

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики

(інститут)

Факультет інформаційних технологій

(факультет)

Кафедра Програмного забезпечення комп'ютерних систем

(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня

бакалавра

(назва освітньо-кваліфікаційного рівня)

студента

Бут Анастасії Сергіївни

(ПІБ)

академічної групи

122-20-1

(шифр)

спеціальності

122 Комп'ютерні науки

(код і назва спеціальності)

освітньої програми

Комп'ютерні науки

(назва освітньої програми)

на тему:

Розробка 3D-сцени в Blender для інтеграції

у VR-середовище

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	<i>проф. Мещеряков Л.І.</i>			
розділів:				
спеціальний	<i>проф. Мещеряков Л.І.</i>			
економічний	<i>доц. Касьяненко Л.В.</i>			
Рецензент				
Нормоконтролер	<i>доц. Гуліна І.Г.</i>			

Дніпро
2024

Міністерство освіти і науки України
НТУ «Дніпровська політехніка»

ЗАТВЕРДЖЕНО:

завідувач кафедри
програмного забезпечення комп'ютерних систем
(повна назва)

М.О. Алексєєв
(підпис) (прізвище, ініціали)

« » 2024 року

ЗАВДАННЯ
на кваліфікаційну роботу
бакалавра
(назва освітньо-кваліфікаційного рівня)

студента 122-20-1 Бут Анастасії Сергіївни
(група) (прізвище та ініціали)
тема кваліфікаційної роботи Розробка 3D-сцени в Blender для інтеграції
у VR-середовище

затверджена наказом ректора НТУ «ДП» від 23.05.2024 № 469-с

Розділ	Зміст виконання	Термін виконання
Спеціальний	На основі матеріалів виробничої практики та інших науково-технічних джерел провести аналіз стану рішення проблеми та постановку задачі. Обґрунтувати вибір та здійснити реалізацію методів вирішення проблеми	20.05.2024 р.
Економічний	Провести розрахунок трудомісткості розробки програмного забезпечення, витрат на створення ПЗ й тривалості його розробки	03.06.2024 р.

Завдання видав _____ проф. Мещеряков Л.І.
(підпис) (посада, прізвище, ініціали)
Завдання прийняв до виконання _____ Бут А.С.
(підпис) (прізвище, ініціали)

Дата видачі завдання: 14.01.2024 р.

Термін подання кваліфікаційної роботи до ЕК: 10.06.2024 р.

РЕФЕРАТ

Пояснювальна записка: 107 с., 43 рис., 1 табл., 4 дод., 41 джерело.

Об'єкт розробки: 3D-сцена в Blender для інтеграції у VR-середовище.

Мета кваліфікаційної роботи: розробка 3D-сцени в Blender на основі фізичної моделі острова, розташованої в Ройтлінгенському університеті, та її інтеграції у середовище віртуальної реальності в рамках проєкту «Virtual Reality Campus» для підвищення якості навчання шляхом застосування інноваційних технологій.

У вступі розглядається аналіз та сучасний стан проблеми, конкретизується мета кваліфікаційної роботи та галузь її застосування, наведено обґрунтування актуальності теми та уточнюється постановка завдання.

У першому розділі проаналізовано предметну галузь, визначено актуальність завдання та призначення розробки, сформульовано постановку завдання, зазначено вимоги до програмної реалізації, технологій та програмних засобів.

У другому розділі проаналізовані наявні рішення, обрано середовище для розробки, виконано проектування і розробку 3D-сцени, описана її структура, алгоритми функціонування, процес адаптації для VR-середовища, а також виклик та завантаження додатку, визначено вхідні і вихідні дані, охарактеризовано склад параметрів технічних засобів.

В економічному розділі визначено трудомісткість розробленої 3D-сцени, проведений підрахунок вартості роботи щодо створення проєкту та розраховано час на його реалізацію.

Практичне значення полягає у створенні високоякісної та оптимізованої 3D-сцени енергетичного острова, яка інтегрується у VR-середовище для забезпечення інтерактивного та імерсивного навчального досвіду з вивчення відновлюваної енергетики.

Актуальність розробки 3D-сцени для інтеграції у VR-середовище визначається зростаючим попитом на інноваційні освітні рішення, здатні підвищити ефективність засвоєння навчального матеріалу та розвинути практичні навички студентів інженерних спеціальностей.

Список ключових слів: 3D-МОДЕЛЬ, ВІРТУАЛЬНА РЕАЛЬНІСТЬ, VR, ІМСЕРСІВНИЙ, 3D-МОДЕЛЮВАННЯ, ОБ'ЄКТ, ПОЛІГОНАЛЬНЕ МОДЕЛЮВАННЯ, ЕНЕРГЕТИКА ВІДНОВЛЮВАНИХ ДЖЕРЕЛ, ВІЗУАЛІЗАЦІЯ ЕНЕРГЕТИЧНОЇ СИСТЕМИ, ПРАКТИЧНЕ НАВЧАННЯ, ІННОВАЦІЙНІ ТЕХНОЛОГІЇ, СТУДЕНТ, ОСВІТА, МОДЕЛЬ, BLENDER, WEBXR.

ABSTRACT

Explanatory note: 107 pp., 43 fig., 1 tbl., 4 extra, 41 sources.

The object of development: a 3D scene in Blender for integration into VR environment.

The purpose of the diploma project: to develop a 3D scene in Blender based on the physical model of an island located at Reutlingen University and integrate it into a virtual reality environment within the "Virtual Reality Campus" project to improve the quality of education through the use of innovative technologies.

The introduction considers the analysis and the current state of the problem, specifies the purpose of the qualification work and the scope of its application, substantiates the relevance of the topic and clarifies the formulation of the problem.

In the first chapter, the subject area is analyzed, the relevance of the task and the purpose of development are determined, the statement of the problem is formulated, the requirements for software implementation, technologies and software are indicated.

In the second section, the available solutions are analyzed, a platform for development is selected, the design and development of the 3D scene is carried out, its structure, functioning algorithms, the process of adaptation for VR environment, as well as the call and loading of the application are described, the input and output data are determined, the composition of the parameters of the technical means is characterized.

In the economic section, the labor intensity of the developed 3D-scene is determined, the cost of work on creating the project is calculated and the time for its implementation is estimated.

The practical value lies in creating a high-quality and optimized 3D scene of an energy island, which integrates into the VR environment to provide an interactive and immersive learning experience in the study of renewable energy.

The relevance of developing a 3D scene for integration into a VR environment is determined by the growing demand for innovative educational solutions capable of increasing the effectiveness of learning material assimilation and developing practical skills of engineering students.

List of keywords: 3D MODEL, VIRTUAL REALITY, VR, IMMERSIVE, 3D MODELING, OBJECT, POLYGONAL MODELING, RENEWABLE ENERGY, ENERGY SYSTEM VISUALIZATION, PRACTICAL TRAINING, INNOVATIVE TECHNOLOGIES, STUDENT, EDUCATION, MODEL, BLENDER, WEBXR.

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	3
ВСТУП.....	4
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ.....	6
1.1. Загальні відомості з предметної галузі.....	6
1.1.1. Переваги використання VR для освітніх цілей.....	6
1.1.2. Виклики та обмеження впровадження VR	8
1.1.3. Огляд наявних рішень із застосуванням VR у навчальних процесах	10
1.2 Призначення розробки та галузь застосування	18
1.3 Підстави для розробки	20
1.4. Постановка завдання	20
1.5. Вимоги до програми або програмного виробу	24
1.5.1. Вимоги до функціональних характеристик	24
1.5.2. Вимоги до інформаційної безпеки.....	25
1.5.4. Вимоги до інформаційної та програмної сумісності	28
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	30
2.1. Функціональне призначення системи	30
2.2. Опис застосованих математичних методів.	31
2.3. Опис використаних технологій та мов програмування.....	34
2.3.1 Середовище розробки 3D-сцени	34
2.3.2 Основна мова програмування	36
2.3.3 Використання додаткових бібліотек	38
2.3.4 Доступ до віртуальної реальності через веб-середовище	39
2.3.5 Інструменти для збірки проекту	40

2.3.6 Середовище розробки та інструментарій.....	42
2.4 Опис структури системи та алгоритмів її функціонування	43
2.4.1 Програмна архітектура	43
2.4.2 Логічна структура 3D-сцени	45
2.4.3. Адаптація 3D-моделей і візуалізації даних для VR-середовища	49
2.4.4. Файлова структура	50
2.5. Обґрунтування та організація вхідних та вихідних даних програми .	51
2.5.1. Вхідні дані	51
2.5.2. Вихідні дані	53
2.6. Опис розробленої системи.....	55
2.6.1 Використані технічні засоби	55
2.6.2. Використані програмні засоби	56
2.6.3 Виклик та завантаження програми.	57
2.6.4. Опис інтерфейсу користувача	65
РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ	76
3.1. Розрахунок трудомісткості та вартості розробки програмного продукту	76
3.2. Рахунок витрат на створення програми	81
ВИСНОВКИ	83
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	85
Додаток А. Лістинг програми	89
Додаток Б. Відгук керівника економічного розділу	105
Додаток В. Перелік документів на оптичному носії.....	106
Додаток Г. Авторські права.....	107

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

VR	Virtual Reality (віртуальна реальність);
3D	тривимірний простір;
API	Application Programming Interface (інтерфейс прикладного програмування);
WebGL	Web Graphics Library (веб-бібліотека для роботи з 3D-графікою);
ROI	Return On Investment (рентабельність інвестицій);
ES6	ECMAScript 6, версія стандарту мови програмування JavaScript;
JSON	JavaScript Object Notation, текстовий формат обміну даними;
XML	eXtensible Markup Language (розширювана мова розмітки);
glTF	Transmission Format, відкритий формат файлів для 3D-моделей;
CMD	Command Prompt.
XR	VR + AR

ВСТУП

Стрімкий розвиток інформаційних технологій відкриває нові можливості у сфері освіти. Віртуальна реальність (VR) – одна з новітніх технологій, яка поступово інтегрується у процес навчання. VR-середовища дозволяють створювати імерсивний освітній контент, який занурює студентів у візуалізовану інформацію, забезпечуючи більш ефективне засвоєння матеріалу. Особливої актуальності використання VR набуває для вивчення складних тривимірних об'єктів чи процесів у таких галузях, як освіта, медицина, інженерія, архітектура тощо.

Однак розробка 3D-контенту для віртуальної реальності є досить складним і трудомістким процесом. Для створення VR-сцен потрібні спеціальні навички моделювання та візуалізації. Blender – одне з найпопулярніших вільних програмних рішень для 3D-моделювання, яке дозволяє розробляти реалістичний 3D-контент високої якості.

Зв'язок проблеми, що розглядається у кваліфікаційній роботі, з об'єктом діяльності фахівців напряму «Комп'ютерні науки та інформаційні технології» полягає у розробці програмного забезпечення для візуалізації та інтеграції 3D-контенту у VR-середовища з метою створення інноваційних освітніх рішень.

Метою даної кваліфікаційної роботи є розробка 3D-сцени засобами Blender для її подальшої інтеграції у веб VR-додаток, призначений для використання у навчальному процесі. Галуззю застосування результатів роботи є сфера віртуальної та доповненої реальності в освіті.

Актуальність теми обумовлена кількома чинниками:

- зростаючим попитом на інноваційні імерсивні освітні рішення, здатні підвищити ефективність засвоєння навчального матеріалу;
- необхідністю створення реалістичного 3D-контенту високої якості для VR-середовищ;

– потребою у розробці доступних і зручних інструментів для створення освітнього VR-контенту з використанням вільного програмного забезпечення;

Для досягнення поставленої мети необхідно виконати такі завдання:

– проаналізувати існуючі рішення для розробки 3D-контенту та інтеграції його у VR-середовища в освітніх цілях;

– визначити вимоги до 3D-сцени, яка розробляється, відповідно до заданої тематики та призначення;

– здійснити проектування та розробку 3D-сцени засобами Blender;

– підготувати розроблену 3D-сцену до інтеграції у веб VR-додаток;

– провести тестування та оцінити якість розробленої сцени;

Результатом кваліфікаційної роботи є 3D-сцена, яка після інтеграції у VR-середовище може бути використана у навчальному процесі для імерсивної візуалізації досліджуваних об'єктів та процесів.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Загальні відомості з предметної галузі

1.1.1. Переваги використання VR для освітніх цілей

Віртуальна реальність - це штучно створене комп'ютерне середовище, яке імітує реальний світ або уявні ситуації за допомогою тривимірної візуалізації та спеціальних пристроїв виводу (VR-окулярів, навушників тощо). VR технології дозволяють користувачеві стати частиною цього віртуального світу, взаємодіяти з ним, бачити та чути його елементи так, ніби вони є реальними. Завдяки ефекту занурення та наявності зворотного зв'язку створюється максимально реалістичне відчуття присутності у змодельованому просторі.

Згідно з аналітикою Statista Market Insights прогнозується, що кількість користувачів метавсесвіту у сфері освіти зросте з 28,3 мільйонів (2024 рік) до 104,6 мільйонів користувачів у 2030 році (рис. 1.1) [1]. Це відкриває широкі можливості для впровадження інноваційних методів навчання та підготовки кваліфікованих фахівців.

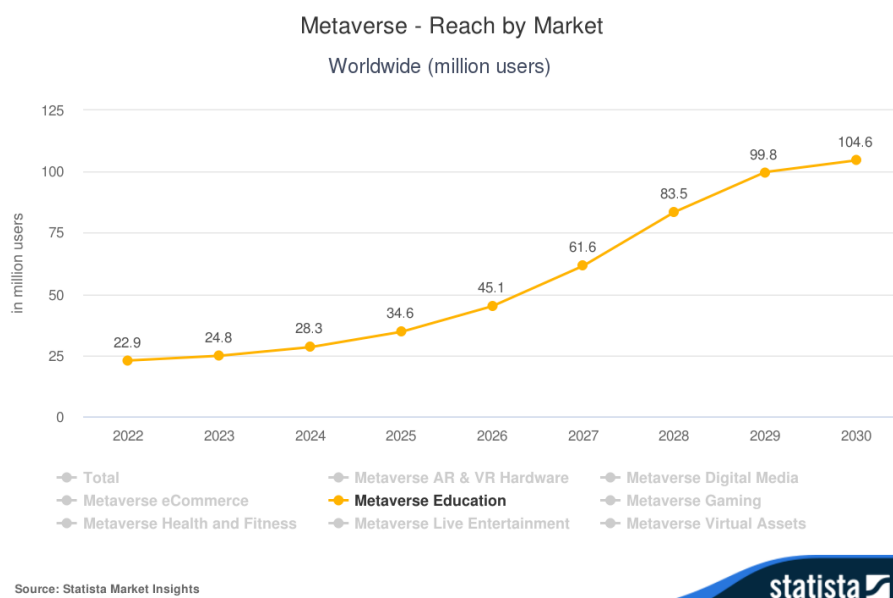


Рис. 1.1. Кількість користувачів метавсесвіту (у мільйонах) у сфері освіти

Потрібно зазначити, що це спричинено рядом переваг використання VR-технологій в освітньому процесі, а саме:

– ефект присутності. Відчуття реального занурення та перебування у віртуальному середовищі. Це досягається завдяки комплексній взаємодії зі штучно змодельованим світом через різні канали сприйняття - зір, слух, дотик тощо. На відміну від пасивного споглядання контенту на екрані чи очно, VR повністю поглинає увагу користувача, забезпечуючи високий рівень концентрації та залучення до навчального процесу. Студенти можуть повністю зануритися у вивчення предмету та позбавлені відволікаючих факторів реального світу.

– реалістична візуалізація та інтерактивність. Ця здатність віртуального середовища надає чудову можливість для інтерактивного, практично-орієнтованого навчання шляхом безпосереднього досвіду. Замість вивчення теоретичних концепцій з книг чи лекцій, студенти можуть вивчати явища, процеси та об'єкти в дії, експериментувати, взаємодіяти з ними, спостерігати з різних ракурсів та в деталях. Більш того, VR полегшує розуміння складних наукових і технічних концепцій через візуалізацію та моделювання процесів, які важко пояснити за допомогою традиційних методів.

– викладання абстрактних, складних для уявлення концепцій. VR відкриває нові творчі можливості візуалізації. Будь-які теоретичні ідеї та поняття можна перетворити на наочні 3D-моделі, з якими можна взаємодіяти, вивчаючи їх різні аспекти. Це особливо корисно у галузях, що охоплюють процеси та явища на мікро- та макрорівнях, такі як квантова фізика, астрономія, молекулярна біологія тощо.

– імітація ситуацій та середовищ, які є недоступними, надзвичайно дорогими або потенційно небезпечними у реальному світі. Наприклад, студенти можуть відвідувати віртуальні моделі промислових об'єктів, таких як електростанції, досліджувати принципи їх функціонування та навіть моделювати аварійні сценарії без жодного ризику для здоров'я чи безпеки. Це

дозволяє набувати практичні навички у сферах, де традиційне навчання обмежене.

– ігровий, розважальний формат. Це є потужним інструментом для підвищення мотивації студентів до навчання та їхнє залучення. Імерсивні середовища надають нові, цікаві способи вивчення матеріалу, що робить навчання більш захоплюючим та наближеним до ігрового процесу [2]. На відміну від традиційних підходів, елементи розваги та захопливий занурювальний досвід можуть зробити сам процес навчання веселим та цікавим, що особливо актуально для нового, цифрового покоління, яке зростає в оточенні технологій та розваг.

– доступність навчання. Наприклад, віртуальні кампуси дають змогу студентам з усього світу отримувати доступ до якісної освіти без необхідності фізичної присутності у конкретному навчальному закладі.

Таким чином, віртуальна реальність виступає не просто технологічною новинкою, а потужним каталізатором змін в освітньому процесі. Вона не лише доповнює традиційні методи навчання, а й відкриває нові горизонти для розвитку інтерактивних, захоплюючих та ефективних навчальних середовищ. Завдяки своїм унікальним можливостям, VR має потенціал зробити освіту більш доступною, персоналізованою та адаптованою до потреб сучасного студента, сприяючи формуванню нового покоління фахівців, готових до викликів майбутнього.

1.1.2. Виклики та обмеження впровадження VR

Попри численні переваги використання віртуальної реальності в освітньому процесі, впровадження цієї технології також має певні виклики та обмеження, які необхідно враховувати, а саме:

– висока вартість відповідного обладнання. Якісні шоломи віртуальної реальності, потужні комп'ютери чи мобільні пристрої для їх підтримки, контролери руху та інші супутні пристрої є доволі дорогими.

Забезпечити таким устаткуванням цілі класи чи групи студентів може бути занадто витратним для багатьох освітніх установ з обмеженими бюджетами.

– вартість створення якісного, змістовного контенту для віртуальної реальності. Розробка реалістичних 3D моделей, інтерактивних середовищ, симуляцій чи навчальних програм вимагає залучення висококваліфікованих фахівців різних профілів - 3D-художників, аніматорів, програмістів, фахівців з предметних галузей тощо. Цей процес потребує значних зусиль та ресурсів.

– шкода здоров'ю. У деяких користувачів можуть виникати симптоми кіберхвороби - нудота, запаморочення, дезорієнтація через конфлікт між зоровими та вестибулярними сигналами [3]. Ефективне використання VR вимагає ретельного вивчення ергономічних аспектів та обмежень щодо часу перебування в середовищі.

– навчання викладачів ефективним методикам інтеграції віртуальної реальності в освітній процес. Без належної підготовки, можливості VR можуть бути нерационально використані. Потрібні зусилля для розробки навчальних програм, кращих практик та рекомендацій щодо створення якісного, педагогічно-обґрунтованого контенту VR.

До того ж, важливо пам'ятати, що хоча технології віртуальної реальності відкривають захопливі нові можливості для навчання, вони не повинні повністю замінити традиційні підходи. Ідеальною є збалансована інтеграція VR як допоміжного навчального інструменту, що доповнює існуючі методики та сприяє кращому засвоєнню матеріалу.

Попри ці потенційні виклики, інновації у VR обладнанні та технологіях дедалі більше знижують бар'єри для їх ширшого впровадження в освіту. Зростаюче усвідомлення переваг занурювального навчання також стимулює збільшення інвестицій та уваги до розвитку цієї галузі. Тож з часом обмеження, пов'язані з вартістю, змістовим наповненням чи іншими аспектами VR будуть все більше долатися.

1.1.3. Огляд наявних рішень із застосуванням VR у навчальних процесах

Віртуальна реальність стрімко змінює підхід до навчання, пропонуючи нові можливості для занурення студентів у навчальний процес та підвищення його ефективності. В останні роки спостерігається значне зростання інтересу до використання VR у навчальних закладах та корпоративному секторі.

Одним із ключових аналізів у цій галузі є дослідження PwC, проведене у 2020 році [4]. Компанія PwC вивчала ефективність використання VR для навчання м'яким навичкам (soft skills) серед своїх співробітників. У дослідженні порівнювалися три методи навчання:

- традиційне очне навчання (classroom);
- дистанційне навчання (E-learn);
- навчання з використанням віртуальної реальності (V-learn).

Результати дослідження виявилися вражаючими. PwC виявила, що V-learn значно перевершує традиційні методи навчання за кількома ключовими показниками:

- швидкість навчання. Спочатку деякі співробітники відчували дискомфорт при використанні гарнітур віртуальної реальності, але з часом більшість знайшла їх зручними та простими у використанні. Навіть з урахуванням додаткового часу, необхідного для ознайомлення нових користувачів з гарнітурами VR (в середньому 10 хвилин), навчання у VR все одно виявилось у чотири рази швидшим, ніж у класі, і в 1,5 рази швидшим, ніж дистанційне навчання, що зазначено на рис. 1.2.

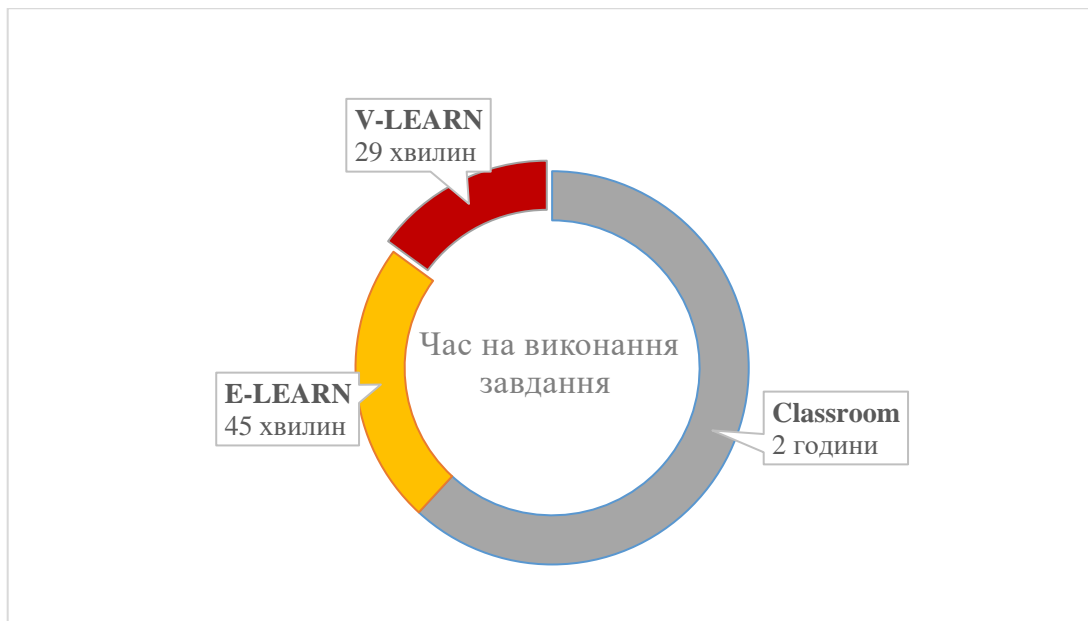


Рис. 1.2. Час на виконання завдання

– упевненість у знаннях. Працівники, які пройшли навчання у віртуальній реальності були на 275% впевненішими у застосуванні отриманих знань після навчання — це на 40% більше, ніж у тих, хто навчався у класі, та на 35% більше, ніж у тих, хто навчався за допомогою дистанційного навчання (рис. 1.3).

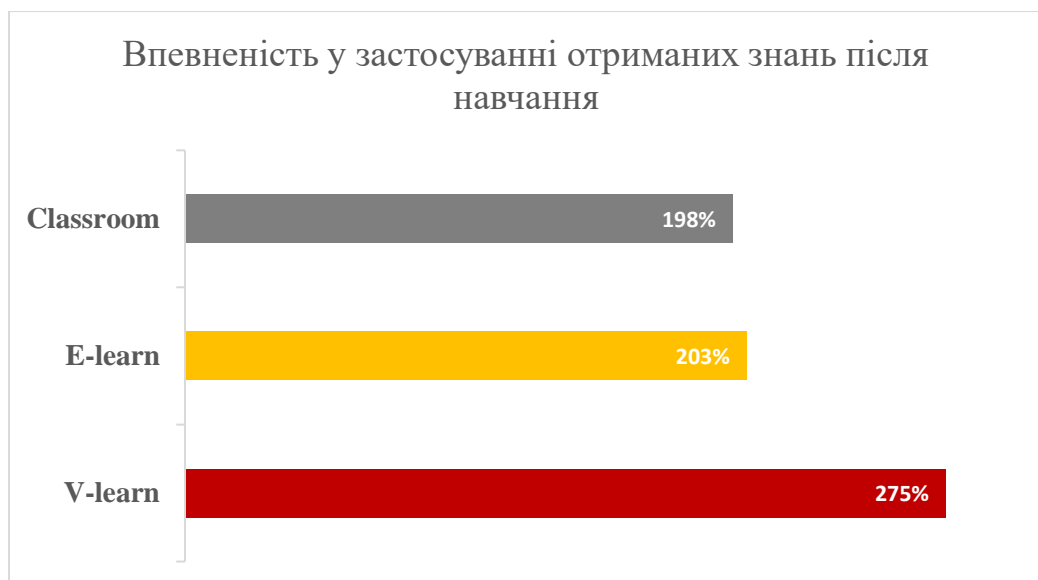


Рис. 1.3. Впевненість у застосуванні отриманих знань після навчання

– емоційний зв'язок із матеріалом. Учасники групи V-learn відчували у 3,75 рази сильніший емоційний зв'язок із матеріалом, ніж ті, хто навчався в очно, та у 2,3 рази сильніший, ніж ті, хто навчався дистанційно (рис. 1.4). Люди глибше пов'язуються з інформацією, краще її розуміють та запам'ятовують, коли залучені їхні емоції [5].



