволило учням отримати найбільш захоплюючий і цікавий досвід навчання. Віртуальна реальність може перенести студентів зі своїх парт у віддалені куточки планети, дозволити їм змішувати легкі хімічні речовини та спостерігати за реакцією без фізичної шкоди, а також дозволяє їм не лише бачити, а й взаємодіяти з віртуальними світами. У результаті роль вчителя зміщується від надання змісту до сприяння навчанню. «Вчителі будуть зосереджені на створенні умов для вивчення, а не на наданні готових знань», – пише Нік Бабич – UI/UX дизайнер [3]. Крім того, він зауважує, що віртуальна реальність має наступні шість переваг:

- найкраще почуття місця: учні можуть вивчати предмет, «проживаючи» його;
- масштабне навчання: викладачі можуть створювати віртуальні лабораторії, щоб скоротити витрати та підвищити доступність;



- практичне навчання: учні можуть навчатися, виконуючи завдання, а не просто читаючи, набуваючи практичного досвіду та навичок;
- емоційна реакція: викладачі створюють для учнів незабутні враження, щоб покращити їхнє засвоєння;
- творчий розвиток;
- візуальне навчання. Педагоги можуть покращити візуальне розуміння освітнього контенту учнями [3].

Віртуальна реальність також розширює можливості для студентських екскурсій та лабораторних досліджень. Колись освітні екскурсії були обмежені вартістю, відстанню та доступністю, але віртуальна реальність усуває ці бар'єри та надає безмежні можливості.

## **1.2 Огляд подібних рішень**

У міру того, як віртуальна реальність стає все більш актуальною, доступною та корисною для навчального середовища, розвивається і ринок VR технологій. Вже існують багато проектів для вивчення історії, хімії, анатомії, археології, іноземних мов тощо.

Вивчаючи матеріал про те, яким має бути додаток VR для освітніх цілей, було виділено наступні п'ять рис:

- Іммерсивний. Дизайнери повинні прагнути створити відчуття, що користувачі перебувають у процесі. Наприклад, якщо ви розробляєте програму з історії, оживіть її для учнів.
- Легкий у використанні. Слід позбавитися необхідності мати спеціальні навички для взаємодії з додатком.

- Значний. Сенс справді важливий для студентів. Ви не можете створити хороший досвід навчання у віртуальній реальності без гарної історії.

- Адаптивний. Як одного разу сказав Альберт Ейнштейн: «Я ніколи не вчу своїх учнів, я лише намагаюся створити умови, в яких вони можуть навчатися». Досвід віртуальної реальності має дозволити учням досліджувати у своєму власному темпі. Додаток має забезпечувати повний контроль за рівнем складності. Дизайнери повинні встановити, як учні навчаються, а потім використовувати ці знання для розробки продуктів віртуальної реальності, які забезпечують ефективне навчання.

- Вимірюваний. Кожен освітній інструмент повинен надавати вимірюваний вплив. Вчителі повинні мати можливість відслідковувати показники освіти, щоб вони могли виміряти знання з предмета. При розробці досвіду для VR-освіти важливо вибрати відповідні показники та чітко вказати, який критерій використовуватиметься для вимірювання успіху та невдачі [4].

Крім того, для наочності та розуміння того, яким чином має виглядати створений нами продукт, було досліджено декілька аналогів VR додатків для навчання.

Перший – це Body VR [5] – освітній досвід віртуальної реальності, який переносить користувача у людське тіло (рис. 1.1). Користувач може мандрувати кровотоком і вивчати, як клітини крові розподіляють кисень по всьому тілу, яким чином функціонують органели, які хімічні реакції відбуваються з людським тілом всередині (рис. 1.2).