Рис. 1.4. Середній емоційний зв'язок, що відчувається до навчального контенту

– зосередження під час навчання. Дослідження показує, що віртуальна реальність є потужним інструментом для підвищення концентрації уваги користувачів. Завдяки захоплюючому досвіду та симуляціям, VR мінімізує відволікання та утримує зір та увагу користувачів. Наприклад, у ході дослідження виявилось, що слухачі, які навчалися за допомогою VR, були в чотири рази більш зосередженими, ніж ті, хто вивчав електронні матеріали, і в півтора рази більш зосередженими, ніж ті, хто навчався в класі. Це свідчить про те, що занурення учнів у VR-досвід сприяє кращому засвоєнню матеріалу та підвищує ймовірність досягнення кращих результатів, що наведено у таблиці 1.1.

Порівняння концентрації уваги різних досліджуваних груп

	Classroom	E-learn	V-learn
Кількість разів працівники виконували кілька завдань одночасно або відволікалися під час навчання	0,78	1,93	0,48
Кількість хвилин, на їхню думку, знадобилося, щоб повернутися до завдання	1	2,63	0,48

– економність бюджету. Досліджуючи ефективність навчання за допомогою віртуальної реальності, PwC виявила, що, хоча початкові інвестиції у VR-навчання вищі, ніж у традиційних методів, воно може бути більш економічно ефективним у масштабі. Незважаючи на те, що розробка VR-курсу коштує дорожче через необхідність залучення 3D-художників та розробників, таке навчання займає менше часу, що призводить до економії коштів у довгостроковій перспективі, особливо при навчанні великої кількості співробітників. Навіть з урахуванням часу на адаптацію, було підраховано, що паритет витрат VR-навчання в порівнянні з аудиторним можна досягти вже при 375 учнях, що зазначено на рис. 1.5.

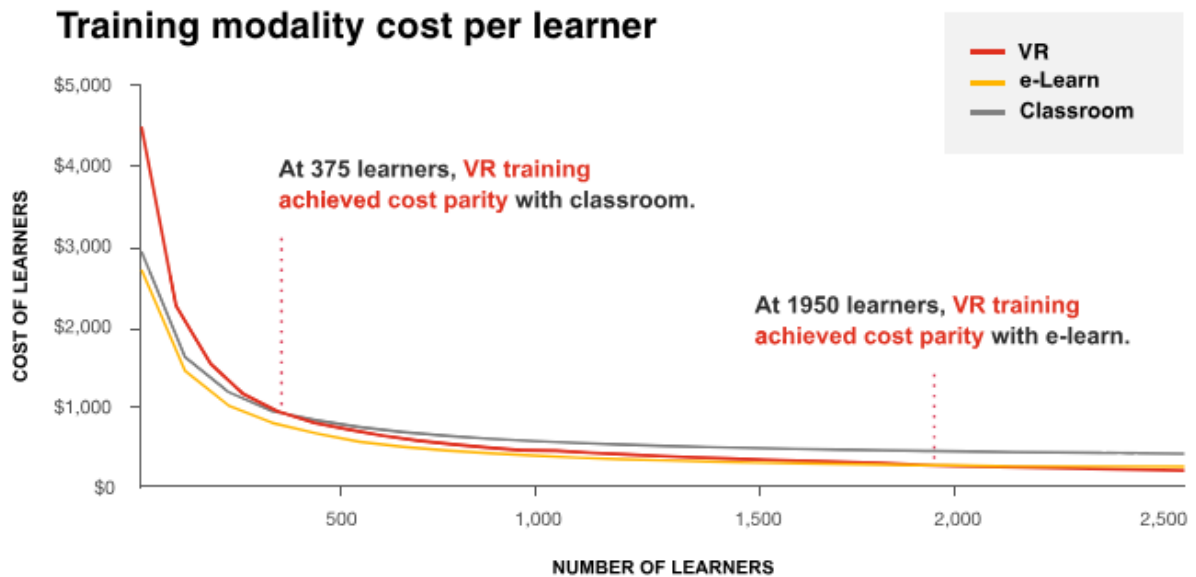


Рис. 1.5. Вартість навчання на одного учня групи V-learn

Також виходячи з цих змінних, було розраховано, що при 3000 учнів вартість VR-навчання падає до 52% менше, ніж аудиторного; при 6000 учнів – на 58% менше, а при 10 000 учнів – на 64% менше. Оскільки час кожного співробітника має цінність, чим більше часу заощаджують працівники, тим швидше компанія може досягти позитивної рентабельності інвестицій (ROI).

Порівняння ROI VR-навчання з електронним навчанням вимагає більшої кількості учнів, ніж аудиторне, оскільки електронне навчання не вимагає залучення викладача/тренера для проведення кожного заняття. Було підраховано, що паритет витрат на VR-навчання в порівнянні з електронним навчанням можна досягти при 1950 учнях. До того ж, було також розраховано, що при 3000 учнів вартість VR-навчання падає на 8% менше; при 6000 учнів – на 20% менше, а при 10 000 учнів – на 26% менше, що ілюстровано на рис. 1.6. Як і у випадку з аудиторним навчанням, чим швидше співробітник може повернутися до роботи, тим швидше можна досягти позитивної рентабельності інвестицій.

52%

more cost-effective than classroom training at 3,000 learners

58%

more cost-effective than classroom training at 6,000 learners

64%

more cost-effective than classroom training at 10,000 learners

Рис. 1.6. Економічність витрат при збільшені кількості учнів

Отже, угруповання з команди інновацій у навчанні та розвитку PwC, групи нових технологій PwC, Oculus for Business та Talespin спільно провели 10-місячний пілотний проект, щоб зрозуміти, чи ефективно використовувати VR для навчання співробітників підприємств навичкам soft skills. У результаті вони дійшли висновку, що навчання за допомогою віртуальної реальності, є ефективнішим методом для навчання в порівнянні з очним навчанням в класі чи дистанційним. Учні групи V-learn, виявляють більшу впевненість, менше відволікаються, мають сильніший емоційний зв'язок з матеріалом. При застосуванні для достатньої кількості учнів, цей метод навчання може бути більш економічно вигідним, ніж навчання в класі або дистанційне.

Вони радять компаніям розглянути мультимодальний підхід у майбутньому, оскільки VR може бути ідеальним для безпечної та динамічної практики набутих знань. Поєднання очного, дистанційного навчання та навчання у віртуальній реальності відкриває шлях до передового підходу в освіті та розвитку співробітників. Такий інноваційний формат має потенціал трансформувати корпоративне навчання, започаткувавши нову еру ефективного та захоплюючого процесу отримання знань [5].

Окрім дослідження PwC, важливим внеском у розвиток VR-технологій в освітньому процесі є проект «A Cloud Enabled Virtual Reality Based Pedagogical Ecosystem for Wind Energy Education», розроблений в університеті Дрекселя [6]. Цей ініціатива демонструє потенціал використання VR у навчанні вузькоспеціалізованих технічних дисциплін, таких як вітроенергетика.

Розроблена в рамках проекту хмарна VR-платформа дозволяє студентам глибше зануритися у вивчення вітроенергетичних систем. Завдяки реалістичному моделюванню, що базується на математичних моделях вітрової турбіни WKA-60, студенти можуть досліджувати її роботу, змінюючи різні параметри, такі як довжина лопатей, кут їх нахилу та швидкість вітру. Це дозволяє не лише візуалізувати роботу вітрової турбіни, але й зрозуміти, як зміна певних параметрів впливає на її ефективність.

Інтерактивність платформи дозволяє студентам самостійно проектувати віртуальні вітрові електростанції, обираючи оптимальне розташування та конфігурацію турбін для максимального вироблення енергії. Такий підхід сприяє розвитку практичних навичок та розуміння принципів проектування вітроенергетичних систем.

Важливою особливістю платформи є надання студентам зворотного зв'язку в режимі реального часу. Приклад опцій налаштування характеристик вітряків зображено на рис. 1.7.

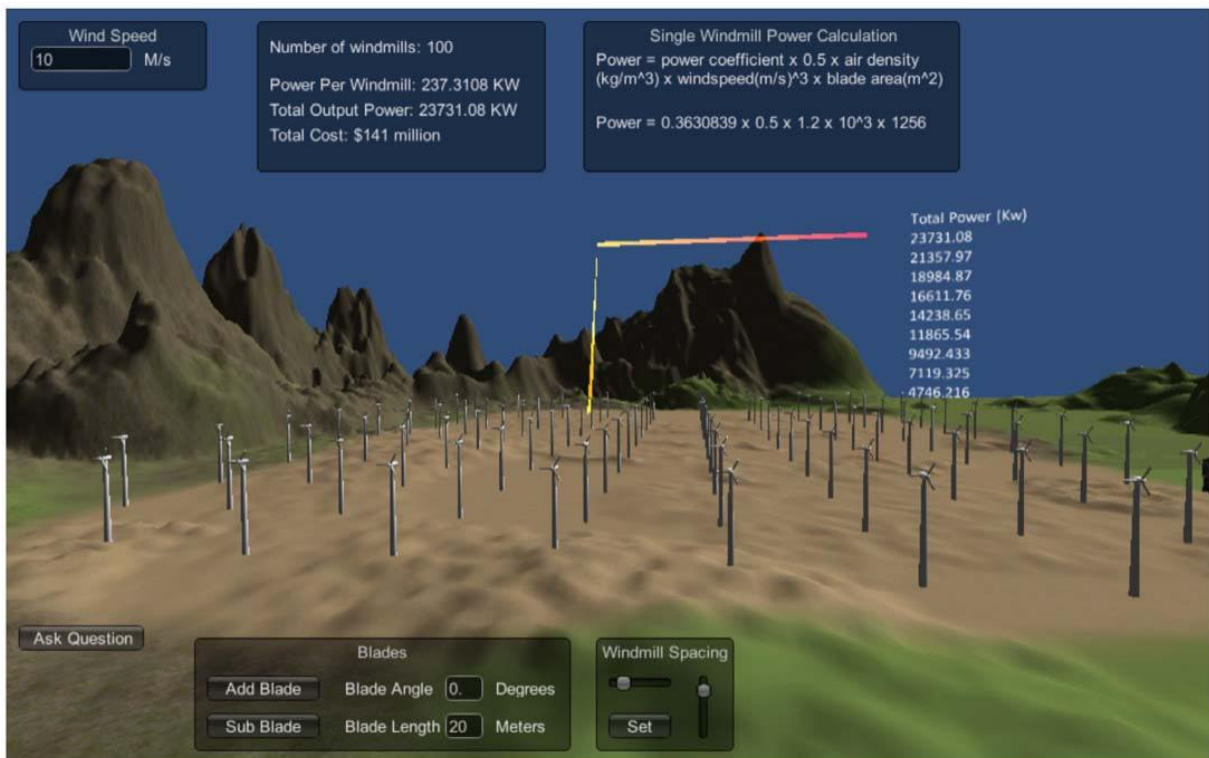


Рис. 1.7. Приклад опцій налаштування характеристик вітряків

Таким чином, студенти можуть бачити, як їхні проектні рішення впливають на загальну потужність вітроелектростанції, що допомагає їм швидко оцінити ефективність різних конфігурацій та вдосконалити свої проекти.

Ще однією перевагою платформи є наявність адаптивної методології оцінювання, яка враховує не лише кінцевий результат, а й процес мислення студента, його дизайнерські та операційні навички [7]. Це дозволяє викладачам отримати повну картину про рівень знань та компетенцій студентів.

Як підсумок, проект «A Cloud Enabled Virtual Reality Based Pedagogical Ecosystem for Wind Energy Education» підкреслює важливість використання реалістичного моделювання, інтерактивності та зворотного зв'язку для створення ефективного навчального середовища у VR-середовищі. Він також демонструє, що технології віртуальної реальності можуть бути успішно застосовані не лише для загальних дисциплін, але й для вузькоспеціалізованих технічних галузей, таких як вітроенергетика, оскільки студенти мають змогу тренуватися у середовищі та обставинах, які неможливо відтворити у реальному житті. Тож це дозволяє набувати практичний досвід у сферах, де традиційні методи навчання мають обмеження.

1.2 Призначення розробки та галузь застосування

У межах цієї кваліфікаційної роботи розглядається розробка 3D-сцени в Blender для інтеграції у VR-середовище з метою використання інноваційних технологій у навчанні, що підвищить продуктивність та зацікавленість студентів.

Причинами виникнення необхідності розробки продукту є:

- покращення навчального процесу можливістю викладачам пояснювати матеріал не лише за допомогою наявних фізичних моделей, а у 3D вигляді в імерсивному середовищі;
- проведення лабораторних робіт та експериментів з різними енергетичними системами;
- моделювання процесів виробництва, розподілу та споживання енергії;
- вивчення принципів роботи обладнання відновлюваних джерел енергії.

Розроблена та інтегрована 3D-сцена у VR повинна в перспективі забезпечувати виконання наступних функцій:

- навчання незалежно від поточного місця знаходження;
- інтерактивна взаємодія з об'єктами моделі (запуск, зупинка, налаштування параметрів);
- відображення статистичних даних про вироблення та споживання енергії;
- імітація різних погодних умов та часу доби для аналізу їх впливу.

Основна термінологія містить загальні терміни з областей інформаційних технологій, комп'ютерної графіки та освіти. Ключові слова містять: 3D-модель, віртуальна реальність, VR, імерсивний, 3D-моделювання, об'єкт, полігональне моделювання, енергетика відновлюваних джерел, візуалізація енергетичної системи, практичне навчання, інноваційні технології, студент, освіта, модель, Blender, WebXR.

Основні терміни, застосовані в цій роботі, необхідні для розуміння основних аспектів елементної структури системи:

- імерсивний – ступень поглинання користувача в змодельоване середовище та створення відчуття присутності та взаємодії з ним;
- острів – розроблена 3D-модель, що включає у себе всі змодельовані об'єкти;
- енергетична система – змодельовані об'єкти, що виділяють енергію або перетворюють її та включають у себе сонячні батареї, вітряки, паливний елемент, сонячний інвертор;
- полігональне моделювання – це підхід для моделювання об'єктів, що ґрунтується на апроксимації поверхонь об'єктів за допомогою множини плоских багатокутників;
- золота спіраль – це пропорція, використана для моделювання форми острова та розташування об'єктів на ньому. Це одна з найбільш досконалих і гармонійних форм у природі і є безкінечною кривою, що рівномірно розширюється зі збереженням постійного кута росту;
- 3D-сцена – це віртуальний простір, де тривимірні об'єкти розміщені та взаємодіють між собою. Цей простір імітує реальний або вигаданий світ, маючи глибину, перспективу та освітлення;
- ambient-шуми - це фонові звуки, які створюють атмосферу певного середовища або обстановки (шум дощу, вітру, хвиль);
- VR-сесія - це інтерактивний сеанс, під час якого користувач занурюється у віртуальну реальність через веб-браузер;
- рендеринг - процес створення віртуального середовища, яке користувач бачить на екрані VR-гарнітури.

Аналізуючи вищезазначені вимоги, було зроблено висновок про те, що в проєкті обов'язково буде реалізовано:

- інтуїтивний користувацький інтерфейс;
- візуалізація 3D графіки з високою якістю;

- оптимізація 3D-моделі для плавного відображення на сайті;
- інтеграція 3D-сцени у VR-середовище.

Розроблене програмне забезпечення може застосовуватись не лише у вищих навчальних закладах технічного спрямування, а й в інженерних компаніях та дослідницьких центрах для навчання персоналу, презентацій та моделювання промислових об'єктів і технологічних процесів.

1.3 Підстави для розробки

Відповідно до ОПП, згідно навчального плану та графіків навчального процесу, в кінці навчання студент виконує кваліфікаційну роботу (проєкт).

Тема роботи узгоджується з керівником проєкту, випускаючою кафедрою, та затверджується наказом ректора.

Отже, підставами для розробки (виконання кваліфікаційної роботи) є:

- освітня програма спеціальності 122 «Комп'ютерні науки»;
- навчальний план та графік навчального процесу;
- наказ ректора Національного технічного університету «Дніпровська політехніка» №469-с від 23.05.2024;
- завдання на кваліфікаційну роботу на тему «Розробка 3D-сцени в Blender для інтеграції у VR-середовище».

1.4. Постановка завдання

Мета кваліфікаційної роботи: розробка 3D-сцени в Blender на основі фізичної моделі острова, розташованої в Ройтлінгенському університеті, та її інтеграції у середовище віртуальної реальності в рамках проєкту «Virtual Reality Campus» для підвищення якості навчання шляхом застосування інноваційних технологій.

Призначення: завдяки віртуальному середовищу студенти зможуть проводити експерименти, працювати з енергетичними системами острова, такими як сонячні панелі, вітряки та теплові насоси тощо, та аналізувати

розподіл енергії. Це забезпечить реалістичне моделювання процесів, що відбуваються у реальному світі, сприятиме глибшому розумінню навчального матеріалу та розвитку практичних навичок у сфері інженерії.

Інформаційний продукт повинен задовольняти наступним основним вимогам:

- можливість користування незалежно від місця знаходження;
- доступність використання з VR-гарнітурою будь-якої потужності обчислення даних;
- зручність використання користувачами завдяки інтуїтивному дизайну;
- реалістичне відображення об'єктів;
- легкість сприйняття інформації;
- підвищення мотивації студентів до навчання;

Технічно-економічна сутність завдання: створення новітнього програмного продукту, який оптимізує витрати університету на облаштування лабораторій, дозволить масштабувати навчання та підвищити його якість завдяки застосуванню передових технологій моделювання та візуалізації.

Технічна сутність полягає у розробці програмного забезпечення для реалізації інтерактивного тривимірного середовища у форматі віртуальної реальності, в якому буде відтворено фізичну модель острова з університету. Це вимагає використання сучасних технологій комп'ютерної графіки, VR-інтерфейсів та веб-технологій для забезпечення зручного доступу користувачів.

З економічної сторони, впровадження такої системи дозволяє значно скоротити витрати на облаштування та обслуговування натурних навчальних лабораторій і макетів в університеті. Віртуальне середовище можна масштабувати та відтворювати необмежену кількість разів без додаткових матеріальних затрат.

Структура моделі острова включає в себе наступні змодельовані найважливіші об'єкти:

- 20 домівок;
- 36 сонячних панелей;
- 12 електромобілей;
- 12 станцій для заряджання електромобілей;
- 3 наводних вітряки;
- 8 кондиціонерів;
- 1 паливний елемент на основі води та вуглекислого газу.

Основним показником, який характеризує елементи та який потрібно відстежувати, є кількість енергії, яку споживають чи виробляють об'єкти. Таким чином, функціональні складові системи повинні працювати на основі цього показнику.

Також варто зазначити, що модель має містити й інші додаткові елементи, такі як: об'єкти ландшафту, рослинність, архітектурні споруди, допоміжні елементи та освітлення.

Призначенням вихідної інформації є візуалізація поточного стану середовища, відображення результатів моделювання та місцезнаходження користувача в VR-просторі, а також забезпечення зворотного зв'язку між системою та користувачем в інтерактивному VR-режимі.

Вхідна інформація для забезпечення повноцінного функціонування симулятора повинна збиратися з двох основних джерел. По-перше, це дані про положення та орієнтацію користувача у реальному фізичному просторі, які надходять безпосередньо з VR-гарнітури та допоміжних датчиків руху. Відстеження місцезнаходження користувача є критично важливим для створення реалістичного відчуття присутності у віртуальному середовищі, адже від цього залежить, під яким кутом і з якої точки буде відображатися 3D-сцена острова.

Іншим ключовим аспектом вхідної інформації є параметри, які відображають актуальний стан віртуального середовища та енергетичної інфраструктури острова. Ці дані охоплюють широкий спектр характеристик, таких як розташування та поточний стан різноманітних об'єктів у 3D-сцені

(житлові будинки, електромобілі, генератори енергії тощо), а також показники функціонування енергетичної системи, зокрема обсяги виробництва та споживання електроенергії, рівень заряду акумуляторних батарей, стабільність енергомережі та інші релевантні параметри.

Щодо організації збору та передачі в обробку вхідної інформації, а також порядку її контролю і коригування, то тут необхідно дотримуватись певних важливих вимог. Вхідні дані мають надходити безперервним потоком у режимі реального часу для забезпечення актуальності і синхронності відображення станів моделі та положення користувача. Затримки або пропуски інформації можуть призвести до розбіжностей між віртуальним і фізичним середовищами.

Умови, при яких припиняється розв'язання завдання автоматизованим способом, включають досягнення заданих параметрів енергоспоживання, виявлення несправностей у віртуальному обладнанні, а також ситуації, коли системі бракує вхідних даних для продовження розрахунків.

Крім того, розв'язання завдання автоматично припиняється при перевищенні допустимих меж навантаження на обчислювальні ресурси системи. При надмірно високих вимогах до оброблення даних і візуалізації, система може досягти критичного рівня завантаженості, що призведе до некоректної роботи або аварійного завершення. Іншим важливим моментом є втрата зв'язку з периферійними пристроями VR, такими як дата-перчатки чи трекери рухів, що унеможливорює повноцінну взаємодію користувача з середовищем.

Зв'язок даної системи з іншими задачами включає розвиток міждисциплінарних підходів у навчанні. Віртуальний кампус може об'єднувати знання з різних галузей, таких як фізика, інформатика, електротехніка та екологія, що сприяє більш комплексному розумінню студентами складних взаємозв'язків у сучасному світі. Таким чином, розробка та впровадження VR-системи не лише підвищує якість освіти, але й відкриває нові можливості для досліджень та інновацій в університеті.

Розподіл функцій між персоналом і технічними засобами реалізується наступним чином:

- викладачі відповідають за розробку сценаріїв і завдань для симуляцій, контроль успішності студентів;
- адміністратори системи здійснюють налаштування, моніторинг і технічну підтримку VR-симулятора;
- студенти та викладачі використовують VR-обладнання та інтерфейс симулятора для виконання завдань і експериментів.
- програмне забезпечення відповідає за обробку вхідних даних, розрахунки, візуалізацію 3D-середовища і відображення результатів моделювання;
- VR-гарнітури і пов'язане обладнання забезпечують реалістичний занурювальний досвід для користувачів;

Залежно від конкретної ситуації і режиму роботи, співвідношення функцій між персоналом і технічними засобами може змінюватись.

1.5. Вимоги до програми або програмного виробу

1.5.1. Вимоги до функціональних характеристик

Система з інтегрованою 3D-моделлю енергетичного острова повинна забезпечувати наступні функціональні характеристики:

- включати тривимірну модель острова з усіма необхідними елементами енергетичної інфраструктури - будівлями, електромобілями, сонячними панелями, вітряками, тепловими насосами та паливним елементом. Модель має максимально імітувати реальний фізичний острів для забезпечення високого рівня реалізму.
- користувачі (студенти) повинні мати змогу не лише спостерігати за змодельованим середовищем, а й безпосередньо взаємодіяти з його об'єктами та системами. Це включає можливість переміщення у середовищі.
- у майбутньому імітувати в віртуальному середовищі різноманітні ситуації та сценарії, такі сезонні зміни погодних умов, зміна навантаження на

енергосистему тощо. Це дозволить студентам досліджувати поведінку системи в різних режимах роботи.

- реалізувати простий та інтуїтивний інтерфейс взаємодії з віртуальним світом за допомогою контролерів руху VR-гарнітури. Важлива також наявність інформаційних панелей та підказок.

- система повинна бути сумісною з основними брендами VR-гарнітур (Oculus, HTC Vive, Valve Index тощо) та забезпечувати оптимальні параметри продуктивності та візуалізації імерсивного досвіду.

- для зручного доступу планується інтеграція VR-застосунку із веб-інтерфейсом за допомогою WebXR API та бібліотеки Three.js, що дозволить запускати його безпосередньо з веб-браузера на різних пристроях.

Реалізація цих ключових функціональних вимог забезпечить створення потужного навчального інструменту для вивчення відновлюваної енергетики шляхом практичного досвіду в реалістичному віртуальному середовищі.

1.5.2. Вимоги до інформаційної безпеки

Забезпечення належного рівня інформаційної безпеки є критично важливим аспектом для коректного та надійного функціонування системи віртуального навчального середовища для вивчення енергетичних систем.

Для забезпечення безперервного функціонування необхідно регулярно створювати резервні копії всіх критичних даних системи - 3D моделей, параметрів конфігурацій, зібраних наборів даних тощо. У разі збоїв або аварійних ситуацій, система повинна мати можливість швидкого відновлення робочого стану із резервної копії. Це реалізується за допомогою використання системи керування версіями GitLab.

Забезпечення надійного рівня інформаційної безпеки є критично важливим аспектом для коректного та безпечного функціонування системи віртуального навчального середовища з вивчення енергетичних систем. При

розробці системи необхідно врахувати вимоги стандарту WebXR, який містить вбудовані механізми захисту користувачів [7].

WebXR розроблений таким чином, щоб не підготувати користувача до початку VR-сесії та забезпечити легкий вихід з неї. Безпека починається на рівні сервера - весь контент WebXR вимагає використання протоколу HTTPS, що запобігає внесенню шкідливого XR-контенту через атаки.

Якщо досвід WebXR вбудований у веб-сторінку за допомогою IFrame, сторінка, яка містить IFrame, повинна включати атрибут HTML «allow=xr-spatial-tracking» у тег IFrame. Ця вимога запобігає запуску VR-досвіду стороннім кодом, наприклад, рекламою, без дозволу розробника.

Усі сеанси WebXR вимагають взаємодії користувача для запиту початку сесії. Веб-сайт повинен мати кнопку або інший елемент, де користувач робить спеціальну дію для початку сеансу XR. Таким чином неможливо просто перемістити користувачів у XR без їхньої згоди. Це запобігає несподіваному та дезорієнтуючому запуску сеансів для користувачів. У visionOS Safari запитає користувача, чи слід дозволити сайту запускати іммерсивний досвід.

Сеанс WebXR також може запитати дозвіл на відстеження рук користувача системою. У такому випадку буде додатковий запит на згоду користувача. Важливо, щоб користувачі були поінформовані та чітко розуміли, на що вони дають згоду.

Стандарт WebXR вимагає, щоб системна взаємодія була зарезервована для виходу з сеансу WebXR. Ця вимога гарантує, що у користувачів є спосіб швидко залишити сеанс, якщо він викликає у них дискомфорт. Можна легко вийти з іммерсивного досвіду в будь-який час, натиснувши кнопку або використовуючи жест для переходу до домашнього вигляду. Повинен надаватися інтерфейс, щоб полегшити вихід.

На кожному рівні WebXR, від мережевого з'єднання до тривалості сеансу та завершення сеансу, безпека та конфіденційність користувачів є пріоритетом.

Дотримання цих вимог щодо інформаційної безпеки дозволить створити надійну та стабільну систему віртуального навчального середовища для вивчення енергетичних систем, захищену від внутрішніх та зовнішніх загроз.

1.5.3. Вимоги до складу та параметрів технічних засобів

Для забезпечення коректної роботи та максимальної продуктивності системи віртуального навчального середовища для вивчення енергетичних систем, необхідно дотримуватись певних вимог щодо складу та параметрів використовуваних технічних засобів.

Найважливішим компонентом є пристрій віртуальної реальності - шолом або окуляри із вбудованими дисплеями та системою трекінгу руху. Рекомендовані параметри:

- роздільна здатність дисплеїв не менше 2К на око (2160x1200 пікселів);
- частота оновлення не менше 60 Гц для коректної візуалізації;
- широкий діапазон спостереження (не менше 110 градусів);
- вбудована система відстежування положення та руху голови;
- сумісність з контролерами руху для інтерактивного керування;

Для відтворення тривимірної графіки та коректного моделювання складних фізичних процесів потрібен потужний комп'ютер, з мінімальними вимогами:

- багатоядерний процесор з тактовою частотою не менше 3 ГГц;
- відеокарта з підтримкою DirectX 11 та не менше 4 ГБ відеопам'яті;
- оперативна пам'ять не менше 8 ГБ DDR4;
- швидкісний SSD-накопичувач місткістю від 512 ГБ;
- операційна система: Unix, Linux, Microsoft Windows XP/7/8/10/11;
- веб-браузер: Microsoft Internet Explorer, Mozilla Firefox, Opera, Google Chrome;
- підключення до мережі Інтернет;

Для забезпечення зручного інтуїтивного введення та точного відстеження рухів користувача система потребує контролерів, сумісних з обраною VR гарнітурою. Наприклад, Oculus Touch, Valve Index Controllers тощо.

Для створення повноцінного занурювального аудіовізуального середовища необхідно інтегрувати систему просторового аудіовиведення. Наприклад, підключення навушників або акустичних систем з підтримкою об'ємного звуку.

Дотримання цих технічних вимог є необхідною умовою для створення високоякісної, продуктивної та імерсивної системи віртуального навчального середовища для вивчення енергетичних систем.

1.5.4. Вимоги до інформаційної та програмної сумісності

Одним з ключових аспектів інформаційної та програмної сумісності є забезпечення належної продуктивності системи під час роботи з ресурсомісткими інструментами, такими як середовище 3D-моделювання Blender та інтегроване середовище розробки Visual Studio Code.

Для зручної роботи з Blender рекомендується використовувати потужний настільний комп'ютер або ноутбук з наступними мінімальними характеристиками:

- багатоядерний процесор з продуктивністю не нижче рівня Intel Core i5 або AMD Ryzen 5;
- оперативна пам'ять об'ємом від 8 ГБ;
- відеокарта з підтримкою апаратного прискорення 3D-графіки та 2-4 ГБ відеопам'яті;
- твердотілий накопичувач (SSD) для зменшення часу завантаження проектів;

Для роботи з Visual Studio Code достатньо більш скромних обчислювальних ресурсів, однак бажано мати сучасний процесор та об'єм оперативної пам'яті не менше 4 ГБ.

Щодо програмного забезпечення, то важливо забезпечити сумісність використаних бібліотек та фреймворків з актуальними версіями браузерів та VR-гарнітур. Зокрема, Three.js та WebXR API мають бути сумісними з Chrome, Firefox, Edge та іншими провідними браузерами. Для роботи з VR-гарнітурами необхідно використовувати останні стабільні версії драйверів відповідних виробників.

Крім того, необхідно забезпечити наявність та сумісність з найновішими версіями Node.js, npm та інших допоміжних утиліт, які потрібні для збірки, тестування та розгортання VR-застосунку.

Дотримання наведених вимог до апаратної та програмної конфігурації дозволить забезпечити стабільну та продуктивну роботу інструментів розробки, а також коректне функціонування кінцевого VR-застосунку.

РОЗДІЛ 2

ПРОЕКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Функціональне призначення системи

Розроблена 3D модель по мотивам фізичної моделі острова, що знаходиться у Ройтлінгенському університеті, та інтегрована у середовище віртуальної реальності призначена для покращення освітнього процесу вищого закладу. Отже, функціональне призначення продукту включає наступне:

- створення інтерактивного віртуального середовища, що імітує реальний острів з енергетичною системою та об'єктами для проведення лабораторних робіт, експериментів та практичних занять студентів інженерних спеціальностей;

- надання користувачам можливості вивчати принципи роботи різних видів енергетичного обладнання: сонячних панелей, вітряків, теплових насосів, паливних елементів та іншого, спостерігаючи їх функціонування у віртуальному світі;

- симуляція процесів генерування, розподілу та споживання енергії на острові, що дозволяє досліджувати баланс між виробництвом та затратами енергії, вплив зовнішніх факторів тощо;

- забезпечення можливості налаштування параметрів енергетичних систем та проведення різноманітних сценаріїв для вивчення наслідків змін;

- візуалізація статистичних даних щодо енергоспоживання, вироблення та накопичення енергії для полегшення аналізу;

Експлуатаційне ж призначення системи надає наступні переваги для навчального закладу:

- модернізація навчального процесу та підвищення його практичної складової шляхом застосування новітніх технологій віртуальної реальності;

- забезпечення можливості проводити лабораторні роботи з енергетичними системами без необхідності облаштування фізичного острова чи окремих енергетичних об'єктів;
- заощадження ресурсів на облаштування та обслуговування фізичних навчальних лабораторій;
- розширення можливостей для вивчення та аналізу реальних виробничих процесів у безпечному віртуальному середовищі;
- підвищення мотивації студентів до навчання за рахунок використання інноваційних інтерактивних методів та технологій;

Таким чином, створена система із застосуванням віртуальної реальності значно розширює можливості для якісної підготовки майбутніх інженерів в області енергетики.

2.2. Опис застосованих математичних методів.

У процесі розробки проєкту «Virtual Reality Campus» було застосовано математичний метод золотого перетину (золотої пропорції) для досягнення гармонійного та естетично привабливого дизайну острова та його урбаністичного середовища. Золотий перетин, що представляє собою ірраціональне число приблизно рівне 1.618, відомий своєю поширеністю в природі та мистецтві, і вважається привабливим для людського ока. Також він тісно пов'язаний з послідовністю чисел Фібоначчі, де кожне наступне число є сумою двох попередніх (0, 1, 1, 2, 3, 5, 8, 13, 21 і т.д.). Відношення між сусідніми числами Фібоначчі наближається до золотого перетину зі збільшенням їх порядку. Цей зв'язок робить золотий перетин ще більш універсальним інструментом для створення гармонійних композицій.

Форма острова була змодельована з використанням золотого спіралі (рис. 2.1), яка є геометричною інтерпретацією золотого перетину та чисел Фібоначчі. Золота спіраль формується шляхом послідовного додавання квадратів, сторони яких співвідносяться як числа Фібоначчі. Така форма острова створює відчуття

природності та гармонії, оскільки повторює форми, які часто зустрічаються в природному середовищі, наприклад, у раковинах молюсків або розташуванні насіння в соняшнику.

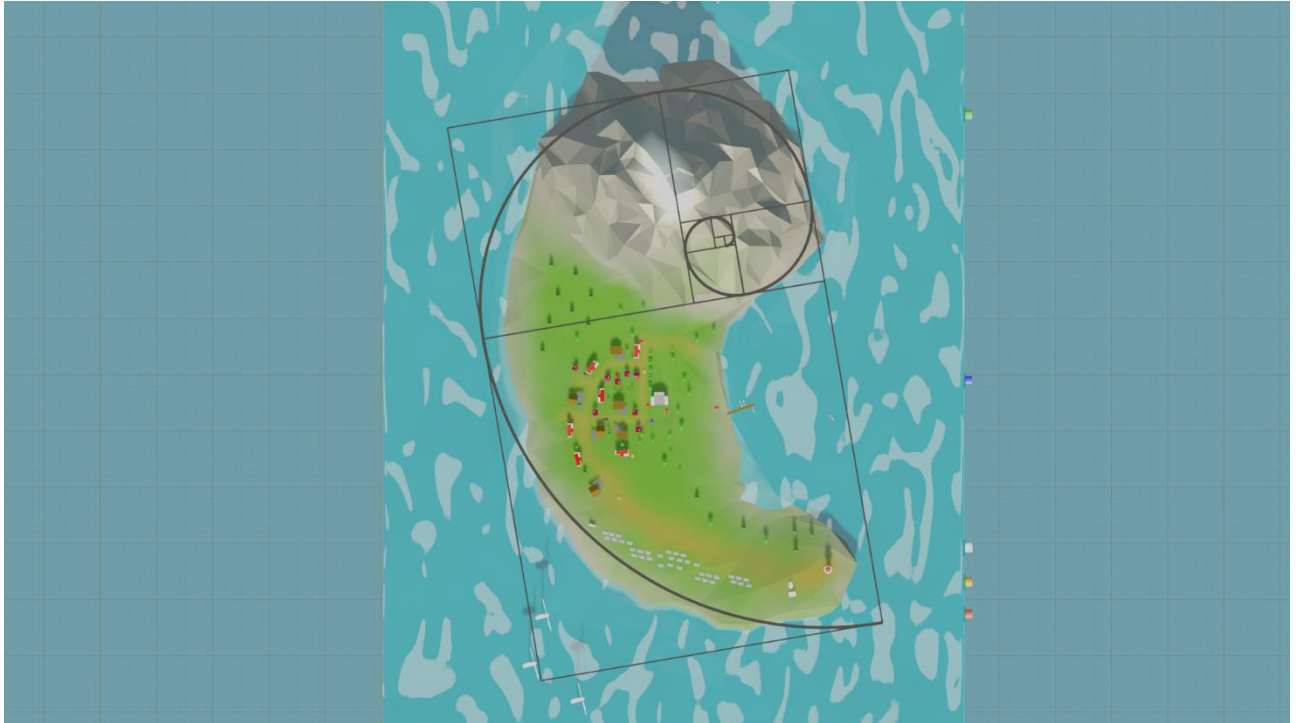


Рис. 2.1. Моделювання форми острову згідно пропорціям золоті спіралі

При плануванні розташування будівель, доріг, зелених зон та інших елементів на острові також використовувався принцип золоті спіралі. Будівлі були розміщені вздовж ліній спіралі, а їх розміри та пропорції також відповідали золотій пропорції, що зображено на рис. 2.2. Використання такого методу дозволило створити візуально збалансований та привабливий урбаністичний ландшафт, який сприяє позитивному сприйняттю користувачами віртуального середовища.



Рис. 2.2. Планування урбаністичної структури згідно пропорціям золоті спіралі та числам Фібоначчі

Використання пропорції золотого перетину та чисел Фібоначчі в моделюванні острова та плануванні коректної урбаністичної інфраструктури базується на дослідженнях, які показують, що об'єкти та композиції, побудовані з використанням цієї пропорції, сприймаються людьми як більш привабливі та гармонійні [8]. Застосування цього математичного методу дозволило створити віртуальний острів, який не лише функціональний, але й естетично привабливий, що сприяє позитивному досвіду користувачів та підвищує їхню зацікавленість у навчальному процесі. Наприклад, дослідження проведене у стінах університету Копенгагену показало, що використання шоломів віртуальної реальності з високою якістю графіки, приємним дизайном та врахуванням індивідуальних особливостей користувачів може покращити сприйняття та засвоєння інформації [9].

2.3. Опис використаних технологій та мов програмування

У процесі розробки віртуального університетського кампусу було застосовано комплексний підхід до вибору технологій та мов програмування. Цей вибір ґрунтувався на необхідності створення ефективного, масштабованого та інтерактивного освітнього середовища, яке б відповідало сучасним стандартам веб-розробки та віртуальної реальності.

Використання цих технологій дозволяє створити інноваційну освітню платформу, яка відповідає сучасним вимогам до інженерної освіти. Вона надає студентам можливість вивчати складні енергетичні системи в інтерактивному, імерсивному середовищі, що сприяє глибшому розумінню матеріалу та розвитку практичних навичок.

Важливо відзначити, що обрані технології не лише забезпечують технічну реалізацію проекту, але й створюють основу для подальшого розвитку та масштабування освітньої платформи. Вони дозволяють легко додавати нові модулі, розширювати функціональність та адаптувати систему до нових освітніх потреб.

2.3.1 Середовище розробки 3D-сцени

Blender - це потужний відкритий пакет для створення 3D-контенту. Згідно зі статистикою Google Trends за останні 5 років (рис. 2.3), можна помітити, що інтерес до програмного забезпечення Blender значно перевищує інтерес відносно інших популярних програм 3D-моделювання, таких як Autodesk 3ds Max, Autodesk Maya, SketchUp та Tinkercad [10]. Середній показник пошукових запитів для Blender майже вдвічі перевищує показники найближчого конкурента - SketchUp. Це свідчить про зростаючу популярність Blender серед професіоналів та ентузіастів 3D-графіки.

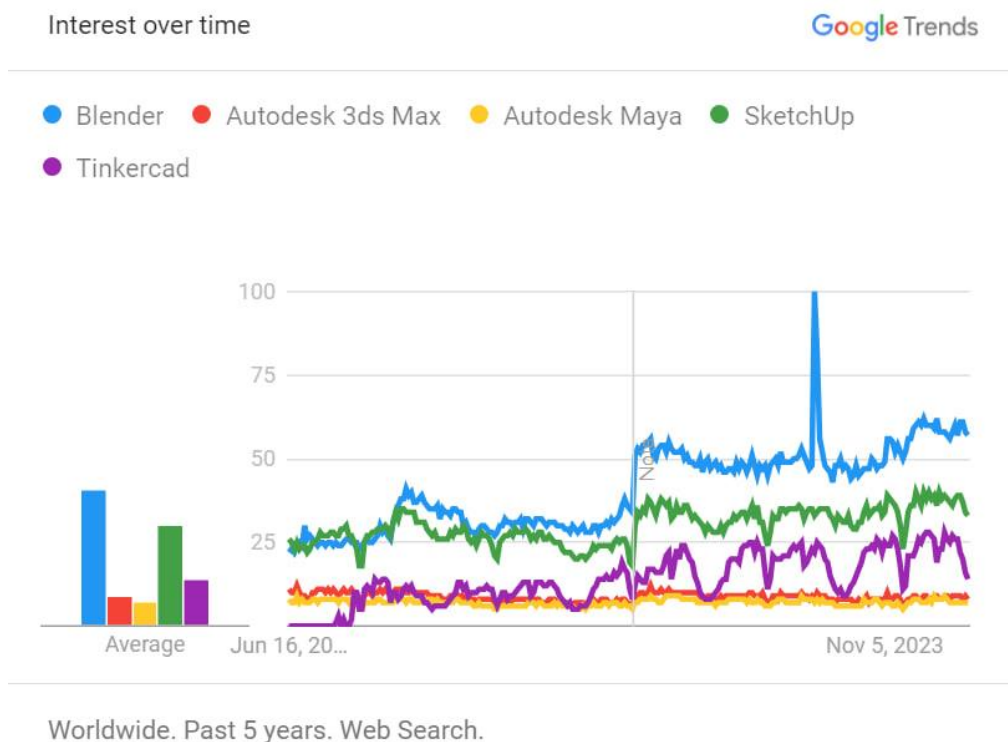


Рис. 2.3. Статистика популярності пошукових запитів «Blender», «Autodesk 3ds Max», «Autodesk Maya», «SketchUp», «Tinkercad»

Важливо відзначити різкий пік інтересу до Blender у середині 2020 року, що може бути пов'язано з випуском версії Blender 2.83 LTS (Long Term Support) у червні 2020 року. Ця версія принесла значні покращення у стабільності та продуктивності, що могло привернути увагу багатьох користувачів [11].

Цікаво, що цей тренд підтверджується і практичним досвідом викладачів. Згідно з інформацією від CG Cookie [12], викладач, який повернувся до використання Blender після чотирирічної перерви, був приємно здивований значними покращеннями у програмі. Він відзначив, що інтерфейс став більш інтуїтивним, а інструменти - потужнішими та зручнішими у використанні. Це свідчить про постійний розвиток Blender та його адаптацію до потреб користувачів.

Крім того, CG Cookie наводить кілька важливих аргументів на користь використання Blender в освітньому процесі:

- доступність. Blender є безкоштовним програмним забезпеченням, що робить його ідеальним вибором для освітніх установ з обмеженим бюджетом.

- крос-платформність. Blender працює на Windows, Mac та Linux, що забезпечує гнучкість у виборі операційної системи.

- всебічність. Blender пропонує інструменти не лише для 3D-моделювання, але й для текстурування, анімації, симуляції, рендерингу, композитингу та навіть відеоредагування, що робить його універсальним інструментом для різноманітних творчих завдань.

- постійне оновлення. Команда розробників Blender регулярно випускає оновлення, додаючи нові функції та покращуючи існуючі, що дозволяє студентам працювати з найсучаснішими інструментами.

- велика спільнота. Blender має активну спільноту користувачів та розробників, які створюють навчальні матеріали, плагіни та розширення, що збагачує освітній досвід.

- підготовка до професійної кар'єри. Зростаюча популярність Blender у професійній сфері означає, що студенти, які опановують цей інструмент, будуть краще підготовлені до майбутньої кар'єри.

Ці фактори, разом із загальним трендом зростання популярності Blender, роблять його оптимальним вибором для проєкту віртуального університетського кампусу. Використання Blender не лише забезпечує нас потужним інструментом для створення 3D-моделей, але й відповідає сучасним тенденціям у сфері комп'ютерної графіки та освіти.

2.3.2 Основна мова програмування

JavaScript є основною мовою програмування проєкту. Згідно з дослідженням Stack Overflow Developer Survey 2023, JavaScript залишається найпопулярнішою мовою програмування серед розробників восьмий рік

поспіль [13]. Це забезпечує широку базу розробників та ресурсів для подальшого розвитку проекту.

Вибір цієї мови програмування обумовлений рядом факторів, які роблять її оптимальною для реалізації складних веб-базованих 3D-середовищ та інтерактивних освітніх платформ:

- універсальність та крос-платформність. JavaScript є єдиною мовою програмування, яка нативно підтримується всіма сучасними веб-браузерами. Це означає, що віртуальний кампус буде доступний на будь-якому пристрої з веб-браузером без необхідності встановлення додаткового програмного забезпечення. Згідно з дослідженням W3Techs, JavaScript використовується на 98.3% всіх веб-сайтів [14], що підкреслює його універсальність та широке поширення.

- потужність для 3D-графіки та VR. З появою WebGL та бібліотек, таких як Three.js, JavaScript став потужним інструментом для створення складної 3D-графіки в браузері. Це критично важливо для розробки проекту, оскільки дозволяє реалізувати детальне та інтерактивне 3D-середовище віртуального кампусу. Дослідження, проведене Peіною у 2020 році, показало, що веб-базовані 3D-середовища, створені за допомогою JavaScript та WebGL, демонструють продуктивність, порівнянну з нативними додатками [15].

- підтримка WebXR. JavaScript є ключовою мовою для роботи з WebXR API, що дозволяє створювати VR-досвід безпосередньо у веб-браузері. Це дає можливість студентам взаємодіяти з віртуальним кампусом через VR-гарнітури без необхідності встановлення спеціального програмного забезпечення. За даними звіту Statista, ринок VR-технологій очікує значне зростання, і веб-базовані VR-рішення гратимуть важливу роль у цьому розвитку [16].

- асинхронність та продуктивність. JavaScript підтримує асинхронне програмування, що критично важливо для створення плавного та відзивчивого користувацького інтерфейсу у складних веб-додатках. Це дозволяє системі

ефективно обробляти великі обсяги даних та складні обчислення без блокування інтерфейсу користувача.

– гнучкість та масштабованість. JavaScript дозволяє легко масштабувати проєкт, додаючи нові функції та компоненти. Це особливо важливо для освітньої платформи, яка може розвиватися та адаптуватися до нових вимог. Модульна структура сучасних JavaScript-додатків, яка використовується у проєкті, сприяє легкому розширенню функціоналу та підтримці коду.

Таким чином, використання мови JavaScript у розробці проєкту «Virtual Reality Campus» є обґрунтованим вибором, який забезпечує потужність, гнучкість та масштабованість, необхідні для створення складного інтерактивного 3D-середовища в веб-браузері. Ця мова програмування надає всі необхідні інструменти для реалізації візуально привабливого, продуктивного та доступного освітнього простору, що відповідає сучасним стандартам веб-розробки та вимогам користувачів.

2.3.3 Використання додаткових бібліотек

Three.js є потужною JavaScript бібліотекою для створення та відображення 3D графіки у веб-браузері [17]. Однією з ключових переваг Three.js є його здатність абстрагувати складність WebGL, надаючи високорівневий API для роботи з 3D-об'єктами, освітленням, текстурами та анімаціями. Це дозволяє зосередитися на створенні детальної та реалістичної моделі віртуального кампусу, не витрачаючи зайвого часу на низькорівневі аспекти рендерингу. У контексті розробленого проєкту, Three.js має ряд важливих переваг:

– якість 3D-моделей. Three.js дозволяє використовувати високоякісні 3D-моделі, які можуть точно відтворювати реальні об'єкти та процеси. Це особливо важливо для 3D-сцени острову, де необхідно детально представити будинки, електромобілі, сонячні панелі та інші елементи інфраструктури.

Реалістичність візуалізації сприяє кращому розумінню студентами принципів роботи енергетичних систем.

- оптимізованість та кросбраузерність. Завдяки цьому віртуальний кампус доступний широкому колу користувачів без необхідності встановлення додаткового програмного забезпечення. Крім того, Three.js забезпечує високу продуктивність навіть на мобільних пристроях, що розширює можливості доступу до освітніх ресурсів.

- потужна система освітлення та тіней. Це дозволяє реалістично симулювати зміну освітлення протягом дня, що є ключовим фактором для демонстрації роботи сонячних панелей та загальної енергетичної динаміки острова. Дослідження проведене у рамках університету Барселони [18] показало, що реалістичне освітлення значно підвищує відчуття присутності у віртуальному середовищі, що, в свою чергу, позитивно впливає на ефективність навчання.

Таким чином, використання Three.js у проєкті моєї кваліфікаційної роботи дозволяє створити високоякісне, інтерактивне та реалістичне 3D середовище, яке служить основою для імерсивного навчального досвіду.

2.3.4 Доступ до віртуальної реальності через веб-середовище

WebXR API – це стандартизований інтерфейс для доступу до пристроїв віртуальної та доповненої реальності у веб-середовищі [19]. Його використання має ряд суттєвих переваг, які значно розширюють освітній потенціал віртуального кампусу:

- високий рівень імерсивності. На відміну від звичайного перегляду 3D-моделі на екрані, VR дозволяє студентам повністю зануритися у віртуальне середовище. Вони можуть «ходити» по території, детально розглядати обладнання, взаємодіяти з різними елементами системи. Це створює відчуття присутності, яке, згідно з дослідженням Журналу комп'ютерного навчання у

2019 році [20], значно підвищує мотивацію студентів та покращує запам'ятовування матеріалу.

- доступ до контролерів VR. Це дозволяє реалізувати природну та інтуїтивно зрозумілу взаємодію з віртуальними об'єктами. Студенти можуть «брати» предмети руками, натискати на віртуальні кнопки, регулювати параметри обладнання – все це сприяє розвитку практичних навичок та глибшому розумінню принципів роботи енергетичних систем.

- адаптивність. Важливою особливістю WebXR є його здатність адаптуватися до різних типів VR-пристроїв – від простих cardboard-окулярів до повноцінних VR-шоломів. Це робить освітній проєкт доступним для широкої аудиторії, незалежно від наявного обладнання.

- реалізація колаборативних сценаріїв навчання. API дозволяє синхронізувати дії кількох користувачів у віртуальному просторі, що дає змогу студентам працювати над спільними проєктами, проводити групові експерименти та обмінюватися досвідом безпосередньо у віртуальному середовищі.

Отже, використання WebXR в поєднанні з іншими технологіями, використаними під час розробки проєкту «Virtual Reality Campus», дозволяє створити метавсесвіт з можливістю вивчати матеріал на моделі острова з побудованою енергетичною системою. Це відкриває широкі можливості для експериментування та аналізу різних сценаріїв роботи енергетичної системи, які було б складно або небезпечно реалізувати в реальному світі.