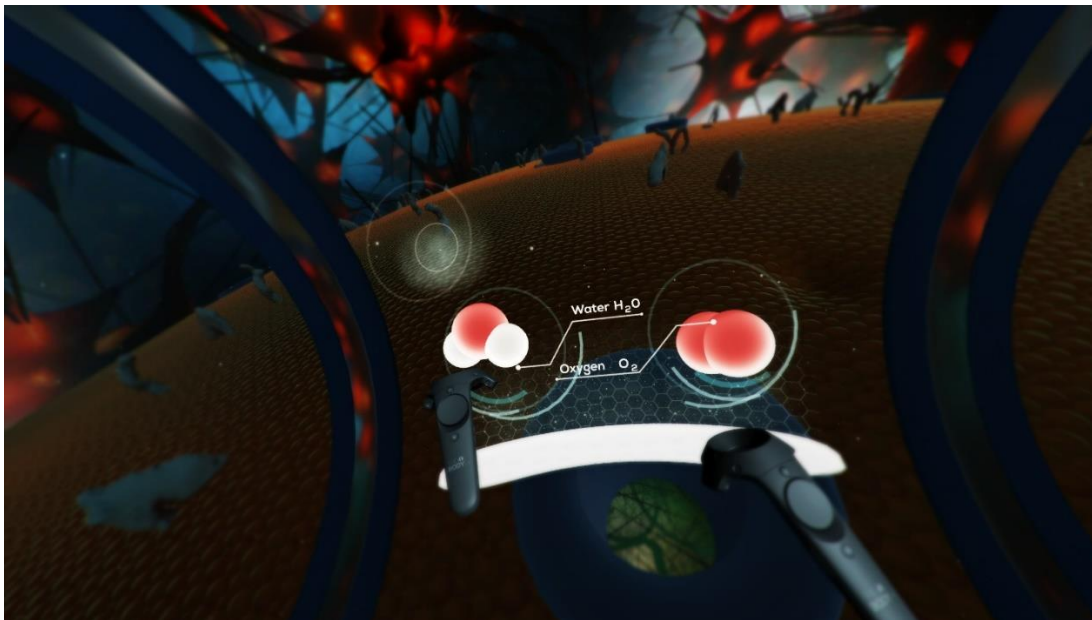


Рисунок 1.1 – Додаток Body VR

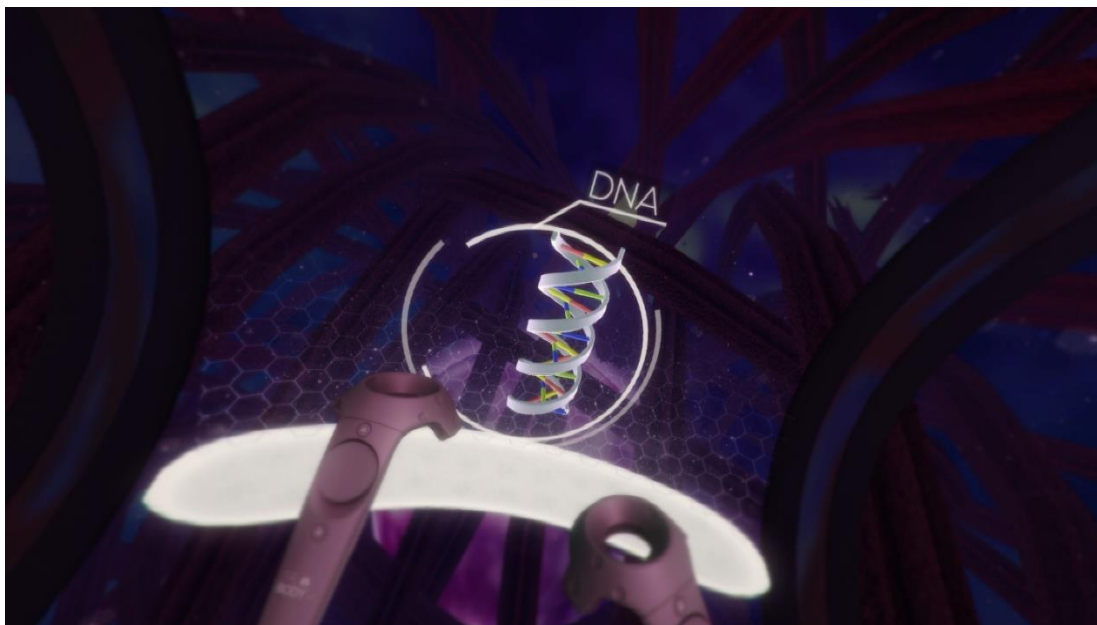


Рисунок 1.2 – Вивчення клітин у додатку Body VR

Другий додаток для розгляду Mondly: Learn Languages in VR [6] – вивчення мов у 3D реальності (рис. 1.3). Користувач має можливість вивчати одну із запропонованих 30 мов. Для ефективного вивчення були створені сценарії, ніби студент опиняється в життєвій ситуації з реальними людьми, де

веде з ними розмову на певну тему: замовляє напій в кафе, їде на таксі, бронює готельний номер тощо.



Рисунок 1.3 – Один із сценаріїв Mondly: Learn Languages in VR

### 1.3 Актуальність створеного додатку

Протягом останніх років ми переконалися, що наявність дистанційного повного навчання – це не просто додаткова перевага закладів освіти, а й адекватна відповідь на гострі виклики сьогодення. Не складно перевести теоретичний матеріал в онлайн, підготувати презентацію, оформити інформаційну частину уроку, проте нагальним залишається питання: як провести практичне/лабораторне заняття, в межах якого викладач має перевірити рівень теоретичної підготовки студентів або учнів, сформулювати в

них навички і компетенції, допомогти напрацювати практичний досвід та інше?

Зазначене актуалізує необхідність розробки додатків у віртуальній реальності, де викладач та студент/учень отримують можливість взаємодіяти один з одним та з системою, що вивчають, проходити різні сценарії, навчатися на конкретних, а не теоретичних, прикладах, отримувати наочний досвід, який неможливо отримати при використанні традиційних засобів.

Також є певний перелік професій, які можуть наражати на небезпеку людей, які працюють у даних сферах. Особливо складним є підготовка спеціалістів таких галузей. Проходження практики на підприємствах, виконання завдань на реальних установках може бути небезпечним для студентів, адже вони не мають достатньо напрацьованого досвіду для управління обладнанням, тим паче не мають можливість пройти усі аварійно-небезпечні, нестандартні ситуації в реальних умовах.

Тому важливо створити додаток, який буде представляти повний практичний досвід, буде включати необхідний інструментарій та сценарії, які у реальному житті неможливо відтворити не наражаючи на небезпеку життя та здоров'я студента та оточуючих.

Звичайно, досвід отриманий з використанням таких технологій не може бути настільки повним, як у реальному житті, проте його наочність, яка максимально відтворює справжні відчуття, допоможе оперативно відреагувати та виконати правильний алгоритм дії у подальшому.

## 1.4 Постановка задачі

Необхідно розробити додаток для симуляції роботи установки по очищенню природного газу з використанням можливостей віртуальних технологій.

Для цього необхідно виконати такі *завдання*:

1. Розробити усі необхідні моделі для проекту.

2. Створити головну сцену, розмітити на ній щойно створені моделі.

Зробити картинку більш привабливою для очей: додати небо, корекцію кольору та інші ефекти пост обробки.

3. Розробити і налаштувати основні механіки: переміщення у віртуальному просторі, ліфт до пункту керування, взаємодія із вентилями контролю установки, виклик аварійної ситуації.

## 2 ВИБІР МЕТОДУ РІШЕННЯ

### 2.1 Вибір апаратної платформи

Світ новітніх технологій не стоїть на місці і повсякчас демонструє все нові і нові досягнення людського мислення. Так, наприклад, звичайним 3D зображенням у кінотеатрі вже нікого не здивуєш, а ось переміститися у віртуальну гробницю фараона – це все зовсім інші емоції та відчуття.

Перші розробки віртуальної реальності датуються дев'яностими роками попереднього століття, проте справжній «бум» інноваційних VR-пристроїв проявився саме у наш час. Сучасні окуляри та шоломи віртуальної реальності (або ж їх узагальнена назва VR-гарнітура), надають якісну, яскраву та чітку картинку, а найголовніше, вони стали доступними усім і кожному.

На сьогоднішній день, на полицях магазинів можна знайти широкий спектр VR-гарнітур різного рівня технологічності та якості, кожен з яких має як свої переваги так і недоліки. Найяскравішими представниками є:

1. Valve Index – дуже прогресивна і технологічна модель споживацьких VR-комплектів. Має підвищену роздільну здатність дисплеїв та частоту їх оновлення. Використовує базові станції для відстеження положення користувача. Контролери мають вбудовані сенсори відстеження положення пальців людини, що відкриває новий рівень взаємодії із віртуальним простором.

2. HTC Vive – комплект, що зарекомендував себе часом. Відповідно до цього має величезну сумісність із різними додатками та бібліотеками та отримав свою популярність у звичайних користувачів, розробників ПЗ та

ентузіастів. Проте, на даний момент, використовуються доволі застарілі рішення та технології.

3. HTC Vive Cosmos – оновлена версія попереднього шолому. Має підвищену роздільну здатність та частоту роботи дисплеїв, вбудовані навушники, оновлені контролери. Головна перевага – це можливість використовувати гібридний режим роботи, тобто відпадає необхідність у базових станціях.

4. Oculus Quest 2 – на відміну від попередніх представників є повністю автономним пристроєм, адже має у собі вбудований комп'ютер. Даний факт суттєво впливає на кінцеву вартість продукту (тому що для роботи не потрібен ПК), однак, звичайно, втрачається потужність системи. Усі інші параметри є середніми для даного класу пристроїв.

Враховуючи усі переваги та недоліки кожного із вищеописаних продуктів, ми віддали перевагу останньому варіанту, а саме Oculus Quest 2 (Рисунок 2.1) як компроміс у ціні, продуктивності та якості. Вирішальним фактором стала повна автономія пристрою, що суттєво впливає на мобільність користувача, яка у свою чергу є основою якісного дистанційного навчання. До того ж дана гарнітура має вбудований мікрофон та навушники, що дозволить налагодити голосовий зв'язок між користувачами системи.



Рисунок 2.1 – Комплект віртуальної реальності Oculus Quest 2



Для повноцінного використання технології необхідна вільна площа мінімум півтора метри завширшки та завдовжки для не скутого нічим переміщення. Також бажано прибрати всі предмети навкруги, щоб випадково не зачепити їх у процесі навчання. Після підключення необхідно провести калібрування та побудову стін-оберігачів (рис. 2.2) для побудови віртуального простору користувача.