2.3.5 Інструменти для збірки проєкту

Webpack є одним з найпопулярніших інструментів для збірки модулів (module bundler) у сучасній веб-розробці [21]. Ця технологія відіграє критичну роль у процесі підготовки проєкту до розгортання та забезпечує ефективну роботу складного JavaScript-додатку у браузері.

Основна ідея Webpack полягає в тому, щоб розглядати всі файли проєкту (JavaScript, зображення, тощо) як модулі, які можна обробляти, трансформувати та об'єднувати. Це дозволяє створювати більш структуровані та масштабовані додатки, розділяючи код на невеликі, легко керовані частини.

У контексті проєкту віртуального кампусу, Webpack надає ряд важливих переваг:

- модульність. Webpack дозволяє розбити складний додаток на окремі модулі. Наприклад, це робить можливим створити окремі модулі для різних компонентів системи (різні кімнати, звуки, тощо), що значно полегшує розробку та подальшу підтримку проєкту.

- оптимізація ресурсів. Webpack автоматично оптимізує та мінімізує JavaScript, CSS та інші ресурси. Це особливо важливо для проєкту, де використовується велика кількість 3D-моделей та текстур. Оптимізація забезпечує швидше завантаження та кращу продуктивність додатку.

- управління залежностями. Webpack автоматично аналізує залежності між модулями та забезпечує їх правильне завантаження. Це критично важливо при роботі з великими бібліотеками, такими як Three.js.

- підтримка сучасних технологій. Webpack дозволяє використовувати новітні можливості JavaScript (ES6+) та транспілювати код для підтримки старіших браузерів. Це забезпечує широку сумісність додатку.

- гнучкість налаштування. Завдяки системі плагінів та ладерів, Webpack можна налаштувати під специфічні потреби проєкту. Наприклад, ми можемо додати спеціальні ладери для обробки 3D-моделей або оптимізації текстур.

Використання Webpack у проєкті «Virtual Reality Campus» дозволяє створити добре структурований, оптимізований та легко підтримуваний код. Це особливо важливо для складних 3D-додатків з використанням VR, де продуктивність та ефективне управління ресурсами є критичними факторами.

2.3.6 Середовище розробки та інструментарій

Node.js, хоча і відомий переважно як платформа для створення серверних додатків, відіграє не менш важливу роль у розробці клієнтських веб-додатків, особливо таких складних, як віртуальний університетський кампус. У світі сучасної веб-розробки Node.js став невід'ємною частиною інструментарію розробника, забезпечуючи потужне середовище для створення, тестування та оптимізації JavaScript-додатків [22]. У контексті розробленого проєкту, де основний фокус зосереджений на клієнтській частині з використанням Three.js та WebXR, Node.js виступає як ключовий елемент, що забезпечує ефективність та гнучкість процесу розробки. Потрібно зазначити, як саме Webpack використовується у цьому випадку:

- середовище для інструментів розробки. Node.js служить платформою для запуску Webpack та інших інструментів, необхідних для розробки (наприклад, лінерів, тестових фреймворків).
- управління пакетами. Через npm (Node Package Manager) можливо легко керувати залежностями проєкту, включаючи Three.js, бібліотеки для роботи з WebXR та інші необхідні пакети.
- скрипти збірки та розгортання. Node.js дозволяє створювати та запускати скрипти для автоматизації процесів збірки, тестування та розгортання проєкту.
- локальний сервер для розробки. Хоча система не має повноцінного сервера, можливо використовувати сервер на Node.js для локальної розробки та тестування.

Підсумовуючи, варто відзначити, що хоча Node.js не використовується для створення серверної логіки у проєкті кваліфікаційної роботи, його роль залишається критично важливою. Node.js створює потужну екосистему для розробки, яка значно спрощує та прискорює процес створення складних клієнтських додатків.

Використання Node.js у розробці віртуального кампусу дозволяє оптимізувати робочий процес, забезпечити ефективне управління залежностями та створити гнучке середовище для розробки та тестування. Це особливо важливо при роботі зі складними технологіями, такими як Three.js та WebXR, які вимагають ретельної оптимізації та управління ресурсами.

Більш того, використання Node.js відкриває можливості для майбутнього розширення проєкту. Якщо в подальшому виникне потреба у додаванні серверної функціональності або інтеграції з базами даних, наявність Node.js у стеку технологій значно спростить цей процес.

Таким чином, Node.js виступає не просто як інструмент, а як фундаментальна технологія, що забезпечує ефективність, масштабованість та гнучкість процесу розробки. Це дозволяє зосередитися на створенні інноваційного освітнього контенту, не відволікаючись на технічні складнощі організації процесу розробки та управління проектом.

2.4 Опис структури системи та алгоритмів її функціонування

2.4.1 Програмна архітектура

Для реалізації проєкту «Virtual Reality Campus» з інтеграцією 3D-моделі острова в VR-середовище було обрано архітектуру, орієнтовану на клієнтську частину (client-side architecture). Такий вибір обумовлений необхідністю забезпечити ефективну взаємодію між користувачами та віртуальним середовищем без залежності від серверної частини на поточному етапі розробки [23].

Клієнтська частина системи представлена веб-застосунком, який відповідає за відображення 3D-моделі та забезпечення інтерактивності у VR-режимі. Для реалізації клієнтської частини використовуються такі технології:

- WebGL для рендерингу 3D-графіки у браузері;
- WebXR API для інтеграції з VR-пристроями;

- Three.js як високорівнева бібліотека для роботи з 3D-моделями та сценами;
- HTML, CSS та JavaScript для створення користувацького інтерфейсу та логіки веб-застосунку.

На даному етапі розробки проєкт не містить повноцінної серверної частини та бази даних. Натомість, використовується локальний сервер для розробки та тестування застосунку. Це дозволяє зосередитися на реалізації ключової функціональності VR-середовища та інтеграції 3D-моделі без необхідності розгортання складної серверної інфраструктури [24].

Загальна схема програмної архітектури проєкту представлена на рис. 2.4.

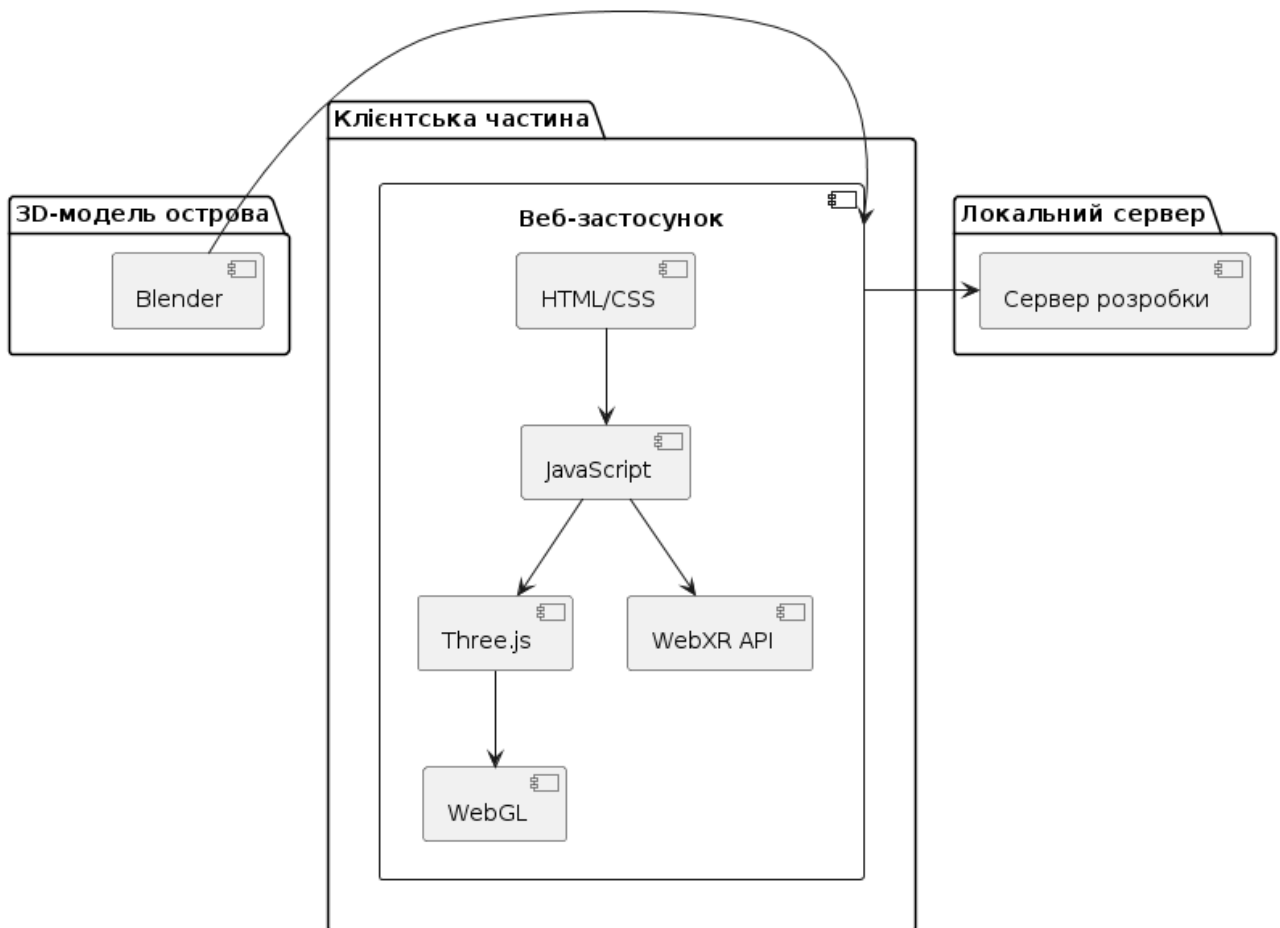


Рис. 2.4 Схема програмної архітектури проєкту «Virtual Reality Campus»
(клієнтська частина)

Така архітектура дозволяє швидко розробляти та тестувати основні функції VR-середовища, забезпечуючи високу продуктивність та інтерактивність завдяки використанню клієнтських технологій.

Важливо зазначити, що обрана архітектура є гнучкою та модульною, що дозволяє в майбутньому додати серверну частину та базу даних для реалізації більш складних функцій, таких як збереження результатів експериментів, синхронізація стану віртуального середовища між користувачами тощо.

2.4.2 Логічна структура 3D-сцени

Розроблена 3D-сцена острова для інтеграції у VR-середовище в рамках проекту «Virtual Reality Campus» складається зі 197 ретельно змодельованих елементів, які в сукупності формують детальне та реалістичне віртуальне середовище для навчання та експериментів з енергетичними системами.

Ключовими компонентами сцени є 20 домівок, сонячні батареї (36 шт.), кондиціонери на будівлях (8 шт.), сонячний інвертор, станції для заряджання електромобілів (12 шт.), що дозволяє симулювати реальні енергетичні потреби житлового сектора. Додатковими джерелами енергії на острові виступають 3 вітряки та паливний елемент. Централізоване управління енергосистемою острова здійснюється через будівлю міської ради.

Транспортна інфраструктура острова представлена 12 електромашинами та дерев'яним мостом. Для забезпечення естетичної привабливості та реалістичності сцени були змодельовані численні елементи ландшафту та оточення, такі як дерева (54 шт.), огорожі (20 шт.), човни (5 шт.), лавки (3 шт.), острів, гора та вода.

Особлива увага була приділена оптимізації 3D-моделей для забезпечення ефективного рендерингу та плавної роботи у режимі VR. Усі об'єкти сцени були змодельовані з використанням низькополігонального моделювання, що дозволило скоротити обчислювальне навантаження без істотного впливу на візуальну якість.

Прикладом такої оптимізації є модель сонячної панелі (рис. 2.5). Початкова версія моделі мала надлишкову деталізацію та кількість полігонів, що могло негативно вплинути на продуктивність рендерингу у VR. Після ретельного аналізу та спрощення геометрії, кількість полігонів було скорочено, при цьому збережено ключові візуальні характеристики об'єкта.

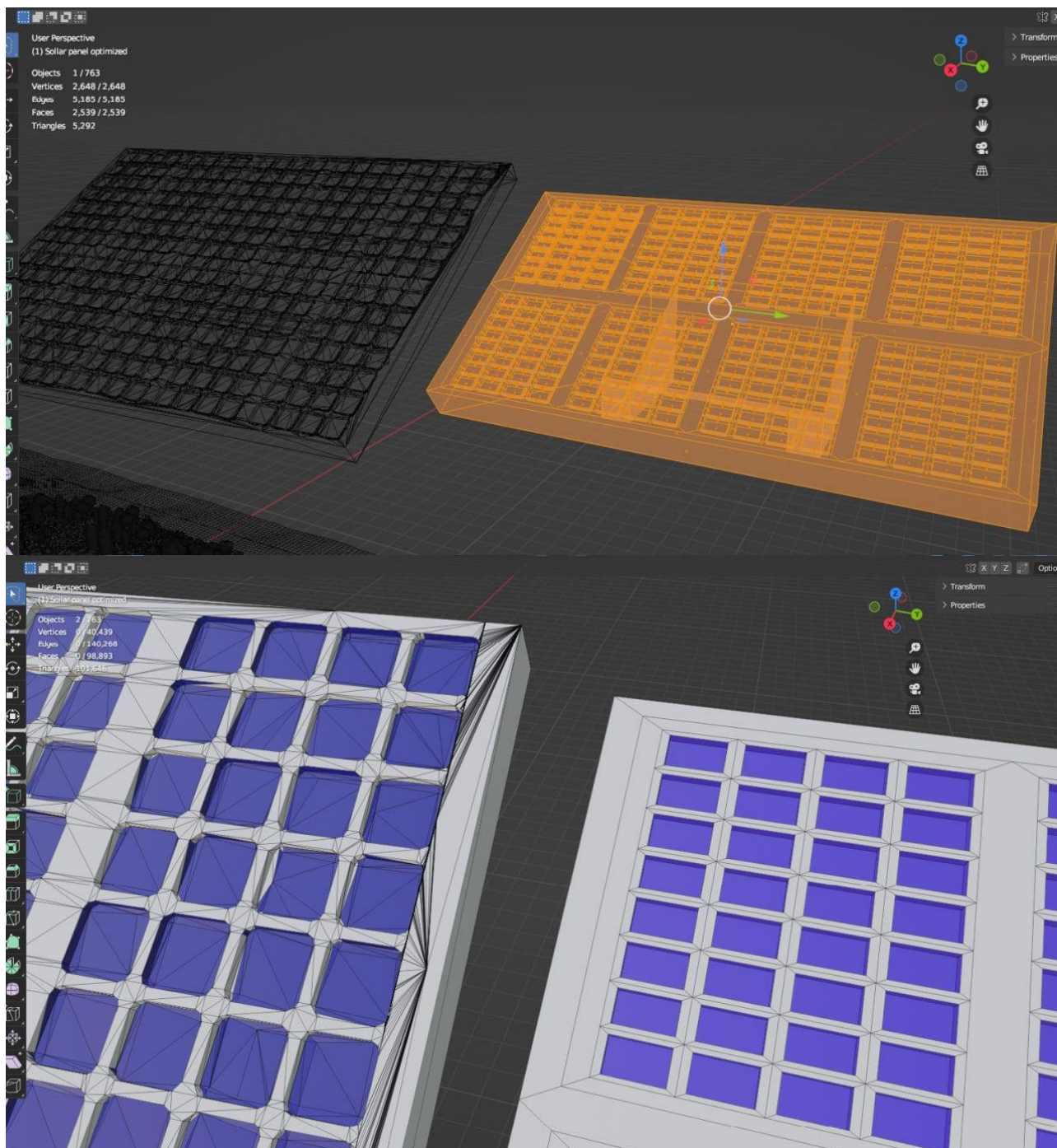


Рис. 2.5. Оптимізація 3D-моделі сонячної панелі

Розвиток сцени можна простежити на прикладі порівняння її ранньої та поточної версій (рис. 2.6 - 2.7). Початковий вигляд острова мав більш схематичний характер та спрощену геометрію. В процесі розробки, сцена збагачувалась додатковими елементами, текстурами та деталями, що дозволило досягти вищого рівня реалістичності та занурення.



Рис. 2.6. - 2.7. Еволюція 3D-сцени острова від ранньої (зверху) до поточної (внизу) версії

Окрім того, оскільки змодельована 3D-модель є відображенням фізичної моделі острова, що знаходиться в Ройтлінгенському університеті, то вона додатково має основу, кнопки (5 шт.) та емблему. На рис. 2.8 – 2.9 зображено фізичну модель наявну в університеті та її інтерпретацію у 3D.

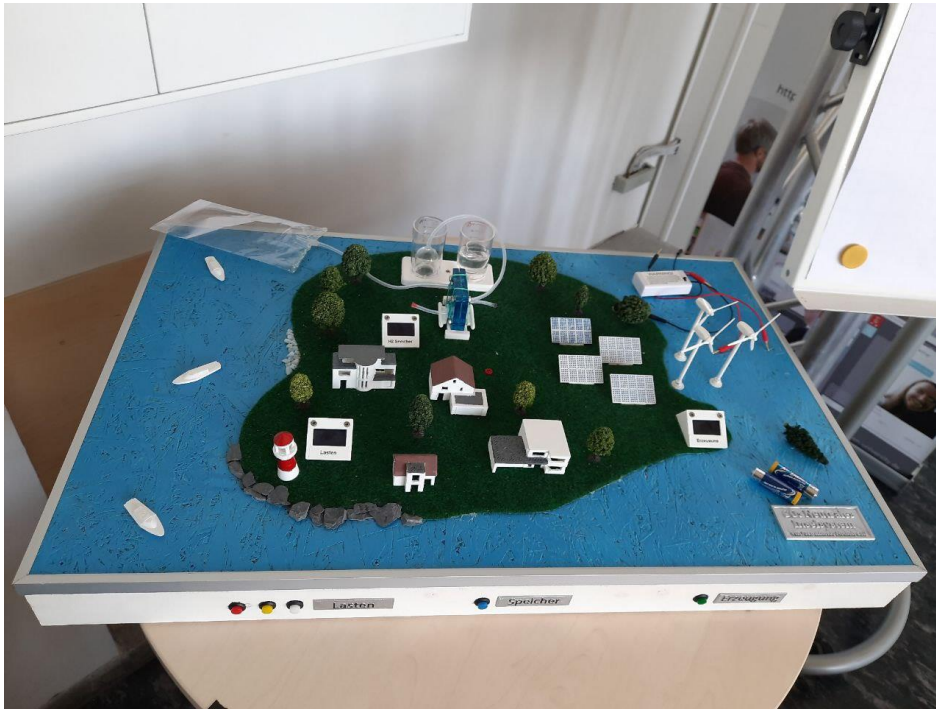


Рис. 2.8. Фізична модель острова, що наявна у Ройтлінгенському університеті



Рис. 2.9 Розроблена 3D-версія острова

Підсумовуючи, розроблена 3D-сцена острова демонструє детально пропрацьоване та оптимізоване віртуальне середовище, яке поєднує в собі ключові елементи енергетичної інфраструктури, житлового сектора та ландшафтного оточення. Така структура сцени створює міцну основу для реалістичного та захопливого навчального досвіду в рамках проекту «Virtual Reality Campus».

2.4.3. Адаптація 3D-моделей і візуалізації даних для VR-середовища

Розробка 3D-сцени в Blender для інтеграції в освітній процес потребує ретельного розгляду процесу адаптації для віртуальної реальності. Потрібно зазначити ключові аспекти адаптації 3D-моделей, середовища VR та візуалізації даних для забезпечення безперебійного та ефективного користувацького досвіду. 3D-моделі, призначені для VR-середовищ, повинні відповідати конкретним вимогам для забезпечення оптимальної продуктивності та якості зображення. Наступні аспекти є надзвичайно важливими:

- оптимізація. VR-додатки часто мають суворіші обмеження щодо продуктивності через рендеринг у реальному часі та потребу у плавних взаємодіях. Крім того, VR-середовища вимагають високої частоти кадрів, щоб уникнути захитування та підтримувати імерсивність. 3D-моделі повинні бути оптимізовані для зменшення кількості полігонів, розмірів текстур та загальної складності, при цьому зберігаючи візуальну достовірність. Оптимізація складності сцени, освітлення та налаштувань рендерингу є вирішальною для досягнення плавної продуктивності;

- текстурування та матеріали. Текстури та матеріали відіграють важливу роль у створенні реалістичних зображень у VR. Текстури високої роздільної здатності з властивостями фізично коректного рендерингу (PBR) можуть покращити досвід користування. Крім того, врахування умов освітлення в цільовому середовищі є важливим для точного представлення матеріалів;

- інтерактивність. У VR користувачі у перспективі матимуть змогу взаємодіяти з 3D-сценою. Додавання інтерактивних елементів, таких як кнопки, тригери або анімації, може підвищити залученість і забезпечити більш інтуїтивний досвід;

- масштаб та пропорції. Забезпечення правильного масштабу та пропорцій 3D-об'єктів є вирішальним для підтримання реалістичності у VR-середовищі. Невірний масштаб може призвести до дискомфортного досвіду для користувачів.

Успішна розробка 3D-сцени в Blender для інтеграції в освітній процес з використанням VR-технологій вимагає детального підходу до адаптації моделей, середовища та візуалізації даних. Оптимізація 3D-моделей для досягнення високої продуктивності, використання високоякісних текстур та матеріалів, додавання інтерактивних елементів, а також забезпечення правильного масштабу та пропорцій є ключовими аспектами для створення реалістичного та занурюючого користувацького досвіду. Виконання цих аспектів дозволить забезпечити ефективну та плавну роботу VR-додатку, що сприятиме більш глибокому залученню та навчанню користувачів.

2.4.4. Файлова структура

Проект «Virtual Reality Campus» з інтегрованою 3D-сценою острова має чітко організовану та логічну файлову структуру, яка забезпечує ефективну навігацію, розробку та підтримку коду. Кожен файл та папка мають своє призначення та відіграють важливу роль у функціонуванні системи.

Основна структура проекту складається з наступних елементів:

- Папка `src`: містить вихідний код проекту;
- Папка `assets`: зберігає статичні ресурси, такі як 3D-моделі, текстури та зображення;
- Папка `node_modules`: зберігає усі залежності системи;

- Папка `components`: містить компоненти інтерфейсу користувача;
- Папка `scenes`: містить файли, які описують 3D-сцени та їх налаштування;
- Папка `Blender`: містить файл з 3D-сценою у форматі `.blend`;
- Папка `utils`: містить допоміжні функції та утиліти;
- Папка `dist`: містить оптимізовану та зібрану версію проєкту, готову до розгортання;
- Папка `rooms`: містить файли, які описують наявні кімнати у проєкті;
- Файл `bundle.js`: зв'язує усі файли JavaScript у проєкті;
- Файл `package.json`: містить метадані проєкту та список залежностей;
- Файл `webpack.config.js`: містить конфігурацію збірки проєкту за допомогою Webpack;
- Файл `assets.js`: вказує шлях до файлу текстури сцени та описує її для відображення у VR;
- Файл `PhotogrammetryObject.js`: створює об'єкт для VR, отримуючи його з файлу 3D-сцени та описує його.

2.5. Обґрунтування та організація вхідних та вихідних даних програми

2.5.1. Вхідні дані

Для забезпечення коректного функціонування системи віртуального університетського кампусу з інтегрованою 3D-моделлю енергетичного острова необхідний певний набір вхідних даних. Ці дані служать основою для генерації віртуального середовища, симуляції енергетичних процесів та забезпечення інтерактивності в режимі реального часу.

Вхідні дані для системи можна розділити на дві основні категорії:

1. Дані про положення та орієнтацію користувача у просторі:

- дані, що надходять безпосередньо з VR-пристроїв, таких як шоломи віртуальної реальності та контролери;
- WebXR API автоматично отримує інформацію про положення та орієнтацію користувача за допомогою вбудованих сенсорів та системи трекінгу VR-пристроїв;
- дані, що передаються у форматі матриць трансформації (матриці поворотів і переміщень), які описують розташування та орієнтацію користувача відносно початкової точки відліку у віртуальному просторі (рис. 2.10);

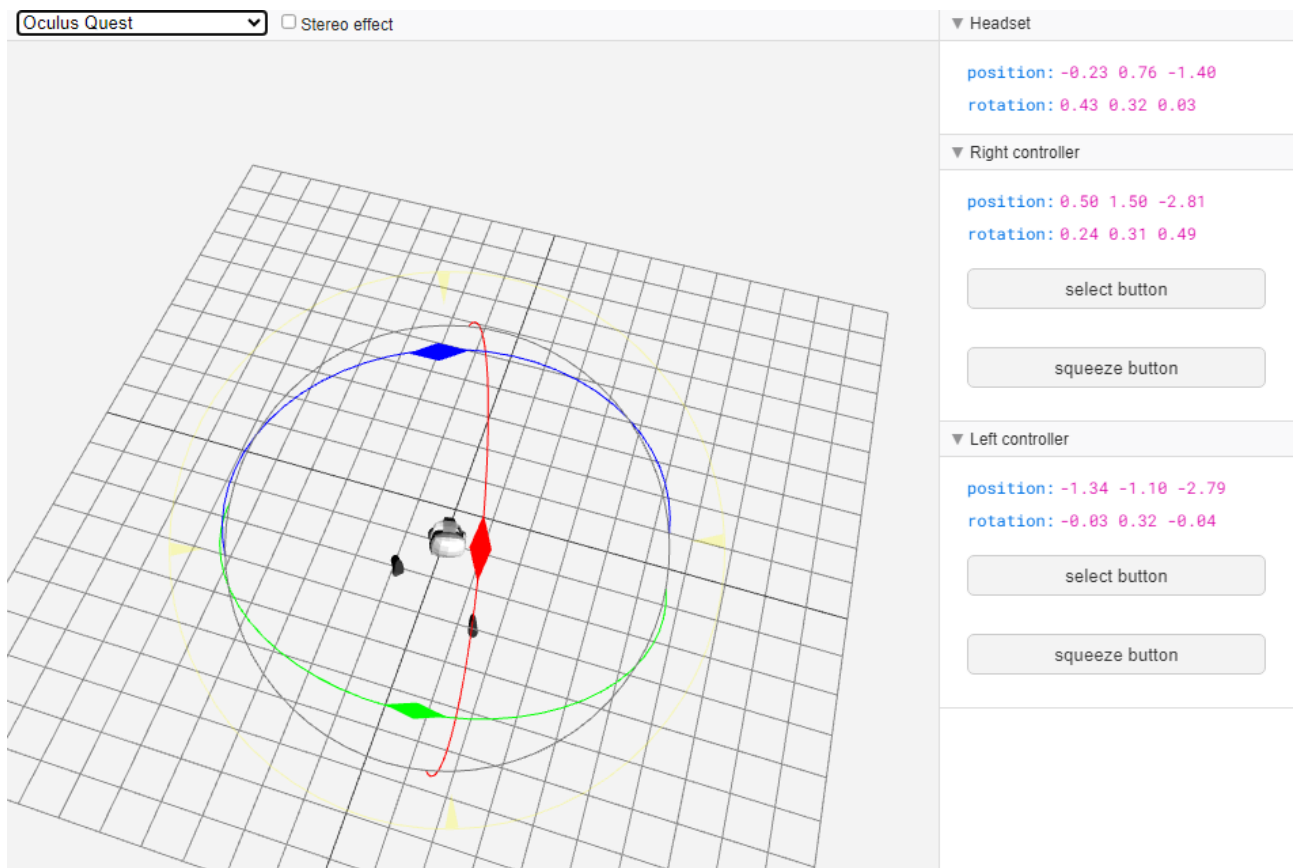


Рис. 2.10 Приклад матриці трансформації, що використовує WebXR

- дані, що оновлюються в режимі реального часу з високою частотою (приблизно 60 разів на секунду) для забезпечення плавності та реалістичності досвіду віртуальної реальності.

2. Параметри стану віртуального середовища та енергетичної системи острова:

– дані, що включають інформацію про поточний стан різних компонентів 3D-сцени, таких як положення та стан об'єктів (будинки, транспортні засоби, джерела енергії тощо);

– у довгостроковій перспективі мають бути додані параметри енергетичної системи, такі як рівні генерації та споживання енергії, заряд акумуляторів, стан мережі тощо;

– дані, що надходять з різних джерел, таких як налаштування сцени, скрипти симуляції або зовнішні файли конфігурації. Формат даних залежить від конкретної реалізації та включає JSON, XML або спеціалізовані формати, такі як glTF для 3D-моделей.

Вхідні дані завантажуються та обробляються під час ініціалізації системи та можуть оновлюватися динамічно під час роботи симуляції. Тим паче, організація та попередня підготовка вхідних даних є важливим аспектом розробки системи віртуального університетського кампусу. Ефективна обробка та використання цих даних дозволяють створити імерсивне та інтерактивне середовище, яке реагує на дії користувача та забезпечує реалістичну симуляцію енергетичних процесів на острові.

2.5.2. Вихідні дані

Вихідні дані системи віртуального університетського кампусу є результатом обробки та інтерпретації вхідних даних, а також симуляції процесів у віртуальному середовищі. Ці дані представляють собою інформацію, яка безпосередньо сприймається та взаємодіє з користувачем через VR-пристрої.

Вихідні дані системи можна розділити на такі основні категорії:

1. Зображення, що відображається користувачеві у VR-пристрої:

- основний вихідний продукт системи - інтерактивне 3D-зображення віртуального острова, яке користувач бачить через VR-шолом;
- зображення, що генерується в режимі реального часу на основі положення та орієнтації користувача, а також стану віртуального середовища;
- WebXR API та Three.js відповідають за рендеринг 3D-сцени з урахуванням усіх необхідних параметрів, таких як освітлення, матеріали, тіні тощо;
- результуюче зображення, що розділяється на два окремі зображення (для лівого та правого ока) з відповідними перспективними трансформаціями для створення стереоскопічного ефекту глибини.

2. Дані, що відображаються на віртуальних інформаційних панелях та інтерфейсах:

- додаткова інформація, яка надається користувачеві у віртуальному середовищі, наприклад, показники енергоспоживання, стан обладнання, підказки тощо;
- дані, що генеруються на основі поточного стану симуляції та відображаються за допомогою елементів інтерфейсу користувача (UI) у 3D-просторі. Формат даних залежить від конкретного типу інформації та може включати текст, числа, графіки, піктограми тощо.

3. Звукові ефекти та просторове аудіо:

- для посилення занурення у віртуальне середовище система може генерувати звукові ефекти та просторове аудіо;
- звуки, що включають ambient-шуми (вітер, спів пташок), звуки взаємодії з об'єктами (кроки, перехід у портал), звукові сповіщення тощо;

- просторове аудіо дозволяє позиціонувати джерела звуку у 3D-просторі відносно користувача, створюючи реалістичний звуковий ландшафт;

- формат використаних аудіоданих – OGG.

Вихідні дані генеруються системою в режимі реального часу та безперервно оновлюються відповідно до дій користувача та змін у віртуальному середовищі. Вони передаються через WebXR API та відображаються за допомогою VR-пристроїв, забезпечуючи інтерактивний та імерсивний досвід для користувача.

Тож організація та формат вихідних даних системи віртуального університетського кампусу відіграють ключову роль у забезпеченні ефективної взаємодії користувача з віртуальним середовищем. Ретельно продумана структура та подання вихідних даних дозволяють створити переконливий та інтуїтивно зрозумілий інтерфейс, який полегшує навігацію та взаємодію з віртуальним островом.

2.6. Опис розробленої системи

2.6.1 Використані технічні засоби

Для розробки та тестування 3D-сцени острова в рамках проєкту «Virtual Reality Campus» використовувався потужний ігровий ноутбук ASUS ROG G17 (2023) G713 [25]. Цей пристрій оснащений процесором Intel Core i9-13900H, відеокартою NVIDIA GeForce RTX 4070 Laptop GPU та 16 ГБ оперативної пам'яті стандарту DDR5, що забезпечує високу продуктивність для роботи з ресурсомісткими 3D-додатками, такими як Blender.

Для забезпечення точного та ергономічного введення використовувалась ігрова мишка Razer Abyssus V2 [26]. Ця модель має оптичний сенсор з роздільною здатністю 5000 DPI та частотою опитування 1000 Гц, що дозволяє досягти високої точності та швидкості відстеження рухів.

Для розширення робочого простору та підвищення ефективності процесу розробки був задіяний додатковий монітор HP Compaq LA1951g з діагоналлю 19 дюймів та роздільною здатністю 1280x1024 [27]. Це дозволило організувати зручне розташування вікон 3D-редактора, текстового редактора та інших допоміжних інструментів.

Тестування сцени у віртуальній реальності здійснювалось за допомогою автономного VR-шолома Oculus Go та його контролера [28]. Oculus Go оснащений дисплеєм з роздільною здатністю 2560x1440 та частотою оновлення 72 Гц, що забезпечує чітке та плавне зображення. Контролер дозволяє здійснювати інтуїтивне керування та взаємодію з об'єктами віртуального середовища.

Також потрібно зазначити, що розроблений додаток сумісний з різними технічними засобами та конфігураціями, за умови, що їх параметри відповідають або перевищують мінімальні вимоги, зазначені у пункті 1.5.3 «Вимоги до складу та параметрів технічних засобів».

2.6.2. Використані програмні засоби

У процесі створення цієї кваліфікаційної роботи було залучено низку допоміжних програмних інструментів, зокрема:

– Blender - потужний відкритий пакет для 3D-моделювання, анімації та рендерингу [29]. Він пропонує широкий набір функцій для полігонального моделювання, текстурювання, освітлення та роботи з матеріалами, що дозволило створити деталізовану та реалістичну 3D-сцену;

– Adobe Photoshop – графічний редактор, що використовувався для створення та редагування текстур [30]. Завдяки розширеним можливостям роботи з зображеннями та потужним інструментам малювання, Photoshop дозволив створити високоякісні текстури для моделей та оптимізувати їх для використання в 3D-сцені;

– Visual Studio Code - Середовищем розробки, що використовувалося для створення клієнтської частини проєкту [31]. Він є потужним та гнучким редактором коду від Microsoft. VS Code надає широкий спектр вбудованих інструментів та розширень для роботи з JavaScript, HTML, CSS та іншими веб-технологіями;

– Node.js - кросплатформна платформа для виконання JavaScript-коду поза веб-браузером та забезпечення серверного оточення [32]. Він надає можливості для створення масштабованих мережових додатків та забезпечує багату екосистему пакетів та бібліотек через менеджер пакетів npm;

– GitLab - система контролю версій, що використовувалася для ефективного управління вихідним кодом проєкту, відстеження змін та колаборації з іншими членами команди розробки [33]. GitLab надає зручний веб-інтерфейс для роботи з Git-репозиторіями, а також містить додаткові функції, такі як система відстеження помилок;

– Google Chrome, Firefox, Safari, Opera - популярних веб-браузери, що використовувалися для тестування та налагодження веб-частини проєкту. Це дозволило забезпечити кросбраузерну сумісність та оптимізувати роботу 3D-сцени у веб-середовищі;

– Windows 11 Home - операційна система, на якій здійснювалася розробка [34]. Ця версія ОС від Microsoft надає зручне та інтуїтивне середовище для роботи з різноманітними програмними інструментами та забезпечує підтримку широкого спектру апаратного забезпечення.

2.6.3 Виклик та завантаження програми

З точки зору розробника, для запуску веб-застосунку на своєму комп'ютері спочатку необхідно розпакувати архів з вихідним кодом проєкту, розташований на флеш-накопичувачі. Для запуску проєкту необхідно виконати наступні кроки в зазначеному порядку:

1. Встановити Node.js версії 12.22.12 на комп'ютер та npm версії 7.5.2

для коректного запуску програми;

2. Встановити веб-браузер Google Chrome, оскільки він має вбудовану підтримку WebGL та забезпечує високу продуктивність при роботі з 3D-графікою.

3. Під'єднати флеш-накопичувач до комп'ютера та відкрити його вміст.

4. Відкрити термінал або командний рядок та перейти до кореневої папки проєкту на флеш-накопичувачі.

5. Виконати команду «npm install» для встановлення всіх необхідних залежностей проєкту, включаючи Webpack.

6. Виконати команду «npm start» для збірки проєкту за допомогою Webpack.

7. Відкрити Google Chrome та перейти за адресою, яка була виведена в терміналі після запуску сервера (зазвичай це <https://localhost:8080>).

8. Встановити розширення WebXR API Emulator для Google Chrome. Це розширення дозволяє емулювати функціональність VR-пристроїв у браузері.

9. За відсутності VR-шолому та контролерів потрібно активувати розширення WebXR API Emulator та перейти в консоль розробника і обрати опцію «Oculus Go» (або будь-який інший запропонований варіант) для емуляції відповідного VR-пристрою в полі WebXR (рис. 2.11). Доступні опції та панель керування симулятором зображенні на рис. 2.12.

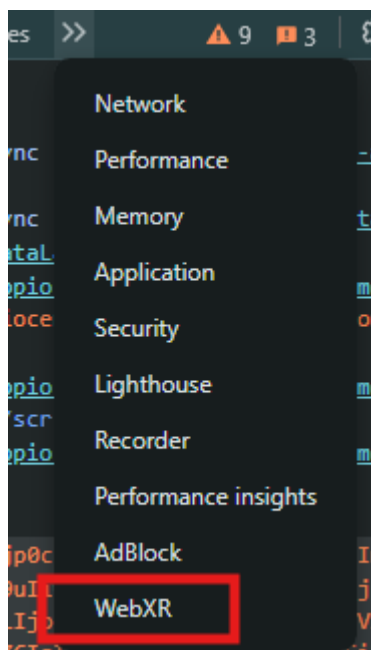


Рис. 2.11. Консоль розробника

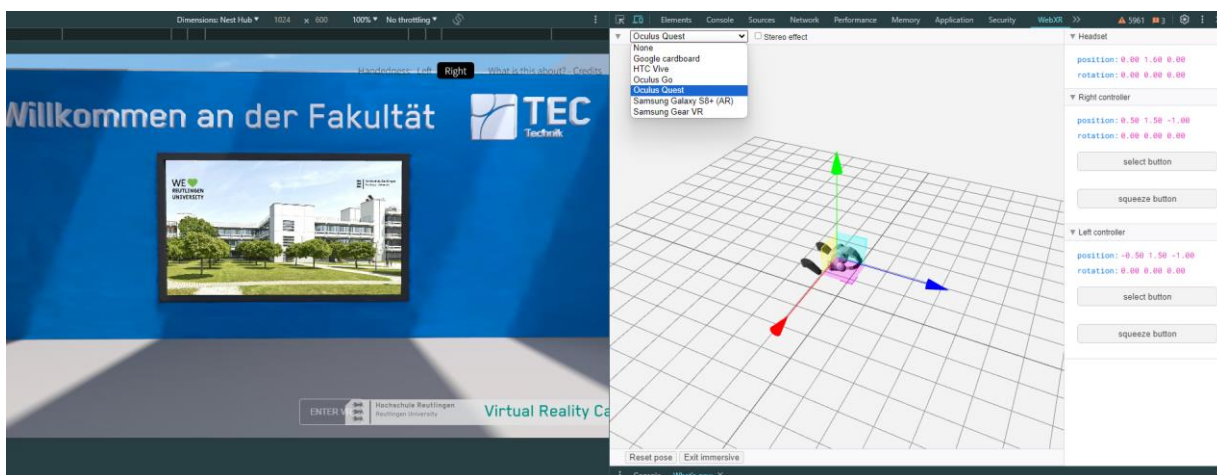


Рис. 2.12. Доступні варіанти емуляції VR-пристроїв та навігаційна панель

10. Натиснути кнопку «Enter VR» на сторінці проєкту для запуску VR-режимі.

11. Оскільки проєкт «Virtual Reality Campus», що розробляється у межах Ройтлінгенського університету має не лише розроблену протягом виконання цієї кваліфікаційної роботи модель острова, а й інші елементи, то потрібно перейти до кімнати, де знаходиться саме розроблена 3D-сцена. Для цього за допомогою навігаційної панелі потрібно повернути VR-шолом ліворуч та використовуючи контролер перейти у відповідний портал у стіні.

12. За наявності VR-пристрою потрібно відкрити браузер безпосередньо у системі та перейти за адресою, що складається із персональної IP-адреси, яку можна дізнатися ввівши у cmd команду «ipconfig» та знайшовши поле «IPv4 Address», та посилання, що було виведено у терміналі після запуску серверу. Приклад наведено на рис. 2.13.

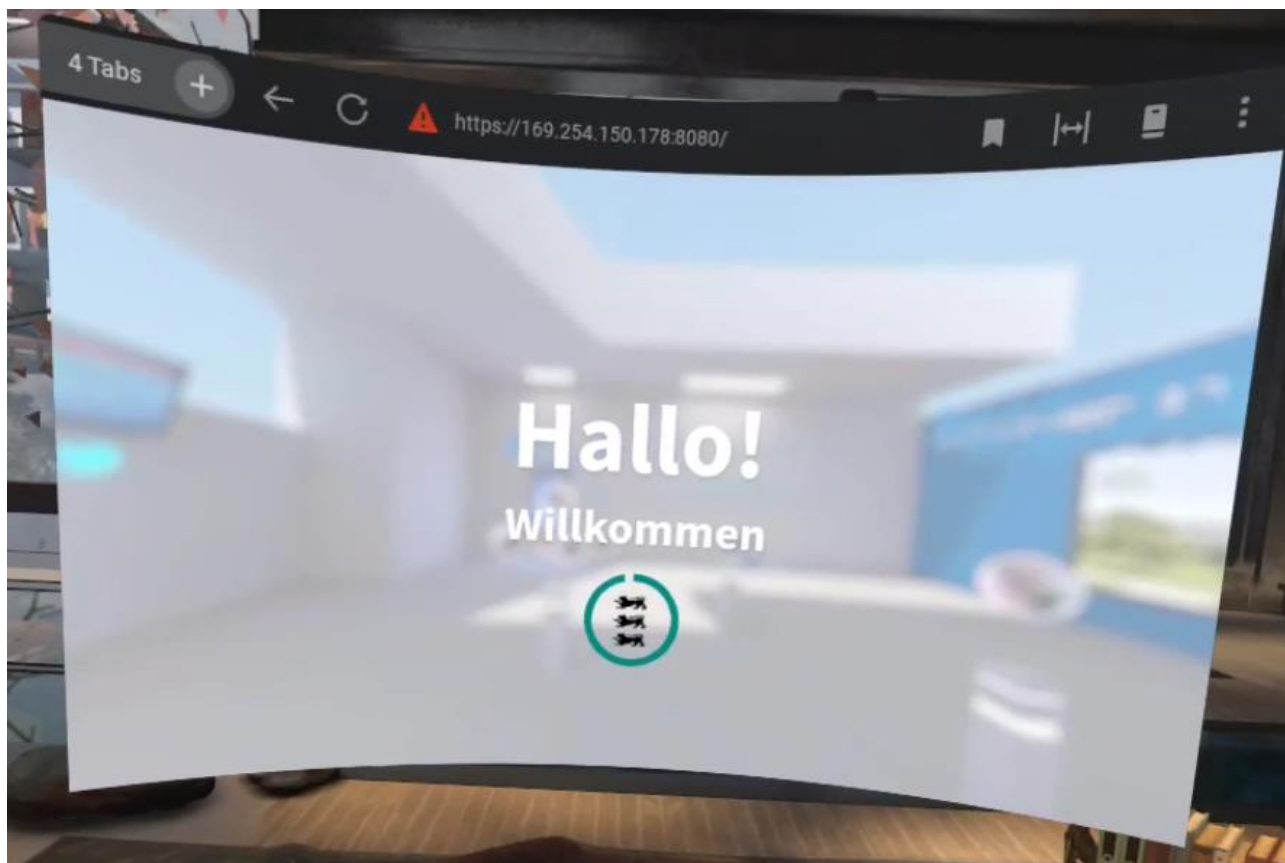


Рис. 2.13. Завантаження веб-сайту за допомогою VR-гарнітури Oculus Go

13. Оскільки веб-сайт безпосередньо відкритий через VR-гарнітуру, то можливо одразу розпочати сесію, натиснувши «ENTER VR» (рис. 2.14).



Рис. 2.14. Вітальне вікно перед початком VR-сесії

14. Як було зазначено вище, проєкт «Virtual Reality Campus» має не лише модель, створення якої описано в ході цієї кваліфікаційної роботи, тож перше, що побачить користувач – це вітальна кімната з логотипом Ройтлінгенського університету, літаючими кульками у яких доступні відео 360°, моделлю літака та порталами на стінах, що ведуть у різні кімнати, що зображено на рис. 2.15 – 2.18. Для того щоб потрапити у кімнату з островом, користувачу потрібно буде перейти у портал, що знаходиться на лівій стіні (рис. 2.18)



Рис. 2.15. Вітальна кімната. Перше, що бачить користувач, розпочавши VR-сесію

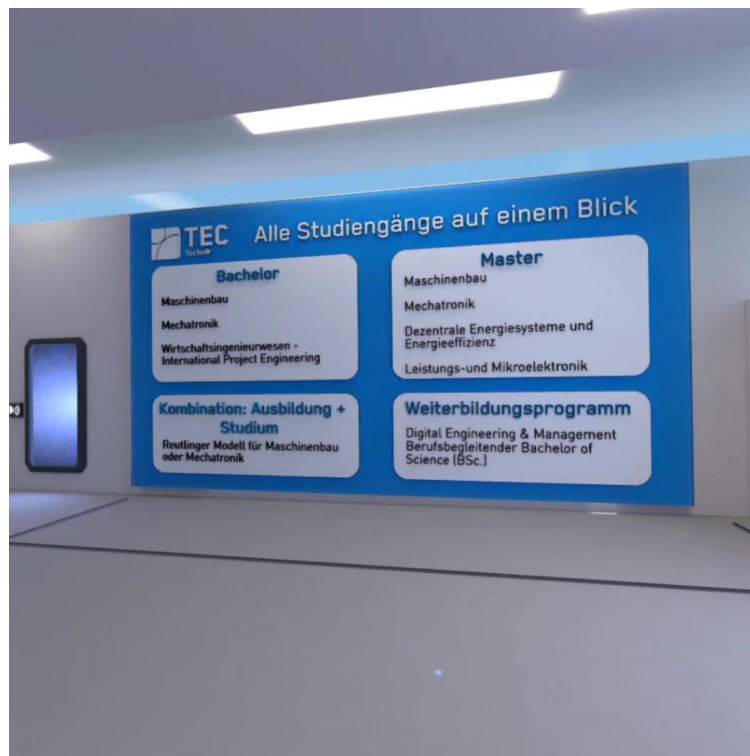


Рис. 2.16. Вітальна кімната. Задня стіна та портал через який користувач потрапив до кімнати



Рис. 2.17. Вітальна кімната. Стіна праворуч з логотипом та фото факультету

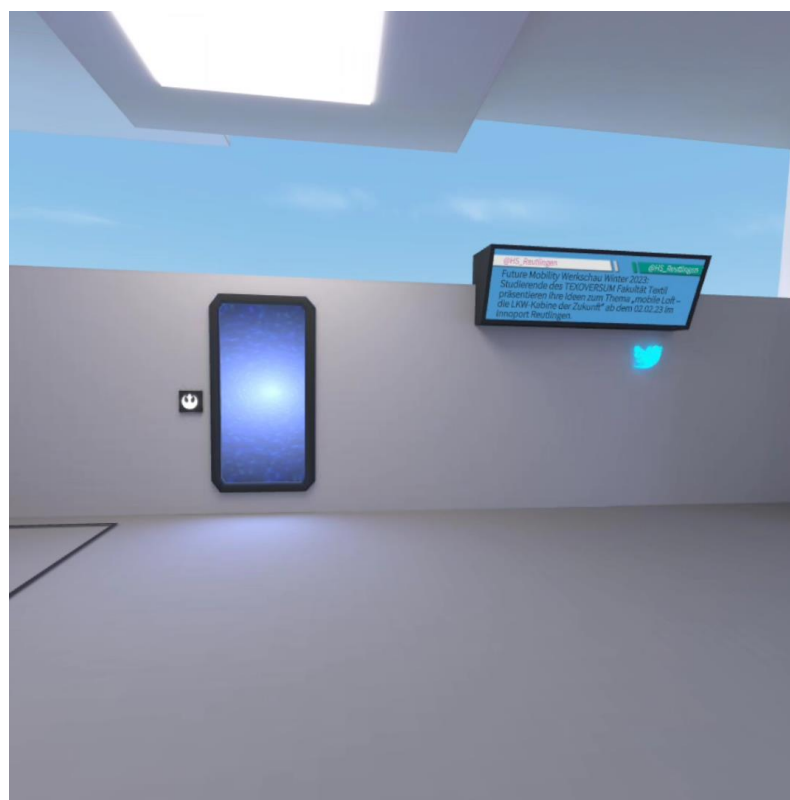


Рис. 2.18. Вітальна кімната. Ліва стіна з порталом на кімнати з островом

15. Надалі користувач потрапляє у кімнату з 3D-моделлю острова, що описана у цій кваліфікаційній роботі (рис. 2.19)

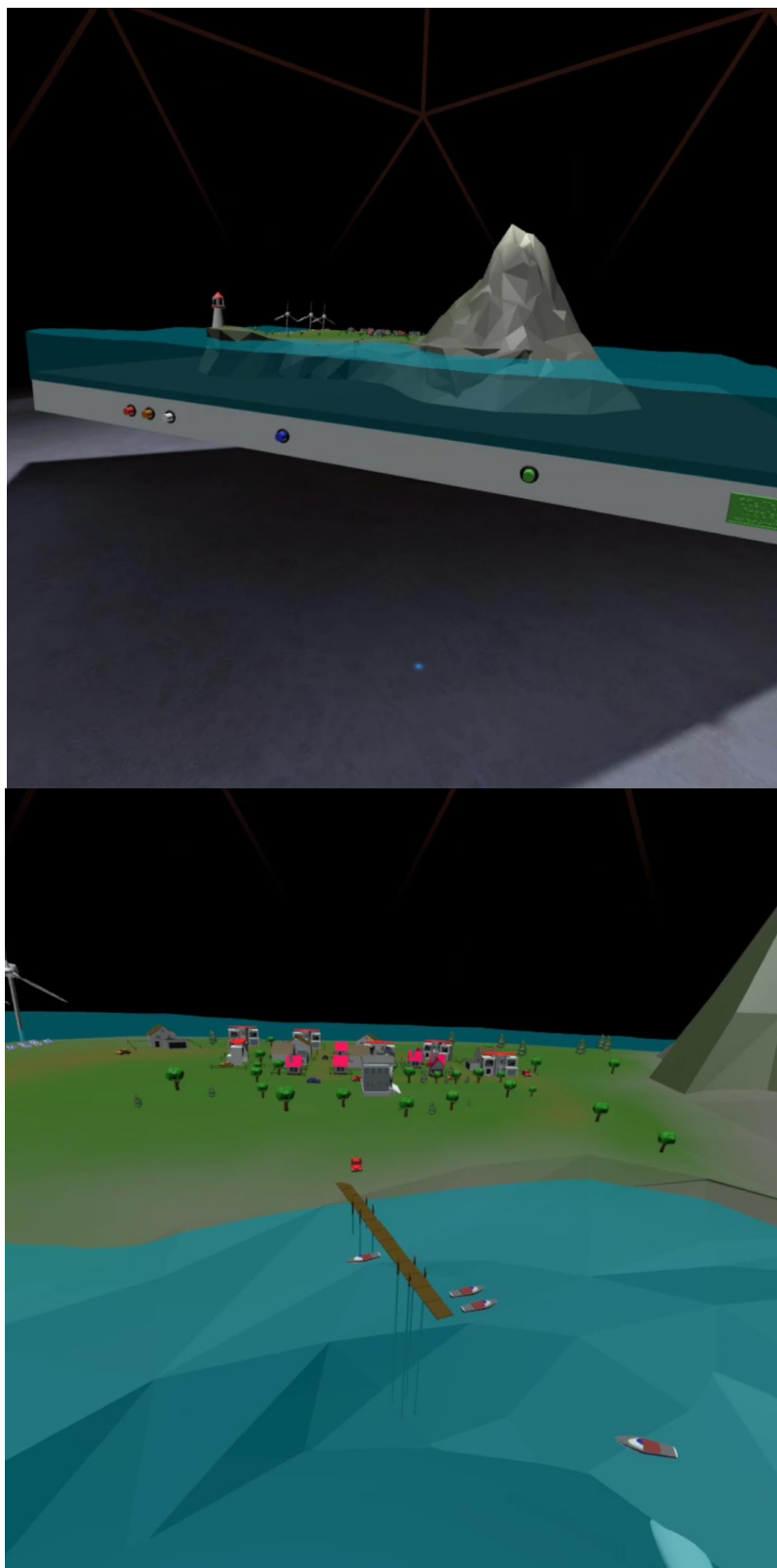


Рис. 2.19. Кімната з островом

2.6.4. Опис інтерфейсу користувача

Користувач матиме змогу взаємодіяти з розробленою 3D-моделлю острова у Blender для інтеграції у VR-середовище складається зі 197 власноруч змодельованих елементів, кожен з яких виконує певну роль у формуванні цілісного та реалістичного інтерфейсу та віртуального простору. Усі об'єкти були змодельовані за допомогою програмного забезпечення Blender та оптимізовані для ефективного відображення у режимі реального часу. Загальний вигляд 3D-сцени зображений на рис. 2.20.