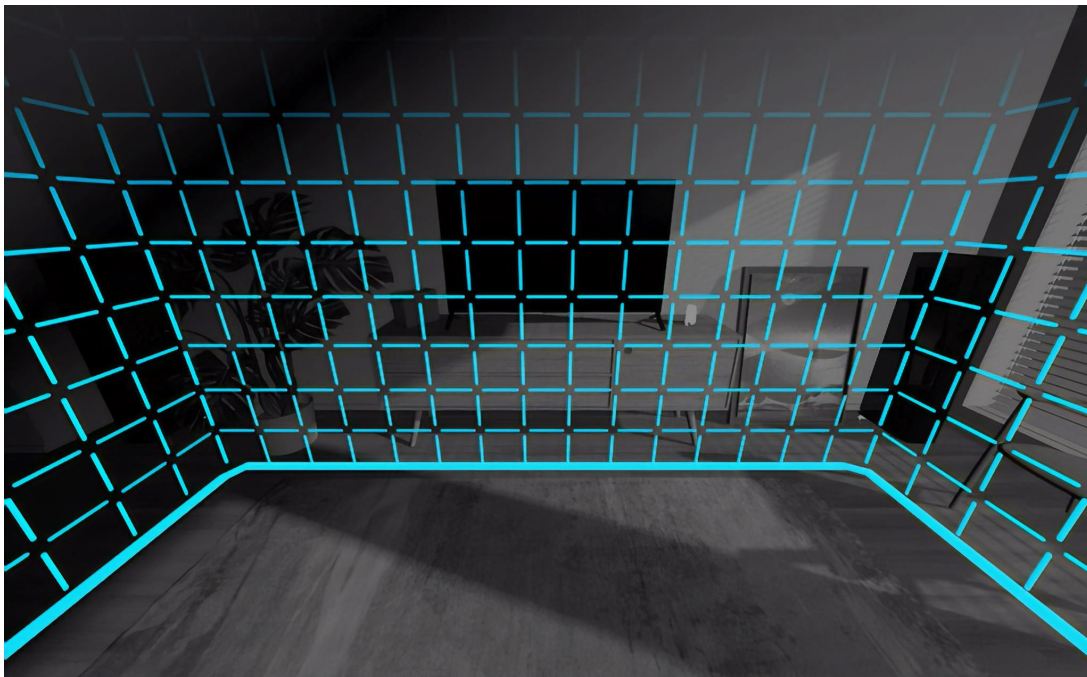


Рисунок 2.2 – Побудова віртуального простору

Незважаючи на те, що ми орієнтуємося на автономний пристрій, треба поцікуватися про максимальну сумісність із усіма можливими VR-гарнітурами для зменшення порогу входження до навчання у віртуальній реальності. Тому потрібно врахувати необхідні вимоги для запуску нашого додатку на шоломах, що працюють у тандемі із персональним комп'ютерами. Звернувшись до офіційного сайту HTC Vive [7] можемо знати відповідні системні вимоги (таблиця 2.1).

Таблиця 2.1 Системні вимоги для роботи шолому HTC Vive

<b>Компонент</b>	<b>Рекомендовані системні вимоги</b>	<b>Мінімальні системні вимоги</b>
Процесор	Intel Core i5-4590/AMD FX 8350 аналогічна або більш нова модель	Intel Core i5-4590/AMD FX 8350 аналогічна або більш нова модель
Графічний процесор	NVIDIA GeForce GTX 1060, AMD Radeon RX 480 аналогічна або більш нова модель	NVIDIA GeForce GTX 970, AMD Radeon R9 290 аналогічна або більш нова модель
Пам'ять	4 ГБ ОЗП або більше	4 ГБ ОЗП або більше
Відеовиходи	HDMI 1.4, DisplayPort 1.2 або більш нова модель	HDMI 1.4, DisplayPort 1.2 або більш нова модель
Порт USB	1x USB 2.0 або більш нова модель	1x USB 2.0 або більш нова модель
Операційна система	Windows 7 SP1, Windows 8.1 або вище, Windows 10	Windows 7 SP1, Windows 8.1 або вище, Windows 10

Так як навіть комп'ютер, що відповідає мінімальним системним вимогам є потужнішим за можливості вбудованого у Oculus Quest 2 чіпу, на який ми орієнтуємося, то саме такий ПК можна прийняти за точку відліку для пошуку обладнання.

## 2.2 Вибір програмного супроводу

Для розробки додатків із використанням технологій віртуальної реальності існує безліч рішень, адже на сьогоднішній день є готовий відкритий стандарт розробки VR/AR додатків – OpenXR. Проте, серед усього цього різноманіття програмних інструментів для невеликої команди розробників вибір зводиться до двох продуктів:

1. Unreal Engine
2. Unity

Обидва ці додатки, в першу чергу, є ігровими рушіями, але їх можливості є майже необмеженими при правильному використанні. Тому немає жодних перешкод для створення освітніх додатків за допомогою інструментарію, який надають дані продукти.

Так як принцип роботи із віртуальною реальністю у цих двох рушіїв однаковий, при виборі необхідно покладатися на інші фактори. Для цього оцінимо переваги та недоліки кожного з додатків.

Unity:

- 1) використання стандартної мови програмування C#;
- 2) Drag & Drop інтерфейс всередині програмного простору;
- 3) величезна бібліотека готових асетів і плагінів (як платних, так і безкоштовних) для розробки власних додатків.

Unreal Engine:

- 1) система візуального програмування;
- 2) потужна система постобробки та симуляція частинок;
- 3) висока якість візуальних ефектів;
- 4) просунута система анімації.

Оскільки склад команди розробників невеликий, простота розробки та наявність великої кількості бібліотек відіграють вирішальну роль. До того ж, ми проектуємо додаток під доволі слабе залізо Oculus Quest 2, через що графіка не є нашим пріоритетом. Тому рушій Unity є ідеальним варіантом, який покриває усі наші запити та потреби проекту.

Для забезпечення відчуття реальності та надання максимально правдоподібного оточення необхідно розробити модель станції осушки газу. Вона складається із декількох цистерн циліндричної форми, колони обробки газу, ліфту для підняття персоналу, а також загороджень, щоб обмежити переміщення користувача та запобігти бажанню вийти зі станції незапланованими шляхами. Дані об'єкти на базовому рівні являють собою примітивні геометричні фігури, з простими текстурними мапами, отже для нашого програмного продукту не потрібні потужні або ж специфічні засоби. Для виконання поставленої задачі добре підходить Autodesk 3ds Max, особливо беручи до уваги, що для даного додатку можна отримати безкоштовну ліцензію для використання у начальних цілях.

Звичайно, даного набору інструментів вже цілком достатньо для втілення у реальність поставлених завдань, проте написання проекту з «нуля» займатиме безліч людино-годин. Тому для покращення досвіду роботи із системою Unity необхідно додати декілька основних плагінів.

Найголовнішим плагін, навколо якого будуються фундамент нашого додатку, є OpenVR Unity XR[8]. Це розширення забирає на себе усі питання щодо сумісності шолому та нашого додатку, дозволяє відстежувати положення голови та контролерів, а також зручно налаштувати керування. Головне його завдання – надати можливість рендерингу на всіх основних пристроях віртуальної реальності через один інтерфейс. У комплекті ми

отримуємо документацію, а також декілька тестових сцен для перевірки працездатності та основних можливостей.

Додатково використаємо розширення під назвою Virtual Reality Toolkit [9]. Даний плагін – це набір готових рішень, які дозволяють швидко та легко створювати VR додатки в Unity. Він охоплює ряд поширених рішень, таких як:

- 1) пересування у віртуальному просторі;
- 2) взаємодії, такі як дотик, захоплення та використання об'єктів;
- 3) взаємодія з елементами інтерфейсу Unity;
- 4) фізика тіла у віртуальному просторі;
- 5) 2D та 3D елементи керування, як-от кнопки, важелі, двері, ящики

тощо.

В результаті вище описаних систем ми можемо скомпонувати модель взаємодії використаного програмного забезпечення у вигляді діаграми (рис. 2.3).

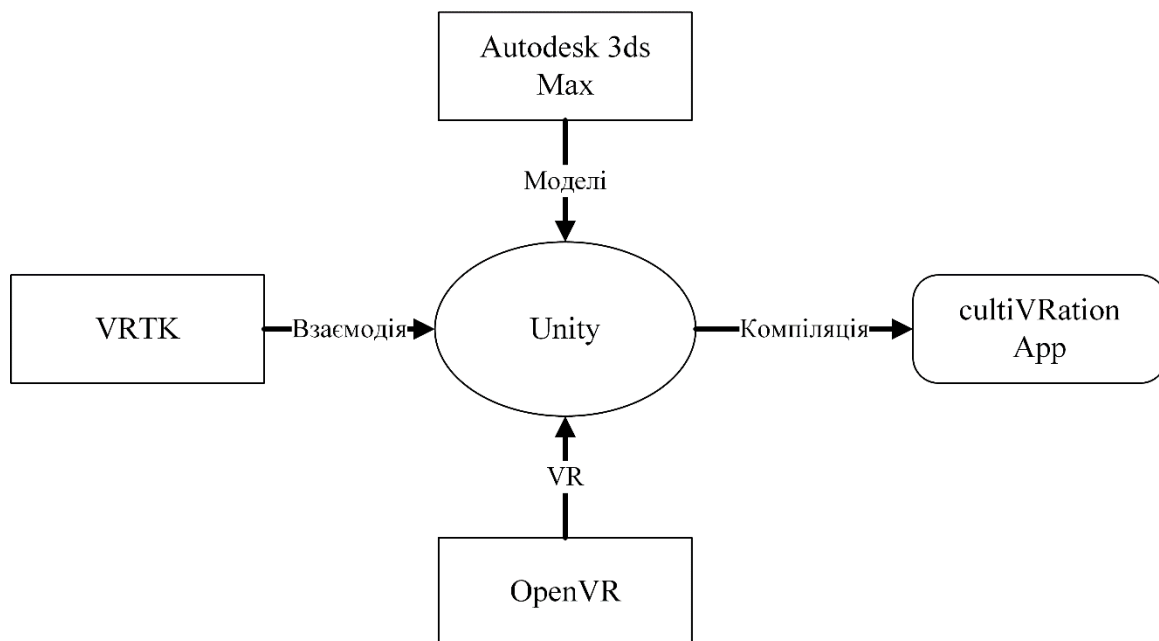


Рисунок 2.3 – Модель взаємодії програмних ресурсів при створенні проекту

### 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ

#### 3.1 Створення віртуального простору

Перш за все необхідно розробити основну модель станції, в рамках якої і буде проходити основна дія нашого додатку. Оскільки дана структура є досить специфічним об'єктом, а наша мета – це максимально достовірно передати досвід роботи оператора станції, для створення зразка установки потрібно провести певні дослідження. Для цього ми звернулися до вільних ресурсів, доступних у мережі Інтернет, для вивчення особливостей будови таких станцій. В результаті був отриманий перелік зображень (Рисунок ), які найбільш точно відображають об'єкт нашого дослідження, і можуть бути використані як шаблон для моделювання. Саме тому вони лягли в основу побудови майбутніх моделей.



Рисунок 3.1 – Приклади-шаблони для створення майбутньої 3-д моделі

Визначившись з основними елементами установки, такими як цистерни для зберігання газу, компресор, газоперекачувальні труби, колони та їх розміщення, можна створювати нашу модель відповідно до знайдених раніше шаблонів. Головним фактором, який потрібно врахувати при моделюванні, є визначення деякого простору для вільного переміщення та запобігання відчуття замкнутості. В результаті, враховуючи усі вищезначені параметри, отримуємо таку модель (рис. 3.2).

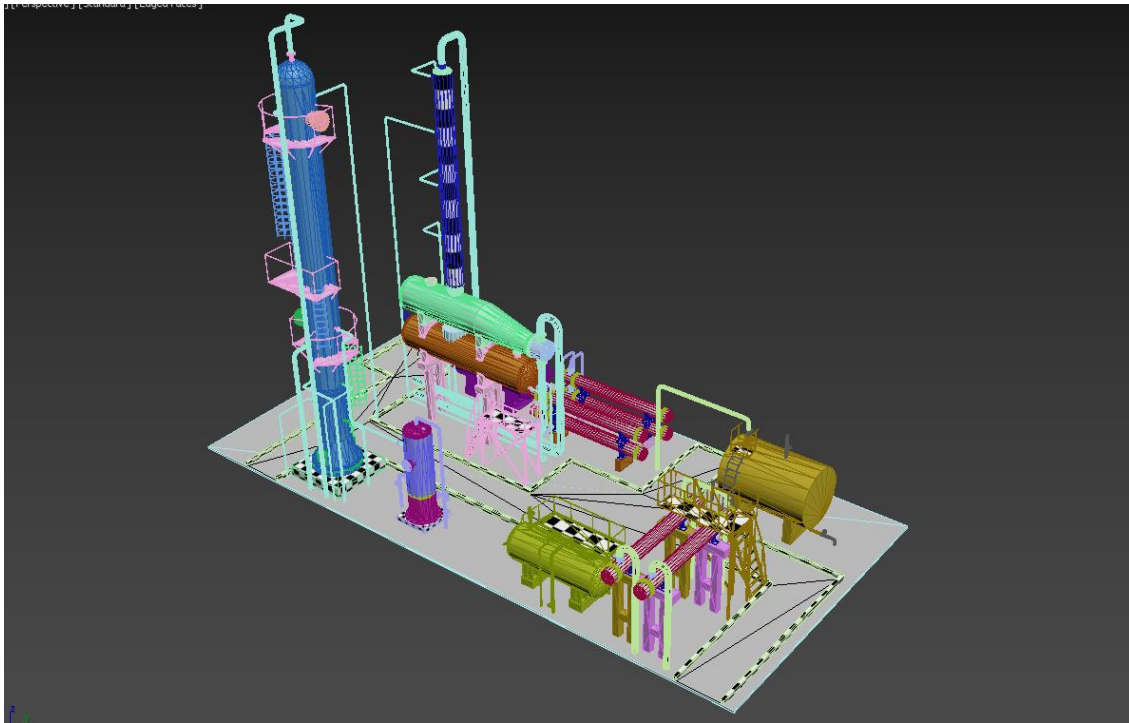


Рисунок 3.2 – Приклади-шаблони для створення майбутньої 3-д моделі

Експортуємо змодельовану станцію в зрозумілий для Unity формат (в нашому випадку fbx) та імпортуємо у завчасно створений проект додатку. Усі інші невеликі моделі, які нам можуть знадобитися під час розробки, можна знайти у відкритих банках моделей або ж у Asset Store.

Тепер необхідно пошукати про текстури для нашої новоствореної моделі та налаштування її матеріалів. Зараз у просторі Інтернету можна знайти безліч ресурсів, які за невелику плату або навіть безкоштовно надають текстури високої якості саме для подібних цілей. Скористаємось одним із таких сервісів та підберемо текстури металу, плитки для тротуару, землі та інше. Використовуючи отримані зображення налаштуємо всі необхідні матеріали, додаючи мапи та конфігуруючи їх інтенсивність для отримання результату, який нас влаштує (рис. 3.3).