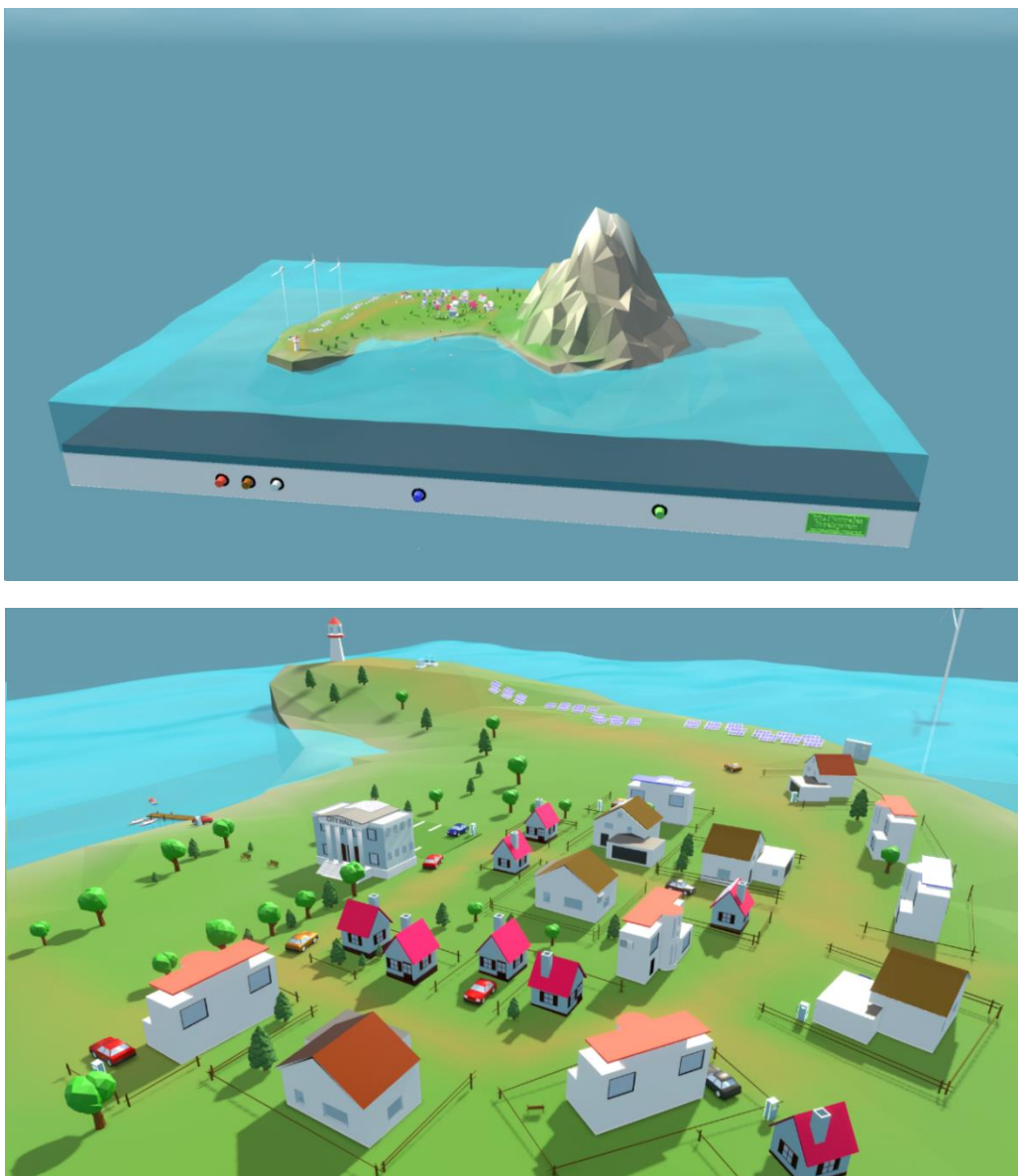


Рис. 2.20. Загальний вигляд розробленої 3D-сцени

Основні категорії розроблених об'єктів включають:

1. Будівлі та споруди:

– житлові будинки (20 шт.). Різноманітні за архітектурою та розмірами, призначені для імітації житлової зони острова (рис. 2.21).



Рис. 2.21. Приклади розроблених житлових будинків

– адміністративна будівля міської ради (1 шт.): центральна споруда, яка слугує центром управління енергетичною системою острова (рис. 2.22)



Рис. 2.22. Змодельована будівля міської ради

2. Об'єкти енергетичної інфраструктури:

– сонячні батареї (36 шт.): змодельовані з високим рівнем деталізації, розміщені на дахах будівель та окремих опорних конструкціях (рис. 2.23 – 2.25).

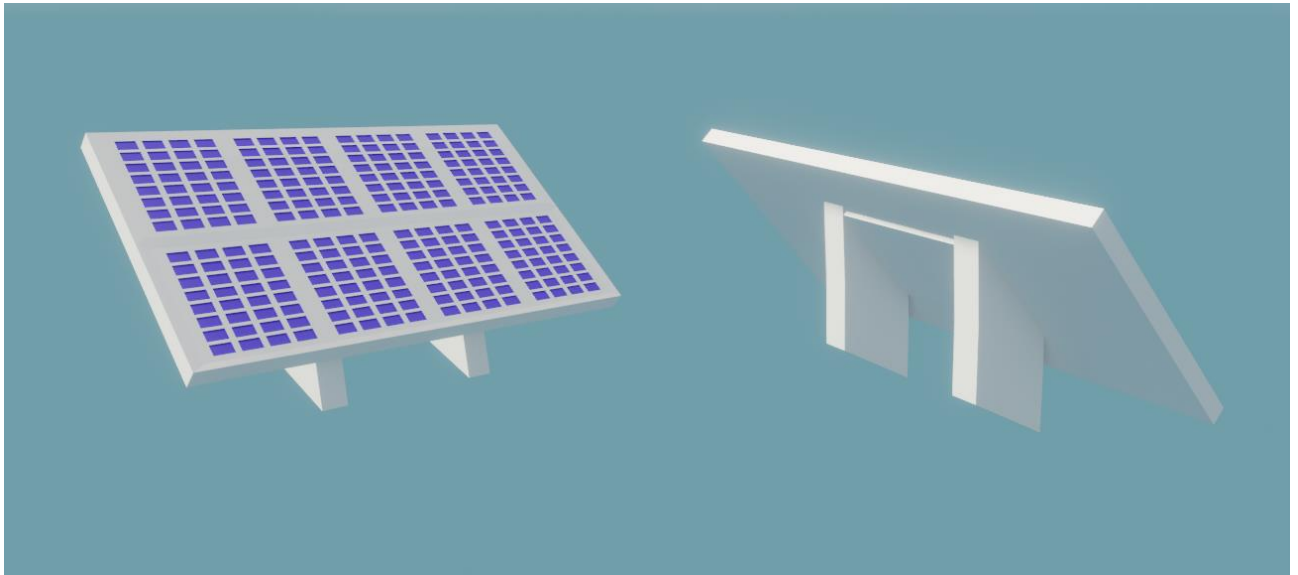


Рис. 2.23. Змодельовані сонячні батареї



Рис. 2.24. Приклад розміщення сонячних батарей на дахах будівель

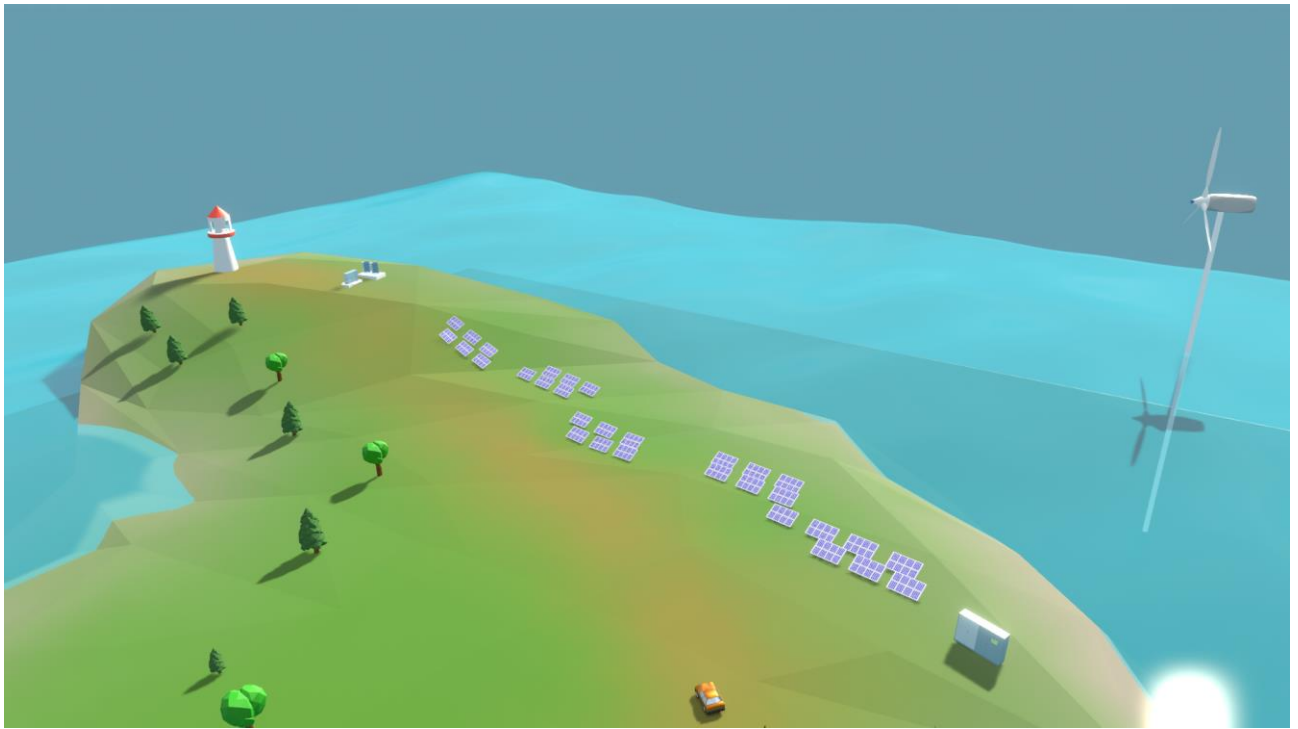


Рис. 2.25. Розташування сонячних панелей на острові

– сонячний інвертер (1 шт.). Забезпечує перетворення сонячної енергії на електричну (рис. 2.26).



Рис. 2.26. Змодельований сонячний інвертер

– вітрогенератори (3 шт.): реалістичні моделі вітряків, які обертаються та генерують енергію (рис. 2.27).

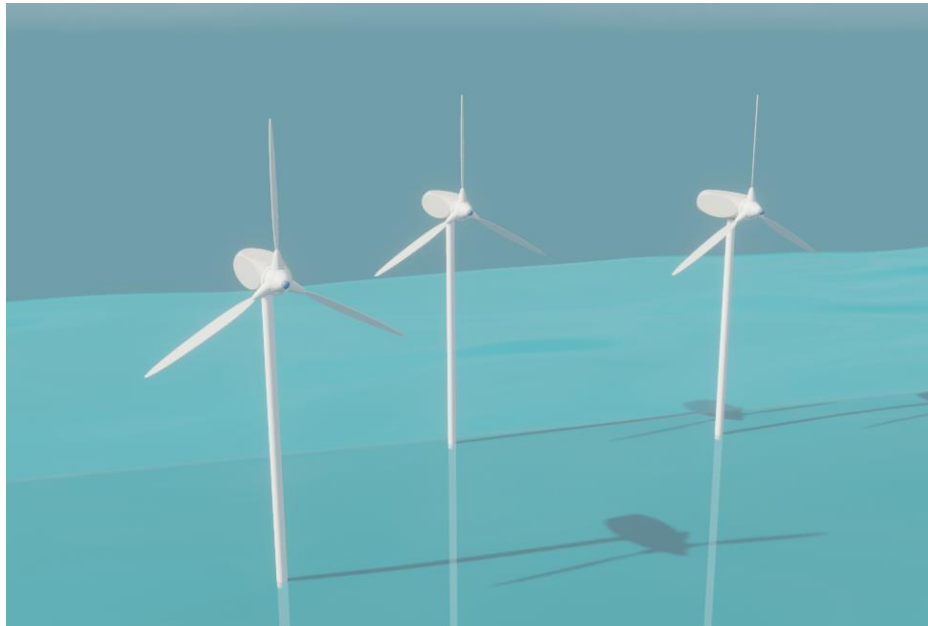


Рис. 2.27. Змодельовані вітряки, що розташовані на воді

– паливний елемент (1 шт). Електрохімічний генератор, який забезпечує пряме перетворення хімічної енергії на електричну (рис. 2.28).

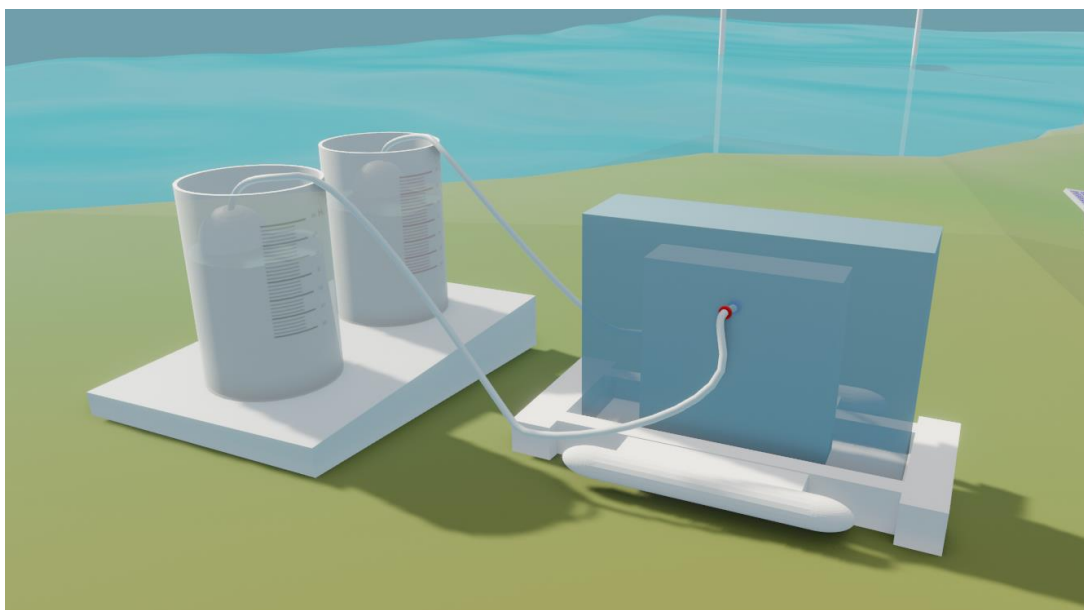


Рис. 2.28. Змодельований паливний елемент

– кондиціонери, 8 шт. Реалістичні моделі кондиціонерів, розміщені на стінах житлових будинків, які імітують системи охолодження та підтримки комфортного мікроклімату в приміщеннях (рис. 2.29).

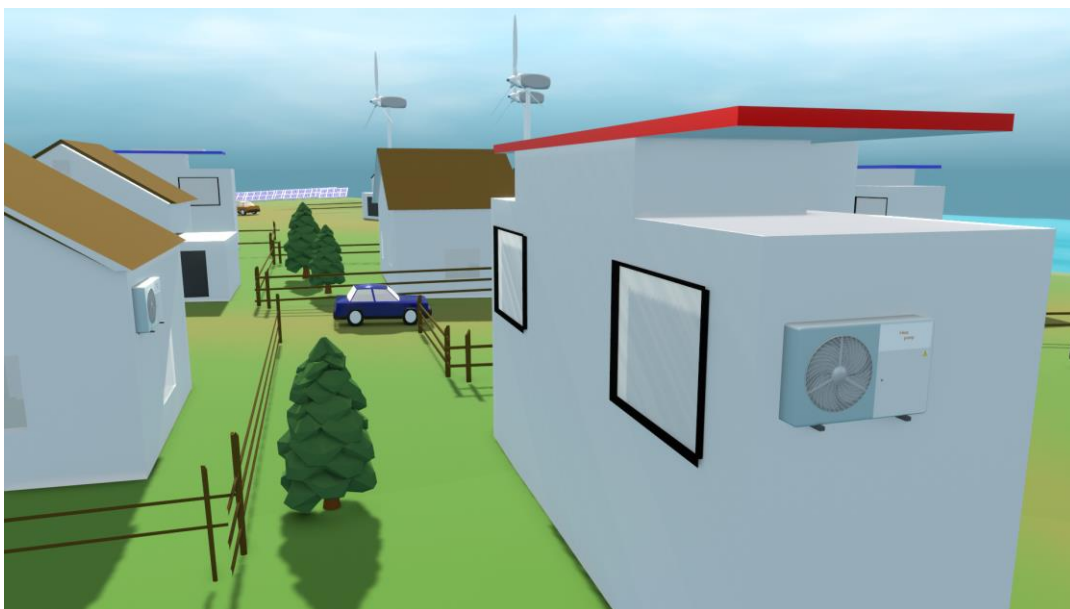


Рис. 2.29. Приклад розміщення змодельованих кондиціонерів на будівелях

3. Транспортні засоби:

– електромобілі (12 шт.) Різні моделі електричних автомобілів, які можуть пересуватися по віртуальному острову (рис. 2.30).



Рис. 2.30 Приклади змодельованих електромобілів

– зарядні станції для електромобілів (8 шт.): об'єкти, які дозволяють імітувати процес зарядки електромобілів (рис. 2.31).

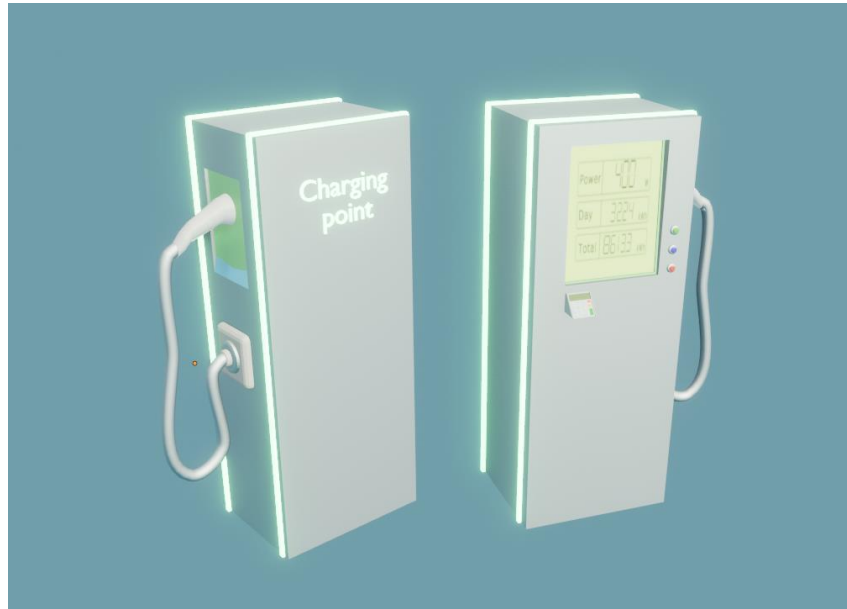


Рис. 2.31. Змодельована зарядна станція

– човни (5 шт.) Моделі човнів, якими можна пересуватися та заряджати за допомогою зарядної електростанції, тому вона розташована недалеко від берегу (рис. 2.32).



Рис. 2.32. Приклад змодельованих човнів поряд берегу з зарядною електростанцією

4. Елементи ландшафту та оточення:

– дерева (66 шт.) різноманітні 3D-моделі дерев для створення реалістичного ландшафту острова (рис. 2.33).



Рис. 2.33. Приклад розташування дерев на острові

– острів з горою (1 шт.) Центральний елемент 3D-сцени, на якому розташовані всі будівлі, об'єкти інфраструктури та ландшафтні елементи. Острів оточений водною поверхнею, яка імітує океан, створюючи враження ізольованості та самодостатності енергетичної системи (рис. 2.34).



Рис. 2.34. Модель острова

— малі архітектурні форми. Лавки (3 шт.), огорожі (20 шт.), зображені на рис. 2.35 та міст (1 шт.), зображений на рис. 2.32 для збагачення віртуального середовища.

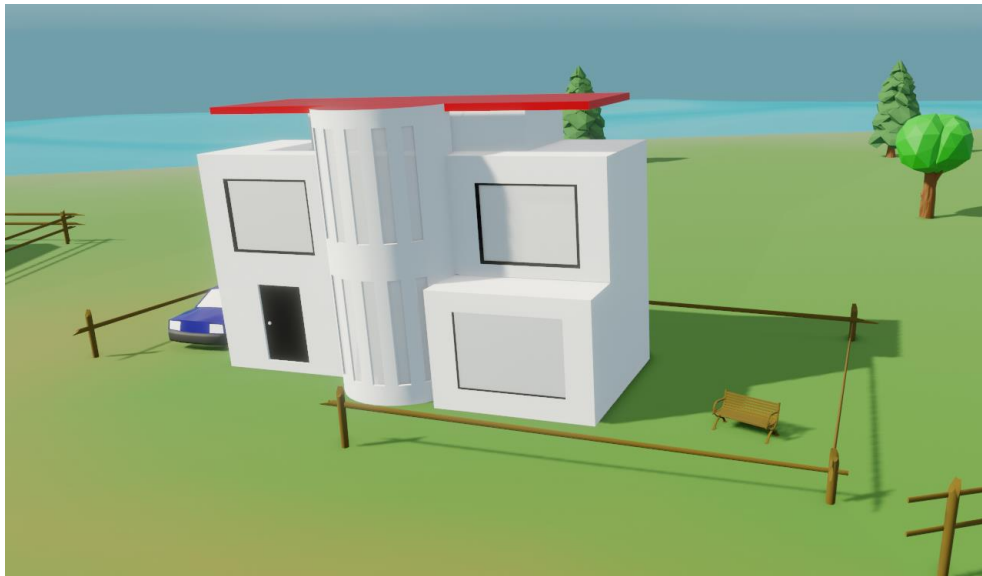


Рис. 2.35. Приклад моделі лавки та огорожі

5. Основа під 3D-сцену з інтерактивними елементами:

- платформа (1 шт.): спеціально розроблена основа, на якій розміщується 3D-модель енергетичного острова. Платформа слугує фундаментом для віртуальної сцени та забезпечує стабільність і зручність взаємодії з моделлю (рис. 2.36);
- кнопки управління (5 шт.): інтерактивні елементи, розташовані на платформі, які дозволяють користувачам взаємодіяти з 3D-моделлю, змінювати параметри, керувати симуляцією тощо (рис. 2.36);
- емблема проекту (1 шт.): графічний елемент, який представляє назву моделі острова «CO₂ нейтральна острівна система» . Емблема розміщена на платформі і слугує візуальним ідентифікатором моделі (рис. 2.36);

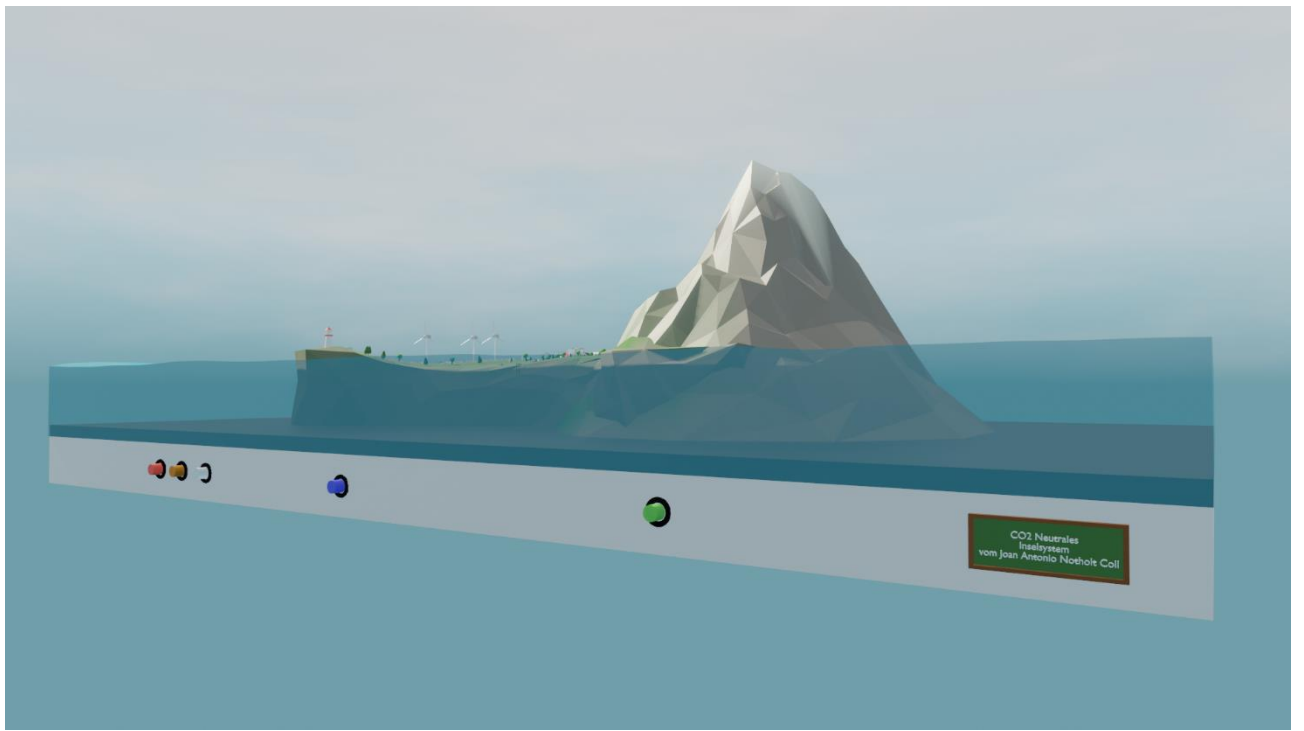


Рис. 2.36. Основа під 3D-сцену з інтерактивними елементами

РОЗДІЛ 3

ЕКОНОМІЧНИЙ РОЗДІЛ

3.1. Розрахунок трудомісткості та вартості розробки програмного продукту

Початкові дані:

1. передбачуване число операторів програми – 1293;
2. коефіцієнт складності програми – 1,8;
3. коефіцієнт корекції програми в ході її розробки – 0,1;
4. годинна заробітна плата програміста – 191,5 грн/год;

Згідно з даними, отриманими з веб-порталу «Українська спільнота програмістів (DOU)» [35], середня місячна заробітна плата 3D Artist з досвідом роботи 1-2 роки у період з червня по грудень 2023 року залишалася незмінною та становила 900 \$/місяць. Враховуючи офіційний курс валют Національного банку України [36] за період розробки проєкту (з 01.09.2023 по 31.05.2024), де середня ціна 1 долара США в середньому дорівнювала 37,447 гривень, що еквівалентно 33 702,3 гривень на місяць. Згідно зі статтею 62 Закону України «Про вищу освіту», студенти вищих навчальних закладів мають право працювати лише у позанавчальний час [37]. З огляду на це, студентів частіше працевлаштовують на умовах неповного робочого часу або неповного робочого тижня (до 88 годин на місяць), щоб забезпечити баланс між навчанням та роботою [38]. Враховуючи стандартний робочий час 176 годин на місяць та обмеження для студентів денної форми навчання, погодинна заробітна плата такого працівника становитиме 191,5 грн/год.

5. коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі – 1,4;

6. коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності – 0,8;

7. вартість машино-години ЕОМ – 0,74 грн/год.

Розрахунок показника ЕОМ було здійснено з урахуванням актуальних умов на період розробки проєкту «Virtual Reality Campus», яка тривала 9 місяців, з 01.09.2023 по 31.05.2024. У цей період тариф на споживання електроенергії становив 2,64 грн за 1 кВт·год [39]. Оскільки модель ноутбука «ROG Strix G17 (2023) G713», використана під час розробки, в середньому споживає 150 Вт·год [25], вартість витрат на електроенергію під час роботи складає: $0,15 \text{ кВт}\cdot\text{год} \times 2,64 \text{ грн/кВт}\cdot\text{год} = 0,4 \text{ грн/год}$.

Крім того, щомісячна абонентна плата за інтернет-пакет «Інтернет gigabit» від провайдера «Vega» у м. Дніпро становить 250 грн/місяць [40]. Враховуючи, що в період з 01.09.2023 по 31.05.2024 середня тривалість місяця складала 30,4 дні, тобто 729,6 годин, вартість користування інтернетом дорівнює: $250 \text{ грн} / 729,6 \text{ год} = 0,34 \text{ грн/год}$. Таким чином, загальна вартість машино-години ЕОМ становить: $0,4 \text{ грн/год} + 0,34 \text{ грн/год} = 0,74 \text{ грн/год}$.

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ можна розрахувати за формулою (3.1) [41]:

$$t = t_o + t_u + t_a + t_n + t_{oml} + t_{\partial}, \text{ людино-годин.} \quad (3.1)$$

де t_o – витрати праці на підготовку й опис поставленої задачі (приймається 50 людино-годин);

t_u – витрати праці на дослідження алгоритму рішення задачі;

t_a – витрати праці на розробку блок-схеми алгоритму;

t_n – витрати праці на програмування по готовій блок-схемі;

t_{oml} – витрати праці на налагодження програми на ЕОМ;

t_{∂} – витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у ПЗ, яке розробляється.

Умовне число операторів (підпрограм) визначається за формулою (3.2) [41]:

$$Q = q \cdot C \cdot (1 + p) \quad (3.2)$$

де q – передбачуване число операторів (1293);

C – коефіцієнт складності програми (1,8);

p – коефіцієнт корекції програми в ході її розробки (0,1).

Розрахуємо, умовне число операторів в програмі за формулою (3.2):

$$Q = 1293 \cdot 1,8 \cdot (1 + 0,1) = 1391,78$$

Витрати праці на вивчення опису задачі t_u обчислюються за формулою (3.3) з урахуванням уточнення опису і кваліфікації програміста [41]:

$$t_u = \frac{Q \cdot B}{(75 \cdot 85) \cdot k}, \text{ людино-годин.} \quad (3.3)$$

де B – коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі (1,4);

k – коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності, і у разі менше двох років (0,8).

$$t_u = \frac{1391,78 \cdot 1,4}{80 \cdot 0,8} = 30,45 \text{ людино-годин.}$$

Витрати праці на розробку алгоритму рішення задачі встановлюються за формулою (3.4) [41]:

$$t_a = \frac{Q}{(20 \dots 25) \cdot k}, \text{ людино-годин.} \quad (3.4)$$

де Q – умовне число операторів (1391,78);

k – коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності (0,8).

Розрахунок:

$$t_a = \frac{1391,78}{24 \cdot 0,8} = 72,49 \text{ людино-годин.}$$

Витрати на складання програми за готовою блок-схемою, також обчислюються за формулою (3.4):

$$t_n = \frac{1391,78}{22 \cdot 0,8} = 79,08 \text{ людино-годин.}$$

Витрати праці на налагодження програми на ЕОМ:

– за умови автономного налагодження одного завдання визначаються за формулою (3.5) [41]:

$$t_{отл} = \frac{Q}{(4.5) \cdot k}, \text{ людино-годин.} \quad (3.5)$$

Розрахунок:

$$t_{отл} = \frac{1391,78}{5 \cdot 0,8} = 347,95 \text{ людино-годин.}$$

– за умови комплексного налагодження завдання обчислюються за формулою (3.6) [41]:

$$t_{отл}^k = 1,5 \cdot t_{отл}, \text{ людино-годин.} \quad (3.6)$$

Розрахунок:

$$t_{отл}^k = 1,5 \cdot 347,95 = 521,93 \text{ людино-годин.}$$

Витрати праці на підготовку документації, розраховуються за формулою (3.7) [41]:

$$t_{\partial} = t_{\partial p} + t_{\partial o}, \text{ людино-годин.} \quad (3.7)$$

де $t_{\partial p}$ – трудомісткість підготовки матеріалів і рукопису;

$t_{\partial o}$ – трудомісткість редагування, печатки й оформлення документації.

Трудомісткість підготовки матеріалів і рукопису потрібно визначити за формулою (3.8) [41]:

$$t_{\partial p} = \frac{Q}{(15..20) \cdot k}, \text{ людино-годин.} \quad (3.8)$$

Розрахунок:

$$t_{\partial p} = \frac{1391,78}{20 \cdot 0,8} = 86,97 \text{ людино-годин.}$$

Трудомісткість редагування, печатки й оформлення документації можна обчислити за формулою (3.9) [41]:

$$t_{\partial o} = 0,75 \cdot t_{\partial p}, \text{ людино-годин.} \quad (3.9)$$

Розрахунок:

$$t_{\partial o} = 0,75 \cdot 86,97 = 65,23 \text{ людино-годин.}$$

Розрахунок праці на підготовку документації, за формулою (3.7):

$$t_{\partial} = 86,97 + 65,23 = 152,2 \text{ людино-годин.}$$

Тож розрахунок трудомісткості розробки програмного забезпечення за формулою (3.1):

$$t = 50 + 30,45 + 72,49 + 79,08 + 347,95 + 152,2 = 732,17 \text{ людино-годин.}$$

3.2. Рахунок витрат на створення програми

Витрати на розробку ПЗ складаються із заробітної плати розробника та вартості машинного часу, який використовується для налагодження програми на ЕОМ та розраховується за формулою (3.10) [41]:

$$K_{ПО} = Z_{ЗП} + Z_{МВ}, \text{ грн.} \quad (3.10)$$

де $Z_{ЗП}$ – витрати на заробітну плату програміста;

$Z_{МВ}$ – витрати машинного часу.

Заробітна плата виконавця визначається за формулою (3.11) [41]:

$$Z_{ЗП} = t \cdot C_{ПР}, \text{ грн.} \quad (3.11)$$

де t – загальна трудомісткість (732,17 людино-годин);

$C_{ПР}$ – середня годинна заробітна плата програміста (191,5 грн/год).

Розрахунок:

$$Z_{ЗП} = 732,17 \cdot 191,5 = 140\,210,56 \text{ грн.}$$

Вартість машинного часу, необхідного для налагодження програми на ЕОМ, обчислюється за формулою (3.12) [41]:

$$Z_{МВ} = t_{отл} \cdot C_{мч}, \text{ грн.} \quad (3.12)$$

де $t_{отл}$ – трудомісткість налагодження програми на ЕОМ (521,93 год);

$C_{мч}$ – вартість машино-години ЕОМ (0,74 грн/год).

Розрахунок:

$$Z_{МВ} = 521,93 \cdot 0,74 = 386,23 \text{ грн.}$$

Розрахунок витрат на створення ПЗ, що обчислюється за формулою (3.10) [41]:

$$K_{ПО} = 140\,210,56 + 386,23 = 140\,596,79 \text{ грн.}$$

Очікуваний період розробки ПЗ розраховується за формулою (3.13) [41]:

$$T = \frac{t}{B_k \cdot F_p}, \text{ міс.} \quad (3.13)$$

де t – загальна трудомісткість (732,17 людино-годин);

B_k – число виконавців (1);

F_p – місячний фонд робочого часу (при 20 годинному робочому тижні

$F_p = 88$ годин).

Розрахунок:

$$T = \frac{732,17}{1 \cdot 88} = 8,3 \text{ міс.}$$

Висновок: для розробки 3D-сцени в Blender для інтеграції у VR-середовище загальна трудомісткість становить 732,17 людино-годин, згідно чого очікуваний період створення програмного забезпечення дорівнює приблизно дещо більше 8 місяців, а витрати на створення – 140 596,79 грн.

ВИСНОВКИ

Відповідно до завдання та мети кваліфікаційної роботи, розроблено деталізовану та оптимізовану 3D-сцену енергетичного острова в Blender для інтеграції у VR-середовище в рамках проєкту «Virtual Reality Campus».

Проаналізувавши дослідження в галузі застосування VR-технологій в освіті, зроблено висновок про їх високу ефективність у порівнянні з традиційними методами навчання. Використання VR забезпечує вищу швидкість засвоєння матеріалу, впевненість у набутих знаннях, емоційний зв'язок із контентом та концентрацію уваги. Тому створення імерсивного віртуального середовища для вивчення відновлюваної енергетики має значний потенціал для покращення якості інженерної освіти.

Практичне значення розробки полягає у створенні реалістичного та інтерактивного навчального простору, який дозволяє студентам отримувати практичний досвід роботи з енергетичними системами в безпечних умовах. 3D-сцена включає детально змодельовані об'єкти інфраструктури острова (будівлі, електромобілі, сонячні панелі, вітрогенератори тощо), які в сукупності формують повноцінне віртуальне середовище для експериментів та аналізу.

Особливу увагу було приділено оптимізації 3D-моделей для ефективної роботи у режимі VR. Застосування низькополігонального моделювання та ретельне налаштування матеріалів і текстур дозволило досягти високої візуальної якості при збереженні продуктивності. Використання принципів золотого перетину та чисел Фібоначчі при плануванні композиції сцени забезпечило її естетичну привабливість та реалістичність.

Розроблена 3D-сцена інтегрується у VR-середовище за допомогою технологій WebXR, Three.js та інструментів збірки проєкту, таких як Webpack. Це дозволяє забезпечити кросплатформний доступ до віртуального кампусу через веб-браузер та VR-пристрої. Система передбачає обробку даних про положення користувача в просторі та параметрів стану віртуального

середовища в режимі реального часу, генерацію відповідного зображення та звуку.

З огляду на висновки в економічному розділі, для розробки 3D-сцени планується задіяти 732,17 людино-годин, очікуваний період становить приблизно 8 місяців за умови неповної зайнятості, а бюджет - 140 596,79 грн.

Стратегія подальшого розвитку проєкту передбачає реалізацію інтерактивності об'єктів сцени, симуляцію енергетичних процесів та інтеграцію з серверною частиною для збереження стану середовища та синхронізації дій користувачів. Це дозволить перетворити розроблену 3D-сцену на повноцінний імерсивний симулятор енергетичної системи острова та розширити спектр навчальних сценаріїв.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Statista Market Insights. Metaverse: market data & analysis. Statista, 2023. URL: <https://www.statista.com/outlook/amo/metaverse> (дата звернення: 02.05.2024).
2. King, N. Immersive Education: Transforming the Landscape of e-Learning. 2024.
3. Chang, E., Kimb, H. T., & Yoo, B. Virtual Reality Sickness: A Review of Causes and Measurements. *International Journal of Human-Computer Interaction*, 2020. 36(17), 1658-1682. DOI: 10.1080/10447318.2020.1778351
4. PwC. The Effectiveness of Virtual Reality Soft Skills Training in the Enterprise. 2020. URL: <https://www.pwc.com/us/vlearning> (дата звернення: 02.05.2024).
5. Tyng, C. M., Amin, H. U., Saad, M. N., & Malik, A. S. The Influences of Emotion on Learning and Memory. *Frontiers in Psychology*, 2017. 8, 1454. DOI: 10.3389/fpsyg.2017.01454
6. Abichandani, P., Fligor, W., & Fromm, E. A Cloud Enabled Virtual Reality Based Pedagogical Ecosystem for Wind Energy Education. 2015 IEEE Frontiers in Education Conference (FIE), 2015. 1-8. DOI: 10.1109/FIE.2015.7344287
7. WebXR Device API. W3C, 2023. URL: <https://www.w3.org/TR/webxr/> (дата звернення: 05.06.2024).
8. Nadal, M., Capó, M., & Cela Conde, C. J. Art and the golden ratio: A review and a proposal. *Psychology of Aesthetics, Creativity, and the Arts*, 2018. 12(4), 386-396. DOI: 10.1037/aca0000187
9. Jensen, M. B., & Konradsen, H. H. A Review of the Use of Virtual Reality Head-Mounted Displays in Education and Training. *Education and Information Technologies*, 2018. 23(4), 1515-1529. DOI: 10.1007/s10639-017-9676-0
10. Google Trends. URL: <https://trends.google.com/> (дата звернення: 15.05.2024).

11. Blender 2.83 Release Notes. Blender Foundation, 2020. URL: <https://www.blender.org/download/releases/2-83/> (дата звернення: 15.05.2024).
12. Why Your Classroom Needs Blender. CG Cookie, 2022. URL: <https://cgcookie.com/posts/why-your-classroom-needs-blender> (дата звернення: 15.05.2024).
13. Stack Overflow Developer Survey 2023. URL: <https://survey.stackoverflow.co/2023/> (дата звернення: 25.05.2024).
14. Usage statistics of JavaScript as client-side programming language on websites. W3Techs, 2023. URL: <https://w3techs.com/technologies/details/cp-javascript> (дата звернення: 25.05.2024).
15. Reina, A., Berto, F., Valdetaro, R., & Anselmetti, M. Implicit Learning Through Embodiment in Immersive Virtual Reality. 2017 IEEE Virtual Reality (VR), 2017. 191-192. DOI: 10.1109/VR.2017.7892258
16. Metaverse: How It's Shaping Up. Statista, 2023. URL: <https://www.statista.com/study/109191/metaverse-how-it-s-shaping-up/> (дата звернення: 02.06.2024).
17. Three.js Documentation. URL: <https://threejs.org/docs/> (дата звернення: 02.06.2024).
18. Slater, M. Immersion and the illusion of presence in virtual reality. British Journal of Psychology, 2018. 109(3), 431-433. DOI: 10.1111/bjop.12305
19. Immersive Web Community Group. URL: <https://immersiveweb.dev/> (дата звернення: 02.06.2024).
20. Makransky, G., Terkildsen, T. S., & Mayer, R. E. Adding immersive virtual reality to a science lab simulation causes more presence but less learning. Learning and Instruction, 2019. 60, 225-236. DOI: 10.1016/j.learninstruc.2017.12.007
21. Webpack Concepts. Webpack, 2023. URL: <https://webpack.js.org/concepts/> (дата звернення: 06.06.2024).
22. Introduction to Node.js. Node.js, 2023. URL: <https://nodejs.org/en/learn/> (дата звернення: 06.06.2024).

23. Gupta, A. Client-Side vs. Server-Side Architecture: What's the Difference? Upwork, 2021. URL: <https://www.upwork.com/resources/client-side-vs-server-side-architecture> (дата звернення: 10.06.2024).
24. How do I set up a local testing server? MDN Web Docs, 2023. URL: https://developer.mozilla.org/en-US/docs/Learn/Common_questions/Tools_and_setup/set_up_a_local_testing_server (дата звернення: 10.06.2024).
25. ROG Strix G17 (2023) G713. ASUS, 2023. URL: <https://rog.asus.com/de/laptops/rog-strix/rog-strix-g17-2023-series/> (дата звернення: 10.06.2024).
26. Razer Abyssus V2 - RZ01-0119 - Support & FAQs. Razer, 2023. URL: https://mysupport.razer.com/app/answers/detail/a_id/3691/~razer-abyssus-v2-%7C-rz01-01190-support-%26-faqs (дата звернення: 10.06.2024).
27. HP Compaq LA1951g 19-inch LCD Monitor. HP Support, 2023. URL: <https://support.hp.com/at-de/product/hp-compaq-la1951g-19-inch-lcd-monitor/3996177> (дата звернення: 10.06.2024).
28. Oculus Go - Our Best VR Optics in Our First Standalone VR Headset. Meta, 2023. URL: <https://www.meta.com/en-gb/blog/quest/oculus-go-available-now-our-best-vr-optics-in-our-first-standalone-vr-headset/> (дата звернення: 10.06.2024).
29. Blender. URL: <https://www.blender.org/> (дата звернення: 15.06.2024).
30. Adobe Photoshop. Adobe, 2023. URL: <https://www.adobe.com/products/photoshop.html> (дата звернення: 15.06.2024).
31. Visual Studio Code. Microsoft, 2023. URL: <https://code.visualstudio.com/> (дата звернення: 15.06.2024).
32. Node.js Downloads. Node.js, 2023. URL: <https://nodejs.org/en/download/> (дата звернення: 15.06.2024).
33. GitLab. URL: <https://about.gitlab.com/> (дата звернення: 15.06.2024).
34. Windows 11 Home. Microsoft, 2023. URL: <https://www.microsoft.com/de-de/d/windows-11-home/dg7gmgf0krt0> (дата звернення: 15.06.2024).

35. Статистика зарплат програмістів, тестувальників і РМ в Україні. DOU, 2023. URL: <https://jobs.dou.ua/salaries/?period=2023-12&position=3D%20Artist&experience=0-2> (дата звернення: 03.06.2024).
36. Офіційний курс гривні щодо іноземних валют. Національний банк України, 2023. URL: <https://bank.gov.ua/ua/markets/exchangerates> (дата звернення: 03.06.2024).
37. Про вищу освіту : Закон України від 01.07.2014 р. № 1556-VII. Ст. 62 // Відомості Верховної Ради України. – 2014. – № 37–38. – Ст. 2004. (зі змінами).
38. Робота студентів: все, що потрібно знати. Трудові ініціативи, 2023. URL: <https://trudovi.org.ua/analytics/roboata-studentiv-vse-shcho-potribno-znaty/> (дата звернення: 03.06.2024).
39. Тарифи на електроенергію з 1.06.2023 по 31.05.2024. Мінфін, 2023. URL: <https://index.minfin.com.ua/ua/tariff/electric/2023-06-01/> (дата звернення: 03.06.2024).
40. Тарифи Інтернет та Інтерактивного ТБ. Vega, 2023. URL: https://vega.ua/for_home/superconnect_fttb (дата звернення: 03.06.2024).
41. Вагонова, О. Г., & Романюк, Н. Н. Факультет менеджменту, кафедра прикладної економіки: методичні вказівки з виконання економічного розділу в дипломних проектах студентів спеціальності «Комп'ютерні системи». Дніпро: НТУ «ДП», 2007.

ЛІСТИНГ ПРОГРАМИ

```
assets.js // Визначає текстури сцени та описує її для відображення у VR
import * as THREE from 'three'

export default {
  // hall
  foxr_tex: {
    url: 'foxr.png',
    options: { encoding: THREE.sRGBEncoding, flipY: false },
  },
  hsrt_tex: {
    url: 'foxr.png',
    options: { encoding: THREE.sRGBEncoding, flipY: false },
  },
  roomname_tex: {
    url: 'foxr.png',
    options: { encoding: THREE.sRGBEncoding, flipY: false },
  },
  roomplate_tex: {
    url: 'Background.png',
    options: { encoding: THREE.sRGBEncoding, flipY: false },
  },
  tecwall_tex: {
    url: 'TECWall.png',
    options: { encoding: THREE.sRGBEncoding, flipY: false },
  },
  tecmain_tex: {
    url: 'TECMain.png',
    options: { encoding: THREE.sRGBEncoding, flipY: false },
  },
  x_wing_tex: {
    url: 'xwing_large.basis',
    options: { encoding: THREE.sRGBEncoding, flipY: false },
  },
  studiengaengeWall_tex: {
    url: 'studiengaengeWall.png',
    options: { encoding: THREE.sRGBEncoding, flipY: false },
  },
  studiWallWhite_tex: {
    url: 'TECStudiWallWhite.png',
    options: { encoding: THREE.sRGBEncoding, flipY: false },
  },
  plane_tex: {
    url: 'plane.png',
    options: { encoding: THREE.sRGBEncoding, flipY: false },
  },
  planeLogo_tex: {
```

```

        url: 'IPPlaneLogo.png',
        options: { encoding: THREE.sRGBEncoding, flipY: false },
    },
    hall_model: { url: 'hall.glb' },
    generic_controller_model: { url: 'generic_controller.glb' },
    lightmap_tex: {
        url: 'lightmap.png',
        options: { encoding: THREE.sRGBEncoding, flipY: false },
    },
    controller_tex: { url: 'controller.basis' },
    doorfx_tex: {
        url: 'doorfx.basis',
        options: { wrapT: THREE.RepeatWrapping, wrapS: THREE.RepeatWrapping },
    },
    sky_tex: {
        url: 'sky.png',
        options: { encoding: THREE.sRGBEncoding, flipY: false },
    },
    clouds_tex: {
        url: 'clouds.basis',
        options: { encoding: THREE.sRGBEncoding, flipY: false },
    },
    teleport_model: { url: 'teleport.glb' },
    beam_tex: { url: 'beamfx.png' },
    glow_tex: { url: 'glow.basis', options: { encoding: THREE.sRGBEncoding } },
    newsticker_tex: {
        url: 'newsticker.jpg',
        options: { encoding: THREE.sRGBEncoding, flipY: false },
    },
    mozillamr_tex: {
        url: 'mozillamr.basis',
        options: { encoding: THREE.sRGBEncoding, flipY: false },
    },
    zoomicon_tex: {
        url: 'zoomicon.png',
        options: { encoding: THREE.sRGBEncoding },
    },
    // panoramas
    panoballfx_tex: {
        url: 'ballfx.basis',
        options: { wrapT: THREE.RepeatWrapping, wrapS: THREE.RepeatWrapping },
    },
    stereopanoL: {
        url: 'stereopanoL.basis',
        options: { encoding: THREE.sRGBEncoding },
    },
    stereopanoR: {
        url: 'stereopanoR.basis',
        options: { encoding: THREE.sRGBEncoding },
    },