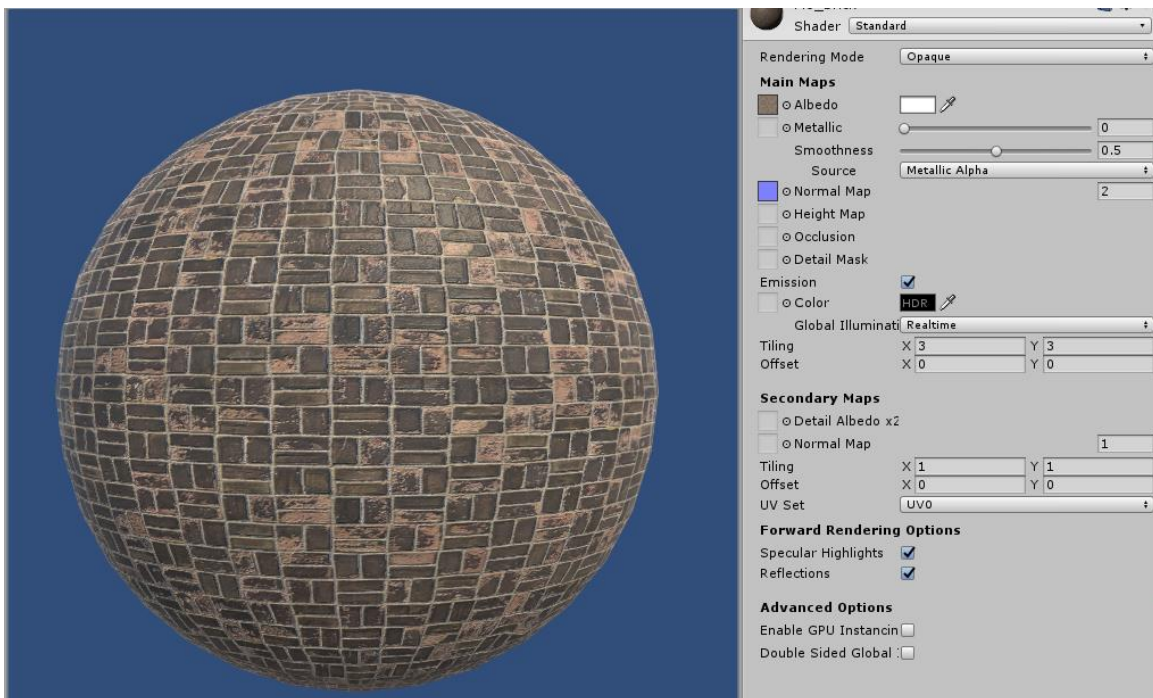


Рисунок 3.3 – Налаштування матеріалу

Після налаштування усіх матеріалів необхідно накласти їх на компоненти нашої моделі (рис. 3.4).





Рисунок 3.4 – Модель станції осушування з накладеними матеріалами

Тепер наша модель виглядає привабливо та реалістично. Однак вихідне зображення ніяк не зможе переконати людину, що вона знаходиться на справжньому підприємстві. Задля вирішення поставленого питання потрібно додати декілька останніх штрихів для надання кінцевого вигляду.

По-перше, налаштуємо камеру для рендеру неба (Skybox). Таким чином, ми переносимо ще один елемент із реального світу у віртуальний, а також створюємо відчуття того, що там, далеко за станцією, розгортаються поля та ліси.

Другим етапом налаштуємо декілька ефектів пост обробки. Тим інструментом ,що додасть глибини нашій картинці, стане Ambient Occlusion. Він надає нашому зображенню додаткові тіні від об'єктів, згладжуючи гострі грані між компонентами та роблячи картинку єдиним цілим. Також увімкнемо легку кольорову корекцію.

В результаті матимемо вже таку картинку (рис. 3.5), яка в шоломі віртуальної реальності здатна розмивати грань між реальним та віртуальним світами.



Рисунок 3.5 – Зображення з даванням Skybox та Post Processing

Зображення вже є прийнятним, але все ж створена нами реальність є занадто статичною. Необхідно якимсь чином оживити зображення. Для цього скористаємось невеликим безкоштовним плагіном Living Birds [10]. Даний інструмент має усе необхідне для того, щоб додати на нашу сцену літаючих сторінь. Нам, як розробнику, потрібно створити об'єкт контролеру птахів, налаштувати їх вигляд та кількість, а також розмітити контрольні точки (рис. 3.6) між якими вони будуть літати. Така система, яка на перший погляд здається простою та непомітною, але суттєво впливає на сприйняття оточення та викликає «вау» ефект.



Рисунок 3.6 – Налаштована система Living Birds

Задля підвищення ефективності навчання за допомогою подібного додатку, необхідно наглядне демонстрування протікаючих процесів. В нашому випадку це відображення осушки природного газу всередині станції. Однак колона, де все відбувається, є закритою та герметичною, а зробити її відкритою означає втратити відчуття реальності у сцені і відтворити нереальну ситуацію. Для вирішення даного питання було додано спеціальний планшет, який ігнорує рендер стінок, які заважають побачити процес осушки. Для цього обираємо об'єкти, які необхідно ігнорувати, та задаємо їм власний шар (рис. 3.7).

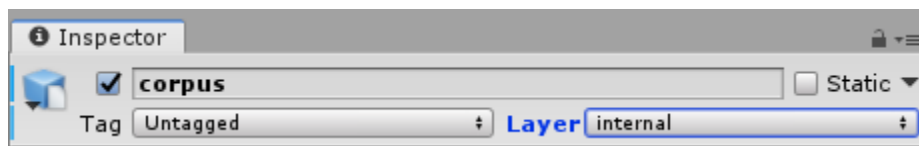


Рисунок 3.7 – Встановлення шару об'єкта

Тепер створюємо об'єкт планшета, один із матеріалів якого буде використовувати RenderTextures. В даний матеріал ми будемо транслювати зображення, яке відобразиться на екрані планшета. Додатково створюємо камеру, робимо її дочірньою до планшета для синхронізації їх рухів. Останнім кроком налаштуємо камеру так (рис. 3.8), щоб її зображення транслювалося у попередньо створену текстуру, а об'єкти, які ми бажаємо приховати, не відображалися.

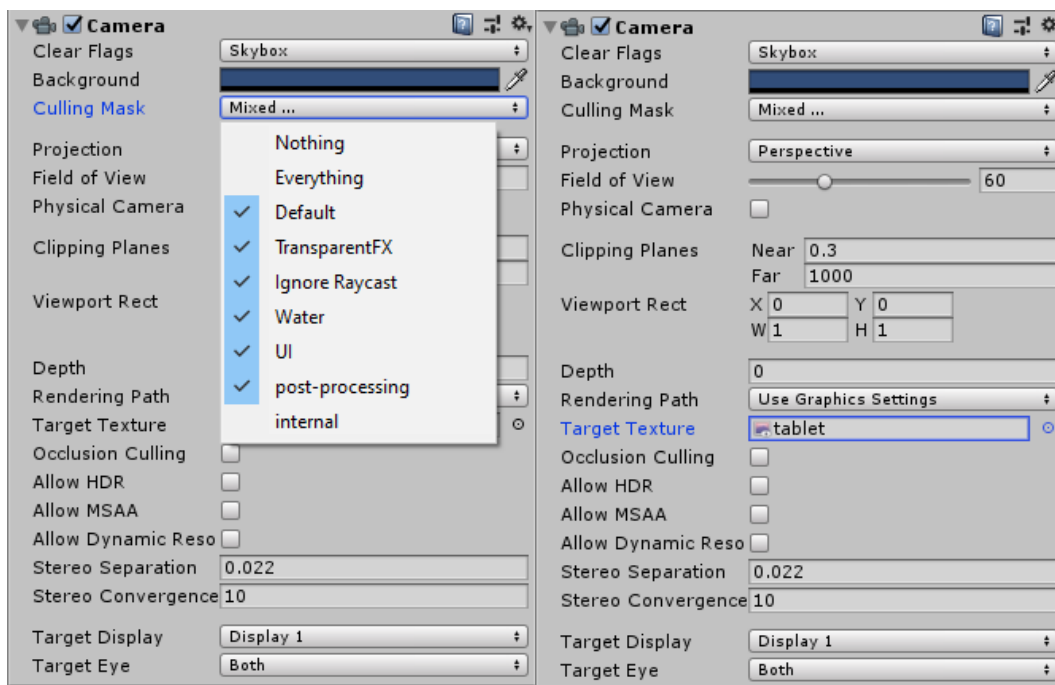


Рисунок 3.8 – Налаштування камери планшета

В результаті маніпуляцій через планшет ми можемо побачити вміст колони, тому потрібно симулювати процеси протікаючі всередині. Так як осушка газу викликає розпилення у вигляді туману, скористаємося вбудованою у Unity системою частинок. Налаштуємо його так, щоб утворювався стовп сірих частинок, які підніматимуться угору. Також створимо симуляцію

руху рідини. Для цього створимо скрип, який рухатиме текстуру води об'єкту кожен кадр. Код має такий вигляд:

```
[SerializeField]
private int targetMaterialSlot = 0;
[SerializeField]
private float speedY = 0.5f;
[SerializeField]
private float speedX = 0.0f;
private Renderer myrender;
private float timeWentX = 0;
private float timeWentY = 0;
private Vector2 targetOffset;

void Start()
{
    myrender = GetComponent<Renderer>();
}

void Update()
{
    timeWentY += Time.deltaTime * speedY;
    timeWentX += Time.deltaTime * speedX;
    targetOffset = new Vector2(timeWentX, timeWentY);
    myrender.material.SetTextureOffset("_MainTex", targetOffset);
}
```

Таким чином, якщо подивитися на газо-обробну колонну, через планшет, що прив'язаний до контролера, то можна детально роздивитися процес осушення природного газу (рис. 3.9).

Також наш додаток включає виконання сценарію аварії на станції. Необхідно реалізувати візуальне оформлення і для даної ситуації. Для цього додаємо лампочку на панелі керування станцією, яка буде світитися червоним світлом під час аварії. Також доцільним є додати звук сирени, щоб помістити користувача у стресову ситуацію, аналогічно тому, як це могло статися у житті. Якщо студент не встигає полагодити станцію, додається ефект розгерметизації системі із бризками рідини, димом та іскрами (рис. 3.10), виконаними за допомогою тієї самої системи частинок.



Рисунок 3.9 – Наглядний процес осушки газу



Рисунок 3.10 – Ефекти аварії на станції

### 3.2 Налаштування елементів взаємодії з віртуальною реальністю

Перш за все, необхідно розробити систему переміщення користувача. На даний момент найбільш інтуїтивною і зрозумілою для більшості людей є телепортація. Так, по натисканні кнопки на контролері має з'явитися промінь (рис. 3.11), який необхідно навести на потрібне місце та відпустити кнопку.



Рисунок 3.11 – Телепортація у VR просторі

Даний механізм реалізований у доданому нами пакеті VRTK. Для цього необхідно перенести на сцену два компоненти `VRTK_Scripts` та `VRTK_SDKManager`, і цього вже достатньо, щоб використовувати примітивну телепортацію. Однак, в такому випадку користувач може переміститися на будь-яку поверхню, що є неприпустимим та нищить занурення у віртуальність. Саме тому, необхідно додати фільтрацію допустимих переміщень. Для цього обираємо усі поверхні, на які дозволене переміщення, та надаємо їм

спеціальний тег `floor`. Після цього, при спробі навести на недоступну поверху, колір променя змінюється на червоний (рис. 3.12), а при відпусканні кнопки користувач залишається на місці.



Рисунок 3.12 – Заборона на телепортацію

Для наочності створимо власний вказівник променя. До стандартної точки додаймо зображення слідів від взуття. Таким чином, для студентів буде легше зрозуміти основну функцію переміщення. Однак, просто прив'язати рендер плоского зображення (`Sprite Renderer`) недостатньо. Так як кут існує між поверхнями та контролером, а також напрям променя завжди різний, необхідно розробити скрипт, який вирівнюватиме положення слідів відносно положення користувача:

```
public class FootHorizontal : MonoBehaviour
{
    public Transform controller;

    void Update()
```



```

{
    transform.eulerAngles = new Vector3(90, controller.eulerAngles.y,
transform.eulerAngles.z);
    transform.position = new Vector3(transform.parent.transform.position.x,
transform.parent.transform.position.y + 0.1f, transform.parent.transform.position.z);
}
}

```

Таким чином, вказівник телепортації матиме наступний вигляд (рис. 3.13).

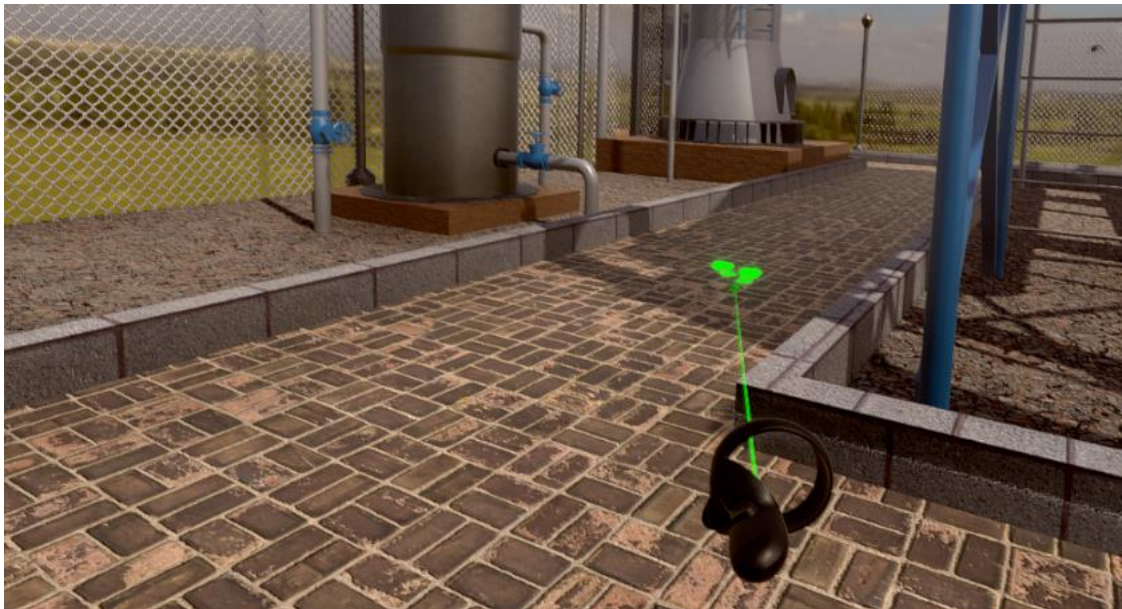


Рисунок 3.13 – Оновлений вказівник луча

Наступним етапом є реалізація ліфту, який підійматиме користувача до панелі керування. Для цього, відстежуємо коли об'єкт користувача заходить до кабіни та починаємо рух угору. Так як наш віртуальний користувач не має фізичної моделі, підіймати його за допомогою колайдерів неможливо. Таким чином, єдиним варіантом для нас залишається змінювати координати користувача разом із кабіною ліфту.