```

```

    },
    pano1small: {
        url: 'stereopano_small.basis',
        options: { encoding: THREE.sRGBEncoding },
    },

    pano2: {
        url: 'eldriveslab.basis',
        options: { encoding: THREE.sRGBEncoding, flipY: false },
    },
    pano3: {
        url: 'mechanicslab.basis',
        options: { encoding: THREE.sRGBEncoding, flipY: false },
    },
    pano4: {
        url: 'electronicslab.basis',
        options: { encoding: THREE.sRGBEncoding, flipY: false },
    },
    pano5: {
        url: 'controllab.basis',
        options: { encoding: THREE.sRGBEncoding, flipY: false },
    },
    pano6: {
        url: 'roboerlab.basis',
        options: { encoding: THREE.sRGBEncoding, flipY: false },
    },
    pano2small: {
        url: 'eldriveslab_small.basis',
        options: { encoding: THREE.sRGBEncoding },
    },
    pano3small: {
        url: 'mechanicslab_small.basis',
        options: { encoding: THREE.sRGBEncoding },
    },
    pano4small: {
        url: 'electronicslab_small.basis',
        options: { encoding: THREE.sRGBEncoding },
    },
    pano5small: {
        url: 'controllab_small.basis',
        options: { encoding: THREE.sRGBEncoding },
    },
    pano6small: {
        url: 'roboerlab_small.basis',
        options: { encoding: THREE.sRGBEncoding },
    },

    // graffiti
    spray_model: { url: 'spray.glb' },
    spray_tex: {
        url: 'spray.basis',

```

```

        options: { encoding: THREE.sRGBEncoding, flipY: false },
    },

    // vertigo
    vertigo_model: { url: 'vertigo.glb' },
    vertigo_door_lm_tex: {
        url: 'vertigo_door_lm.jpg',
        options: { encoding: THREE.sRGBEncoding, flipY: false },
    },
    vertigo_lm_tex: {
        url: 'vertigo_lm.basis',
        options: { encoding: THREE.sRGBEncoding, flipY: false },
    },
    checkboard_tex: {
        url: 'checkboard.basis',
        options: {
            wrapT: THREE.RepeatWrapping,
            wrapS: THREE.RepeatWrapping,
            repeat: [4, 4],
        },
    },

    // sound
    sound_model: { url: 'sound.glb' },
    sound_door_model: { url: 'sound_door.glb' },
    sound_shadow_tex: { url: 'sound_shadow.png' },
    sound_door_lm_tex: {
        url: 'sound_door_lm.jpg',
        options: { wrapT: THREE.RepeatWrapping, wrapS: THREE.RepeatWrapping },
    },
    grid_tex: {
        url: 'grid.png',
        options: {
            wrapT: THREE.RepeatWrapping,
            wrapS: THREE.RepeatWrapping,
            repeat: [20, 20],
        },
    },

    // photogrammetry object
    pg_floor_tex: {
        url: 'travertine2.basis',
        options: {
            encoding: THREE.sRGBEncoding,
            flipY: false,
            wrapT: THREE.RepeatWrapping,
            wrapS: THREE.RepeatWrapping,
        },
    },
    pg_floor_lm_tex: {
        url: 'pg_floor_lm.jpg',

```



```

        options: { encoding: THREE.sRGBEncoding, flipY: false },
    },
    re_island_tex: {
        url: 'reisland.png',
        options: { encoding: THREE.sRGBEncoding, flipY: false },
    },
    pg_door_lm_tex: {
        url: 'pg_door_lm.jpg',
        options: { encoding: THREE.sRGBEncoding, flipY: false },
    },
    //pg_object_tex: { url: 'angel.basis', options: { encoding: THREE.sRGBEncoding, flipY:
false} },
    pg_object_model: { url: 'angel.min.glb' }, // TODO: try draco version, angel.min.gl
    pg_bg_tex: {
        url: 'pg_bg.jpg',
        options: { encoding: THREE.sRGBEncoding, flipY: false },
    },
    pg_flare_tex: {
        url: 'flare.jpg',
        options: { encoding: THREE.sRGBEncoding, flipY: false },
    },
    //pg_panel_tex: { url: 'panel.basis', options: { encoding: THREE.sRGBEncoding, flipY:
false} },

    // paintings
    painting_seurat_tex: {
        url: 'paintings/seurat.basis',
        options: { encoding: THREE.sRGBEncoding, flipY: false },
    },
    painting_sorolla_tex: {
        url: 'paintings/sorolla.basis',
        options: { encoding: THREE.sRGBEncoding, flipY: false },
    },
    painting_bosch_tex: {
        url: 'paintings/TEC_Gebaeude.basis',
        options: { encoding: THREE.sRGBEncoding, flipY: false },
    },
    painting_degas_tex: {
        url: 'paintings/degas.basis',
        options: { encoding: THREE.sRGBEncoding, flipY: false },
    },
    painting_rembrandt_tex: {
        url: 'paintings/rembrandt.basis',
        options: { encoding: THREE.sRGBEncoding, flipY: false },
    },

    // sounds
    birds_snd: { url: 'ogg/birds.ogg' },
    chopin_snd: { url: 'ogg/chopin.ogg' },
    forest_snd: { url: 'ogg/forest.ogg' },
    wind_snd: { url: 'ogg/wind.ogg' },

```

```

    teleport_a_snd: { url: 'ogg/teleport_a.ogg' },
    teleport_b_snd: { url: 'ogg/teleport_b.ogg' },
  }

```

PhotogrammetryObject.js // Створює об'єкт для VR, отримуючи його з файлу 3D-сцени та описує його.

```

import * as THREE from 'three'
var scene, doorMaterial, door

function createDoorMaterial(ctx) {
  return new THREE.ShaderMaterial({
    uniforms: {
      time: { value: 0 },
      selected: { value: 0 },
      tex: { value: ctx.assets['doorfx_tex'] },
    },
    vertexShader: ctx.shaders.basic_vert,
    fragmentShader: ctx.shaders.door_frag,
  })
}

export function setup(ctx) {
  const assets = ctx.assets
  scene = assets['pg_object_model'].scene
  scene.rotation.y = -Math.PI / 2

  //scene.getObjectByName('object').material =
  // new THREE.MeshBasicMaterial({map: assets['pg_object_tex']});
  //scene.getObjectByName('x-wing').material =
  // new THREE.MeshBasicMaterial({map: assets['x_wing_tex'], transparent: true});
  //NEW
  scene.getObjectByName('Island').material = new THREE.MeshBasicMaterial({
    map: assets['re_island_tex'],
    transparent: true,
  })
  scene.getObjectByName('floor').material = new THREE.MeshBasicMaterial({
    map: assets['pg_floor_tex'],
    lightMap: assets['pg_floor_lm_tex'],
  })
  scene.getObjectByName('bg').material = new THREE.MeshBasicMaterial({
    map: assets['pg_bg_tex'],
  })
  scene.getObjectByName('flare').material = new THREE.MeshBasicMaterial({
    map: assets['pg_flare_tex'],
    blending: THREE.AdditiveBlending,
  })
  //scene.getObjectByName('panel').material =
  // new THREE.MeshBasicMaterial({map: assets['pg_panel_tex']});
  scene.getObjectByName('door_frame').material = new THREE.MeshBasicMaterial({
    map: assets['pg_door_lm_tex'],
  })

```

```

})

doorMaterial = createDoorMaterial(ctx)
door = scene.getObjectByName('door')
door.material = doorMaterial

scene.getObjectByName('teleport').visible = false

ctx.raycontrol.addState('doorPhotogrammetry', {
  colliderMesh: scene.getObjectByName('door'),
  onHover: (intersection, active) => {
    //teleport.onHover(intersection.point, active);
    const scale = intersection.object.scale
    scale.z = Math.min(scale.z + 0.05 * (2 - door.scale.z), 1.5)
  },
  onHoverLeave: () => {
    //teleport.onHoverLeave();
  },
  onSelectStart: (intersection, e) => {
    ctx.goto = 0
    //teleport.onSelectStart(e);
  },
  onSelectEnd: intersection => {
    //teleport.onSelectEnd(intersection.point);
  },
})

```

```

let teleport = scene.getObjectByName('teleport')
teleport.visible = true
teleport.material.visible = false
ctx.raycontrol.addState('teleportPhotogrammetry', {
  colliderMesh: teleport,
  onHover: (intersection, active) => {
    ctx.teleport.onHover(intersection.point, active)
  },
  onHoverLeave: () => {
    ctx.teleport.onHoverLeave()
  },
  onSelectStart: (intersection, e) => {
    ctx.teleport.onSelectStart(e)
  },
  onSelectEnd: intersection => {
    ctx.teleport.onSelectEnd(intersection.point)
  },
})
}

```

```

export function enter(ctx) {
  ctx.renderer.setClearColor(0x000000)
  ctx.scene.add(scene)
  ctx.raycontrol.activateState('doorPhotogrammetry')
}

```

```

    ctx.raycontrol.activateState('teleportPhotogrammetry')
  }

  export function exit(ctx) {
    ctx.raycontrol.deactivateState('doorPhotogrammetry')
    ctx.raycontrol.deactivateState('teleportPhotogrammetry')

    ctx.scene.remove(scene)
  }

  export function execute(ctx, delta, time) {
    doorMaterial.uniforms.time.value = time

    if (door.scale.z > 0.5) {
      door.scale.z = Math.max(door.scale.z - delta * door.scale.z, 0.5)
    }
  }
}

```

Hall.js // Створює об'єкти для VR-кімнати вітальні, отримуючи їх з файлів 3D-сцен та описує їх.

```

import * as THREE from 'three';
import * as panoballs from '../stations/PanoBalls.js';
import * as paintings from '../stations/Paintings.js';
import * as newsticker from '../stations/NewsTicker.js';
//import * as xylophone from '../stations/Xylophone.js';
import * as graffiti from '../stations/Graffiti.js';
import * as infopanel from '../stations/InfoPanels.js';

var
  scene,
  hall,
  teleportFloor,
  fader,
  doors = [],
  objectMaterials,
  controllers,
  auxVec = new THREE.Vector3();

function createDoorMaterial(ctx) {
  return new THREE.ShaderMaterial({
    uniforms: {
      time: {value: 0},
      selected: {value: 0},
      tex: {value: ctx.assets['doorfx_tex']}
    },
    vertexShader: ctx.shaders.basic_vert,
    fragmentShader: ctx.shaders.door_frag
  });
}

```

```

export function setup(ctx) {
  const assets = ctx.assets;
  scene = new THREE.Object3D();

  // setup hall model

  const hallLightmapTex = assets['lightmap_tex'];
  const skyTex = assets['sky_tex'];
  const cloudsTex = assets['clouds_tex'];
  const foxrTex = assets['foxr_tex'];
  const newstickerTex = assets['newsticker_tex'];
  const mozillamrTex = assets['mozillamr_tex'];
  const hsrtTex = assets['hsrt_tex'];
  const roomnameTex = assets['roomname_tex'];
  const roomplateTex = assets['roomplate_tex'];
  const tecwallTex = assets['tecwall_tex'];
  const tecmainTex = assets['tecmain_tex'];
  const studiengaengeWallTex = assets['studiengaengeWall_tex'];
  const studiWallWhiteTex = assets['studiWallWhite_tex'];
  const planeTex = assets['plane_tex'];
  const planeLogoTex = assets['planeLogo_tex'];

  const hallMaterial = new THREE.MeshBasicMaterial({ map: hallLightmapTex });

  objectMaterials = {
    'hall': hallMaterial,
    'screen': new THREE.MeshBasicMaterial({ map: newstickerTex }),
    'xylophone': hallMaterial,
    'xylostick-left': hallMaterial,
    'xylostick-right': hallMaterial,
    'xylostickball-left': hallMaterial.clone(),
    'xylostickball-right': hallMaterial.clone(),
    'lightpanels': new THREE.MeshBasicMaterial(),
    'rebel': new THREE.MeshBasicMaterial(),
    'doorA': createDoorMaterial(ctx),
    'doorB': createDoorMaterial(ctx),
    'doorC': createDoorMaterial(ctx),
    'doorD': createDoorMaterial(ctx),
    'sky': new THREE.MeshBasicMaterial({ map: skyTex }),
    'clouds': new THREE.MeshBasicMaterial({ map: cloudsTex, transparent: true }),
    'foxr': new THREE.MeshBasicMaterial({ map: foxrTex, transparent: true }),
    'mozillamr': new THREE.MeshBasicMaterial({ map: mozillamrTex, transparent: true }),
    'Background': new THREE.MeshBasicMaterial({ map: roomplateTex, transparent: true }),
    'TECWall': new THREE.MeshBasicMaterial({ map: tecwallTex, transparent: true }),
    'HSRT': new THREE.MeshStandardMaterial({ color: 0x707073 }),
    'VRCampus': new THREE.MeshLambertMaterial({ color: 0x008e82, emissive: 0x008e82,
emissiveIntensity: 0.25 }),
    'TECLogoBlue': new THREE.MeshBasicMaterial({ map: tecmainTex, transparent: true }),
    'Studiengaenge': new THREE.MeshBasicMaterial({ map: studiengaengeWallTex,
transparent: true }),

```

```

    'studiWhite': new THREE.MeshBasicMaterial({map: studiWallWhiteTex, transparent:
true}),
    'studiBlue': new THREE.MeshLambertMaterial({color: 0x0040bd, emissive: 0x0040bd,
emissiveIntensity:0.15}),
    'studiBlack': new THREE.MeshLambertMaterial({color: 0x000000}),
    'Labore': new THREE.MeshLambertMaterial({color: 0x0040bd, emissive: 0x0040bd,
emissiveIntensity:0.6}),
    'plane': new THREE.MeshBasicMaterial({map: planeTex, transparent: true}),
    'IPPlane': new THREE.MeshBasicMaterial({map: planeLogoTex, transparent: true}),
    'SteigAuf': new THREE.MeshLambertMaterial({color: 0x0040bd, emissive: 0x0040bd,
emissiveIntensity:0.6}),
};

```

```

hall = assets['hall_model'].scene;
hall.traverse(o => {
  if (o.name == 'teleport') {
    teleportFloor = o;
    //o.visible = false;
    o.material.visible = false;
    return;
  } else if (o.name.startsWith('door')) {
    doors.push(o);
  }

  if (o.type == 'Mesh' && objectMaterials[o.name]) {
    o.material = objectMaterials[o.name];
  }
});

```

```

paintings.setup(ctx, hall);
//xylophone.setup(ctx, hall);
graffiti.setup(ctx, hall);
newsticker.setup(ctx, hall);
panoballs.setup(ctx, hall);
infopanel.setup(ctx, hall);

```

```

ctx.raycontrol.addState('teleport', {
  colliderMesh: teleportFloor,
  onHover: (intersection, active) => {
    ctx.teleport.onHover(intersection.point, active);
  },
  onHoverLeave: () => {
    ctx.teleport.onHoverLeave();
  },
  onSelectStart: (intersection, e) => {
    ctx.teleport.onSelectStart(e);
  },
  onSelectEnd: (intersection) => {
    ctx.teleport.onSelectEnd(intersection.point);
  }
});

```

```

ctx.raycontrol.addState('doors', {
  colliderMesh: doors,
  onHover: (intersection, active) => {
    const scale = intersection.object.scale;
    scale.z = Math.min(scale.z + 0.05 * (5.5 - scale.z), 5);
  },
  onHoverLeave: (intersection) => {
  },
  onSelectStart: (intersection) => {
    const transitions = {
      doorA: 1,
      doorB: 2,
      doorC: 3,
      doorD: 4
    };
    ctx.goto = transitions[intersection.object.name];
  },
  onSelectEnd: (intersection) => {}
});

// fade camera to black on walls
fader = new THREE.Mesh(
  new THREE.PlaneBufferGeometry(),
  new THREE.MeshBasicMaterial({color: 0x000000, transparent: true, depthTest: false})
);
fader.position.z = -0.1;
fader.material.opacity = 0;

scene.add(hall);
ctx.camera.add(fader);
}

export function enter(ctx) {
  ctx.systemsGroup['roomHall'].play();
  ctx.renderer.setClearColor( 0xC0DFFB );
  ctx.scene.add(scene);

  //xylophone.enter(ctx);
  graffiti.enter(ctx);
  infopanel.enter(ctx);
  ctx.raycontrol.activateState('doors');
  ctx.raycontrol.activateState('teleport');
  paintings.enter(ctx);
  panoballs.enter(ctx);
}

export function exit(ctx) {
  ctx.systemsGroup['roomHall'].stop();
  ctx.scene.remove(scene);
}

```

```

ctx.raycontrol.deactivateState('doors');
ctx.raycontrol.deactivateState('teleport');

//xylophone.exit(ctx);
}

export function execute(ctx, delta, time) {
  panoballs.execute(ctx, delta, time);
  paintings.execute(ctx, delta, time);
  //xylophone.execute(ctx, delta, time, controllers);
  graffiti.execute(ctx, delta, time);
  newsticker.execute(ctx, delta, time);
  infopanel.execute(ctx, delta, time);
  updateUniforms(time);
  //checkCameraBoundaries(ctx);

  for (var i = 0; i < doors.length; i++) {
    if (doors[i].scale.z > 1) {
      doors[i].scale.z = Math.max(doors[i].scale.z - delta * doors[i].scale.z, 1);
    }
  }
}

function updateUniforms(time) {
  objectMaterials.doorA.uniforms.time.value = time;
  objectMaterials.doorB.uniforms.time.value = time;
  objectMaterials.doorC.uniforms.time.value = time;
  objectMaterials.doorD.uniforms.time.value = time;
  objectMaterials.doorD.uniforms.selected.value = 1; //test
  panoballs.updateUniforms(time);
}

function checkCameraBoundaries(ctx) {
  auxVec.copy(ctx.camera.position).add(ctx.cameraRig.position);
  const cam = auxVec;
  const margin = 0.25;
  var fade = 0;
  if (cam.y < margin) { fade = 1 - (cam.y / margin); }
  else if (cam.x < -5.4) { fade = (-cam.x - 5.4) / margin; }
  else if (cam.x > 8) { fade = (cam.x - 8) / margin; }
  else if (cam.z < -6.45) { fade = (-cam.z - 6.45) / margin; }
  else if (cam.z > 6.4) { fade = (cam.z - 6.4) / margin; }
  fader.material.opacity = Math.min(1, Math.max(0, fade));
}

```

Sound.js // Додає ambient-шути для кімнат

```

import * as THREE from 'three';
var scene, listener, timeout, mixer, door, doorMaterial;

```



```

const soundNames = [
  'bells',
  'horn',
  'cowbell',
  'guiro',
  'mandolin',
  'squeaker',
  'train',
  'whistle',
  'motorhorn',
  'surdo',
  'trumpet',
];

var sounds = { };
soundNames.forEach( i => { sounds[i] = { animations: [], mesh: null, player: null, shadow: null} })

const MAX_REPETITIONS = 3;
var repetitions = MAX_REPETITIONS - 1;

function createDoorMaterial(ctx) {
  return new THREE.ShaderMaterial({
    uniforms: {
      time: {value: 0},
      selected: {value: 0},
      tex: {value: ctx.assets['doorfx_tex']}
    },
    vertexShader: ctx.shaders.basic_vert,
    fragmentShader: ctx.shaders.door_frag
  });
}

export function setup(ctx) {
  const assets = ctx.assets;
  scene = assets['sound_model'].scene;
  door = assets['sound_door_model'].scene;

  door.getObjectByName('door_frame').material =
    new THREE.MeshBasicMaterial({map: assets['sound_door_lm_tex']});

  doorMaterial = createDoorMaterial(ctx);
  door.getObjectByName('door').material = doorMaterial;

  door.scale.set(0.5, 0.5, 0.5);
  door.position.set(0.4, 0.6, 1);
  door.rotation.set(0, 0.4, 0);

  listener = new THREE.AudioListener();

  mixer = new THREE.AnimationMixer(scene);

```

```

for (let id in sounds) {
  const mesh = scene.getObjectByName(id);
  if (!mesh) { continue; }

  const sound = new THREE.PositionalAudio(listener);
  const audioLoader = new THREE.AudioLoader();
  audioLoader.load('assets/ogg/' + id + '.ogg', buffer => {
    sound.setBuffer(buffer);
    //sound.setRefDistance(20);
  });

  sounds[id].player = sound;
  sounds[id].mesh = mesh;
  mesh.visible = false;
  mesh.add(sound);

  const clip = THREE.AnimationClip.findByName(assets['sound_model'].animations, id);
  if (clip) {
    const action = mixer.clipAction(clip, mesh);
    action.loop = THREE.LoopOnce;
    sounds[id].animations.push(action);
  }

  for (let j = 0; j < mesh.children.length; j++) {
    const obj = mesh.children[j];
    const clip = THREE.AnimationClip.findByName(assets['sound_model'].animations,
` ${id} _ ${obj.name} `);
    if (!clip) { continue; }
    const action = mixer.clipAction(clip, mesh);
    action.loop = THREE.LoopOnce;
    sounds[id].animations.push(action);
  }

  let shadow = new THREE.Mesh(
    new THREE.PlaneBufferGeometry(3, 3),
    new THREE.MeshBasicMaterial({
      color: mesh.children[0].material.color,
      map: assets['sound_shadow_tex'],
      transparent: true,
      opacity: 0,
      depthTest: false,
      blending: THREE.AdditiveBlending
    })
  );
  shadow.position.set(mesh.position.x, 0.001, mesh.position.z);
  shadow.rotation.x = -Math.PI / 2;
  scene.add(shadow);
  sounds[id].shadow = shadow;
}

ctx.raycontrol.addState('sound', {

```

```

colliderMesh: door.getObjectByName('door'),
onHover: (intersection, active) => {
  //teleport.onHover(intersection.point, active);
  const scale = intersection.object.scale;
  scale.z = Math.min(scale.z + 0.05 * (2 - door.scale.z), 1.5);
},
onHoverLeave: () => {
  //teleport.onHoverLeave();
},
onSelectStart: (intersection, e) => {
  ctx.goto = 0;
  //teleport.onSelectStart(e);
},
onSelectEnd: (intersection) => {
  //teleport.onSelectEnd(intersection.point);
}
});

```

```

const floorTexture = assets['grid_tex'];
const floor = new THREE.Mesh(
  new THREE.PlaneBufferGeometry(20, 20),
  new THREE.MeshBasicMaterial({map: floorTexture})
);
scene.add(floor);
floor.rotation.x = -Math.PI / 2;

```

```

ctx.raycontrol.addState('teleportSound', {
  colliderMesh: floor,
  onHover: (intersection, active) => {
    ctx.teleport.onHover(intersection.point, active);
  },
  onHoverLeave: () => {
    ctx.teleport.onHoverLeave();
  },
  onSelectStart: (intersection, e) => {
    ctx.teleport.onSelectStart(e);
  },
  onSelectEnd: (intersection) => {
    ctx.teleport.onSelectEnd(intersection.point);
  }
});
}

```

```

var currentSound = -1;
function playSound() {
  let sound;
  if (currentSound >= 0) {
    sound = sounds[soundNames[currentSound]];
    sound.player.stop();
    if (sound.animations.length) {
      sound.mesh.visible = false;
      sound.animations.forEach( i => {i.stop()});
    }
  }
}

```

```

    }
  }
  repetitions++;
  if (repetitions == MAX_REPETITIONS) {
    repetitions = 0;
    // get next sound
    do {
      currentSound = (currentSound + 1) % soundNames.length;
      sound = sounds[soundNames[currentSound]];
    } while (!sound.mesh);
  }

  sound.player.play();
  if (sound.animations.length) {
    sound.mesh.visible = true;
    sound.animations.forEach( i => {i.play()});
  }
  sound.shadow.material.opacity = 1;
  timeout = setTimeout(playSound, 2000);
}

export function enter(ctx) {
  ctx.renderer.setClearColor(0x000000);
  ctx.scene.add(scene);
  ctx.scene.add(door);
  ctx.camera.add(listener);

  timeout = setTimeout(playSound, 2000);
  ctx.raycontrol.activateState('teleportSound');
  ctx.raycontrol.activateState('sound');
}

export function exit(ctx) {
  ctx.scene.remove(scene);
  ctx.scene.remove(door);
  ctx.camera.remove(listener);
  ctx.raycontrol.deactivateState('teleportSound');
  ctx.raycontrol.deactivateState('sound');
  clearTimeout(timeout);
}

export function execute(ctx, delta, time) {
  mixer.update(delta);
  const sound = sounds[soundNames[currentSound]];
  if (sound && sound.shadow.material.opacity > 0) {
    sound.shadow.material.opacity -= delta * 0.5;
  }
  doorMaterial.uniforms.time.value = time;

  if (door.scale.z > 0.5) {
    door.scale.z = Math.max(door.scale.z - delta * door.scale.z, 0.5);
  }
}

```

ВІДГУК

**керівника економічного розділу
на кваліфікаційну роботу бакалавра**

на тему:

«Розробка 3D-сцени в Blender для інтеграції у VR-середовище»

студентки групи 122-20-1 Бут Анастасії Сергіївни

Керівник економічного розділу

доц. каф. ПЕП та ПУ, к.е.н

Л.В. Касьяненко

ПЕРЕЛІК ДОКУМЕНТІВ НА ОПТИЧНОМУ НОСІЇ

Ім'я файлу	Опис
Пояснювальні документи	
КваліфікаційнаРобота_Бут.docx	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
КваліфікаційнаРобота_Бут.pdf	Пояснювальна записка до кваліфікаційної роботи в форматі PDF.
Програма	
VirtualRealityCampus.zip	Архів. Містить коди програми і откомпільовану програму.
Презентація	
Презентація_Бут.pptx	Презентація кваліфікаційної роботи.

АВТОРСЬКІ ПРАВА

Національний технічний університет «Дніпровська політехніка» приймає обмежувальні позначки з наступним формулюванням, яке узгоджене Центральною екзаменаційною комісією і може використовуватися без додаткового схвалення керівника з навчальної роботи. Обмеження на розділи або сторінки допускається лише в необов'язковій частині; додаткові текстові доповнення або зміни не дозволяються.

«Ця кваліфікаційна робота базується на внутрішніх та конфіденційних даних компанії Ройтлінгенський університет. Ця кваліфікаційна робота не може бути доступна неуповноваженим третім особам без явної згоди компанії та автора. Розмноження та публікація кваліфікаційної роботи без явного дозволу - навіть у витягах - не дозволяється. Період обмеження закінчується через п'ять років після подання кваліфікаційної роботи <03.07.2029>».

Додаткові угоди про конфіденційність підписуються виключно віце-президентом з навчальної роботи (див. «Пам'ятку про конфіденційність кваліфікаційної роботи»).