Підійнявшись на ліфті, користувач отримує доступ до контрольної панелі. Тут розташовані 6 вентилів, які були реалізовані на основі подібних

засобів інтеракції доступних у VRTK. Однак, їх необхідно модифікувати для наших цілей, а саме додати підписи із процентами для наочності, обмежити граничні положення, а також відслідковувати зміну значень. Для взаємодії із даними вентилями використовується «курок» та обертання контролера навколо своєї осі. Таким чином користувач робить рух подібний тому, як людина обертає справжній кран.

Користувачі, що мають право адміністратора, мають можливість розпочати аварійну ситуацію по натисканні спеціальної кнопки. Для того, щоб відслідкувати дії користувача, у OpenVR є спеціальний скрипт, що містить функції, які викликаються при натисканні кожної окремої кнопки. Тому додаємо код, що перевіряє права адміністратора та починає сценарій аварійної ситуації:

```
private void DoGripPressed(object sender, ControllerInteractionEventArgs e)
{
    GameObject adminObject = GameObject.FindGameObjectWithTag("Admin");
    SetAdmin adminComponent = adminObject.GetComponent<SetAdmin>();
    if (adminComponent.isAdmin)
    {
        AccidentController.instance.startAccident();
    }
}
```

Після натискання запускається таймер, за який студент має виставити вентилі у правильне положення для запобігання розгерметизації системи. У випадку коли час закінчився і користувач не зміг встановити вірні значення вмикаються ефекти описані у розділі 3.1.

## ВИСНОВКИ

У даній кваліфікаційній бакалаврській роботі висвітлено проблематику створення додатку для відпрацьовування практичних навичок студентів, які є спеціалістами різних галузей, особливо пов'язаних із ризиком для життя та здоров'я. Для прикладів ми розглянули різні рішення-аналоги, які є дещо схожими на наш проект. Завдяки цьому ми сформуваємо актуальність розробки та чіткі завдання до виконання.

У другому розділі ми розглянули різні варіанти гарнітур для обрання у якості апаратного супроводу та зупинили свій вибір на Oculus Quest 2 як на компромісі у ціні, продуктивності та якості. Крім цього, необхідно звернути увагу на вибір програмного супроводу. Оскільки склад команди розробників невеликий, простота розробки та наявність великої кількості бібліотек відіграють вирішальну роль ми зробили вибір на користь Unity. До того ж ми проектуємо додаток під доволі слабке залізо Oculus Quest 2, через що графіка не є нашим пріоритетом.

У третьому розділі детально описано усі кроки роботи над створенням віртуального простору та налаштування елементів взаємодії з віртуальною реальністю. Додатково наведені елементи коду та скріншоти для демонстрації роботи.

Даний проект вже є кінцевим продуктом, але має великий потенціал для розвитку та впровадження нових функцій, сценарію тощо.

## СПИСОК ЛІТЕРАТУРИ

1. Peter Rubin. The WIRED Guide to Virtual Reality [Electronic resource] // Wired, 2020. – Режим доступу до ресурсу: <https://www.wired.com/story/wired-guide-to-virtual-reality/>
2. Stephen Gossett. Virtual Reality in Education: Overview and Examples [Electronic resource] // BuiltIN, 2022. – Режим доступу до ресурсу: <https://builtin.com/edtech/virtual-reality-in-education>
3. Nick Babich. How VR In Education Will Change How We Learn And Teach [Electronic resource] // Adobe, 2019-2021. – Режим доступу до ресурсу: <https://xd.adobe.com/ideas/principles/emerging-technology/virtual-reality-will-change-learn-teach/>
4. How Virtual Reality Is Changing [Electronic resource] // LSU Online, 2022. – Режим доступу до ресурсу: <https://online.lsu.edu/newsroom/articles/how-virtual-reality-changing-education/>
5. Body VR [Electronic resource] // Steam, 2022. – Режим доступу до ресурсу: [https://store.steampowered.com/app/451980/The\\_Body\\_VR\\_Journey\\_Inside\\_a\\_Cell/?l=russian](https://store.steampowered.com/app/451980/The_Body_VR_Journey_Inside_a_Cell/?l=russian)
6. Mondly Learn Languages in VR [Electronic resource] // Steam, 2022. – Режим доступу до ресурсу: [https://store.steampowered.com/app/1141930/Mondly\\_Learn\\_Languages\\_in\\_VR/](https://store.steampowered.com/app/1141930/Mondly_Learn_Languages_in_VR/)
7. Системні вимоги для роботи шолому HTC Vive. [Електронний ресурс] // HTC, 2011-2022. – Режим доступу до ресурсу: [https://www.vive.com/ua/support/vive/category\\_howto/what-are-the-system-requirements.html](https://www.vive.com/ua/support/vive/category_howto/what-are-the-system-requirements.html)

8. OpenVR Unity XR Plugin [Electronic resource] // GitHub, 2022. – Режим доступу до ресурсу: <https://github.com/ValveSoftware/unity-xr-plugin>
9. VRTK – Virtual Reality Toolkit [Electronic resource] // VTRK. – Режим доступу до ресурсу: <https://vrtoolkit.readme.io/>
10. Living Birds [Electronic resource] // Unity Technologies, 2022. – Режим доступу до ресурсу: <https://assetstore.unity.com/packages/3d/characters/animals/birds/living-birds-15649>
11. Арестов А. В. Особливості застосування технології віртуальної реальності у навчальному процесі: магістерс. робота / Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського». Київ, 2018. 112 с.
12. Трач Ю. VR – технології як метод і засіб навчання. *Освітологічний дискурс* / Київський університет імені Б. Грінченка. 2017. № 3-4 (18-19). Київ, 2017. с. 309-322.
13. Гончаренко С. А., Кузьмук Д. А., Шаповалов С. П. Додаток доповненої реальності «Augmented reality education». *МА: ІМА – 2020: матеріали та програма науково-технічної конференції СумДУ. 2020. С. 36.*
14. Гончаренко С. А., Кузьмук Д. А., Шаповалов С. П. Впровадження додатків віртуальної реальності в процесі університетського навчання. *Інформатика, математика, автоматика: матеріали науково-технічної конференції. Суми, СумДУ, 2021. С. 65-66.*
15. Гончаренко С. А., Кузьмук Д. А. Застосування технологій віртуальної реальності в навчальному процесі університету: наукова робота, Мелітополь, 2021. 30 с.

## ДОДАТОК А

Файл AccidentController.cs:

```

using UnityEngine;
using Photon.Pun;
using Photon.Realtime;
using ExitGames.Client.Photon;
using UnityEngine.UI;

public class AccidentController : MonoBehaviour, IOnEventCallback
{
    public static AccidentController instance = null;

    public bool isActive = false;
    private double startTime;
    public int time = 180;
    public ControlPanel controlPanel;
    public Text changeAdminText;
    public Text timeText;

    public void OnDisable()
    {
        PhotonNetwork.RemoveCallbackTarget(this);
    }
    void OnEnable()
    {
        PhotonNetwork.AddCallbackTarget(this);
        if (instance == null)
        {
            instance = this;
        }
        else if (instance == this)
        {
            Destroy(gameObject);
        }
    }

    void Update()
    {
        if (isActive)
        {
            timeText.text = (time - (PhotonNetwork.Time - startTime)).ToString();
            if (PhotonNetwork.Time - startTime > time)
            {
                GameObject.FindGameObjectWithTag("BedValves").transform.GetChild(0).gameObject.SetActive(
true);

                GameObject.FindGameObjectWithTag("BedValves").transform.GetChild(1).gameObject.SetActive(
true);

                isActive = false;
            }
        }
    }
}

```

```

        else
        {
            if((controlPanel.valve1_value == 4 && controlPanel.valve2_value == 5))
            {
                StopAvaria();
            }
        }
    }

    public void startAccident()
    {
        if (!isActive)
        {
            isActive = true;
            startTime = PhotonNetwork.Time;
            RaiseEventOptions options = new RaiseEventOptions { Receivers =
ReceiverGroup.Others };
            SendOptions sendOptions = new SendOptions { Reliability = true };
            PhotonNetwork.RaiseEvent(50, null, options, sendOptions);
        }
    }
    public void StopAvaria()
    {
        isActive = false;
        controlPanel.off();
    }

    public void OnEvent(EventData photonEvent)
    {
        if(photonEvent.Code == 50)
        {
            startAccident();
        }
    }
}

```

### Файл ControlPanel.cs:

```

using UnityEngine;

public class ControlPanel : MonoBehaviour
{
    public GameObject arrow;
    public float valve1Value;
    public float valve2Value;
    public GameObject lampNormal;
    public GameObject lampEmergency;
    public ParticleSystem steamFog;
    public GameObject waterSplash;

    void FixedUpdate()
    {
        if
(GameObject.FindGameObjectWithTag("Head").GetComponent<AccidentController>().isActive)

```

```

    {
        if (valve1Value != 4 || valve2Value != 5)
        {
            arrow.GetComponent<Animator>().SetTrigger("emergency");
            arrow.GetComponent<Animator>().ResetTrigger("fix_problem");
            lampEmergency.SetActive(true);
            lampNormal.SetActive(false);
            if (valve1Value != 4)
            {
                steamFog.emissionRate = 200;
            }
            else
            {
                waterSplash.SetActive(true);
            }
        }
        else
        {
            arrow.GetComponent<Animator>().SetTrigger("fix_problem");
            arrow.GetComponent<Animator>().ResetTrigger("emergency");
            lampEmergency.SetActive(false);
            lampNormal.SetActive(true);
            steamFog.emissionRate = 10;
            waterSplash.SetActive(false);
        }
    }
}

public void off()
{
    arrow.GetComponent<Animator>().SetTrigger("fix_problem");
    arrow.GetComponent<Animator>().ResetTrigger("emergency");
    lampEmergency.SetActive(false);
    lampNormal.SetActive(true);
    steamFog.emissionRate = 10;
    waterSplash.SetActive(false);
}
}

```

### Файл Elevator.cs:

```

using UnityEngine;
using Photon.Pun;
using Photon.Realtime;
using ExitGames.Client.Photon;

public class Elevator : MonoBehaviour, IOnEventCallback
{
    public GameObject downPosition;
    public GameObject upperPosition;
    public float speed;

    private bool isMoving = false;

    public bool playerInside;
}

```



```

public GameObject liftCabine;
public GameObject playerPrefab;
public bool liftUp;
public int personInElivator = 0;
private double startMoving;

public float timerLifting;
public AudioClip liftSound;
private AudioSource aus;
private bool audioPlay;
private bool isInLift;

void Start()
{
    aus = GetComponent<AudioSource>();
}

public void OnDisable()
{
    PhotonNetwork.RemoveCallbackTarget(this);
}

void OnEnable()
{
    PhotonNetwork.AddCallbackTarget(this);
}

void FixedUpdate()
{
    if(Vector2.Distance(new Vector2(transform.position.x, transform.position.z), new
Vector2(playerPrefab.transform.GetChild(2).position.x,
playerPrefab.transform.GetChild(2).position.z)) <= 0.78)
    {
        if (!isInLift)
        {
            double[] data = { 1, PhotonNetwork.Time };
            RaiseEventOptions options = new RaiseEventOptions { Receivers =
ReceiverGroup.All };
            SendOptions sendOptions = new SendOptions { Reliability = true };
            PhotonNetwork.RaiseEvent(10, data, options, sendOptions);
            isInLift = true;
        }
    }
    else
    {
        if (isInLift)
        {
            double[] data = { -1, PhotonNetwork.Time };
            RaiseEventOptions options = new RaiseEventOptions { Receivers =
ReceiverGroup.All };
            SendOptions sendOptions = new SendOptions { Reliability = true };
            PhotonNetwork.RaiseEvent(10, data, options, sendOptions);
            isInLift = false;
        }
    }
    if (playerInside)
    {

```

```

        if (PhotonNetwork.Time - startMoving > 3 ) {
            liftUp = true;
            playerInside = false;
        }
    }

    if (liftUp)
    {
        if (!audioPlay)
        {
            aus.PlayOneShot(liftSound);
            audioPlay = true;
        }
        if (isMoving)
        {
            liftCabine.transform.position =
Vector3.MoveTowards(liftCabine.transform.position, downPosition.transform.position,
speed);
            playerPrefab.transform.position =
Vector3.MoveTowards(playerPrefab.transform.position, new
Vector3(playerPrefab.transform.position.x, 0.212f, playerPrefab.transform.position.z),
speed);
        }
        else
        {
            liftCabine.transform.position =
Vector3.MoveTowards(liftCabine.transform.position, upperPosition.transform.position,
speed);
            playerPrefab.transform.position =
Vector3.MoveTowards(playerPrefab.transform.position, new
Vector3(playerPrefab.transform.position.x, 9.3f, playerPrefab.transform.position.z),
speed);
        }
        Debug.Log(Vector3.Distance(liftCabine.transform.position,
upperPosition.transform.position) <= 0.00001 ||
Vector3.Distance(liftCabine.transform.position, downPosition.transform.position) <=
0.00001);
        if (Vector3.Distance (liftCabine.transform.position,
upperPosition.transform.position) <= 0.00001f ||
Vector3.Distance(liftCabine.transform.position, downPosition.transform.position) <=
0.00001)
        {
            liftUp = false;
            isMoving = !isMoving;
        }
    }
}

public void OnEvent(EventData photonEvent)
{
    if (photonEvent.Code == 10)
    {
        double[] data = (double[])photonEvent.CustomData;
        personInElivator += (int)data[0];
        if(personInElivator >= PhotonNetwork.CurrentRoom.PlayerCount)
        {

```

```
        startMoving = data[1];
        playerInside = true;
    }
    else
    {
        playerInside = false;
    }
}
}
```

### Файл FootHorizontal.cs:

```
using UnityEngine;

public class FootHorizontal : MonoBehaviour
{
    public Transform controller;

    void Update()
    {
        transform.eulerAngles = new Vector3(90, controller.eulerAngles.y,
transform.eulerAngles.z);
        transform.position = new Vector3(transform.parent.transform.position.x,
transform.parent.transform.position.y + 0.1f, transform.parent.transform.position.z);
    }
}
```