

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Національний університет «Острозька академія»**  
**Навчально науковий інститут ІТ та бізнесу**  
**Кафедра інформаційних технологій та аналітики даних**

**КВАЛІФІКАЦІЙНА РОБОТА**

на здобуття освітнього ступеня магістра  
на тему: «Управління проєктом створення візуальної частини  
інтерактивного простору Національного університету "Острозька  
академія"»

**Виконав:** студент 2 курсу, групи МУП2  
другого (магістерського) рівня вищої освіти  
спеціальності 122 Комп'ютерні науки  
ОПП «Управління проєктами»  
*Кошубінський Павло Романович*

**Керівник:** *Місай В.В., викладач, фахівець практик  
кафедри ІТБ*

**Рецензент:** *кандидат технічних наук, доцент, доцент  
кафедри прикладної математики  
Донецького національного університету імені Василя  
Стуса*

*Загоруйко Любов Василівна*

**РОБОТА ДОПУЩЕНА ДО ЗАХИСТУ**

Завідувач кафедри інформаційних технологій та аналітики даних  
\_\_\_\_\_ (проф., д.е.н. Кривицька О.Р.)

Протокол № 5 від 04 грудня 2025 р.

Острог - 2025

**АНОТАЦІЯ**  
**кваліфікаційної роботи**  
**на здобуття освітнього ступеня бакалавра**

**Тема:** *Управління проєктом створення візуальної частини інтерактивного простору Національного університету "Острозька академія"*

**Автор:** *Кошубінський Павло Романович*

**Науковий керівник:** *Місай В.В., викладач кафедри ЕММІТ.*

**Захищена «.....»..... 20\_\_ року.**

**Пояснювальна записка до кваліфікаційної роботи:** *с. 102, 22 рис., 15 табл., 40 джерел.*

**Ключові слова:** *інтерактивність, віртуалізація, екстер'єр, Blender, Unreal Engine 5, моделювання, 3Д редагування, створення, конструювання.*

**Короткий зміст праці:**

*Завданням кваліфікаційної роботи/проєкту, було дослідження та практичне застосування ефективних методологій управління проєктом зі створення візуальної частини інтерактивного простору Національного університету «Острозька академія». У роботі проаналізовано сучасні підходи до управління проєктами, зокрема каскадну модель, Agile-методології (Scrum, Kanban) та гібридні підходи .*

*Обґрунтовано вибір гібридної методології, що поєднує елементи каскадного планування для загальних віх та гнучкість Scrum (двотижневі спринти) для безпосередньої розробки . Практична реалізація охопила повний цикл управління проєктом: від аналізу вимог, структурної декомпозиції робіт та аналізу ризиків до створення 3D-контенту в Blender та його інтеграції у рушій Unreal Engine 5 .*

*Ключовим результатом стало успішне застосування новітніх технологій UE5 – віртуалізованої геометрії **Nanite** та динамічного глобального освітлення **Lumen** , що дозволило досягти високої візуальної якості при збереженні оптимальної продуктивності. Проєкт підтвердив ефективність*

технологічного конвеєра «Blender-to-UE5» , досягнувши цільових показників продуктивності (32 FPS на мінімальних конфігураціях) та перевиконавши план за кількістю створених ассетів та елементів інтерактивності.

**ABSTRACT**  
**of the qualification work**  
**for obtaining a master's degree**

*The task of the qualification work/project was to research and practically apply effective project management methodologies for creating the visual part of the interactive space of the National University of "Ostroh Academy." The paper analyzes modern approaches to project management, including the waterfall model, Agile methodologies (Scrum, Kanban), and hybrid approaches .*

*The choice of a hybrid methodology is substantiated, combining elements of waterfall planning for general milestones and the flexibility of Scrum (two-week sprints) for direct development . The practical implementation covered the full project management cycle: from requirements analysis, work breakdown structure, and risk analysis to creating 3D content in Blender and integrating it into the Unreal Engine 5 engine .*

*A key result was the successful application of the latest UE5 technologies – **Nanite** virtualized geometry and **Lumen** dynamic global illumination , which allowed for high visual quality while maintaining optimal performance. The project confirmed the effectiveness of the "Blender-to-UE5" pipeline, achieving target performance indicators (32 FPS on minimum configurations) and exceeding the plan for the number of created assets and interactive elements.*

## ЗМІСТ

<b>ЗМІСТ .....</b>	<b>4</b>
<b>ВСТУП.....</b>	<b>5</b>
<b>РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ЗАСАДИ УПРАВЛІННЯ ПРОЄКТАМИ СТВОРЕННЯ ІНТЕРАКТИВНИХ ВІРТУАЛЬНИХ ПРОСТОРІВ.....</b>	<b>9</b>
1.1. Сучасні теорії та методології управління проєктами .....	9
1.2. Специфіка управління проєктами інтерактивної візуалізації .....	15
1.3. Сучасні тенденції та інструменти архітектурної візуалізації.....	22
<b>Висновки до розділу 1.....</b>	<b>28</b>
<b>РОЗДІЛ 2. УПРАВЛІННЯ ПРОЄКТОМ СТВОРЕННЯ ВІЗУАЛЬНОЇ ЧАСТИНИ ІНТЕРАКТИВНОГО ПРОСТОРУ НаУОА .....</b>	<b>30</b>
2.1. Аналіз та планування проєкту .....	30
2.2. Реалізація проєкту: створення 3D-контенту .....	42
2.3. Інтеграція та візуалізація в Unreal Engine 5 .....	47
2.4. Контроль, оптимізація та оцінка ефективності проєкту .....	52
<b>Висновки до розділу 2.....</b>	<b>60</b>
<b>РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА АПРОБАЦІЯ РЕЗУЛЬТАТІВ ПРОЄКТУ .....</b>	<b>62</b>
3.1. Експериментальна перевірка обраних методологій управління проєктом.....	62
3.2. Аналіз ефективності використання інструментів Blender та Unreal Engine 5.....	80
3.3. Оцінка досягнення цілей проєкту та рекомендації щодо вдосконалення.....	86
<b>Висновки до розділу 3.....</b>	<b>90</b>
<b>ВИСНОВКИ .....</b>	<b>92</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....</b>	<b>94</b>
<b>ДОДАТКИ.....</b>	<b>99</b>

## ВСТУП

Сучасна архітектурна візуалізація переживає радикальну трансформацію, здійснюючи перехід від статичних фотореалістичних зображень до повноцінних інтерактивних цифрових середовищ. Цей зсув зумовлений стрімким зростанням потужності графічних процесорів, розвитком ігрових рушіїв (зокрема, Unreal Engine 5), появою технологій віртуальної реальності та зміною очікувань клієнтів, які прагнуть не просто бачити, а й "відчувати" простір. В освітньому середовищі це створює унікальні можливості для створення інноваційних презентаційних та навчальних інструментів. Створення презентабельного інтерактивного простору Національного університету «Острозька академія», який би відтворював автентичність історичних будівель з високою візуальною якістю та оптимальною продуктивністю, є актуальним та складним завданням, що потребує ефективного управління проектом на перетині технологій та мистецтва.

Метою кваліфікаційної роботи є управління проектом створення візуальної частини інтерактивного простору Національного університету «Острозька академія» та аналіз ефективності обраних гібридних методологій і технологічного стеку (Blender та Unreal Engine 5) для вирішення поставлених завдань.

Для досягнення поставленої мети необхідно було вирішити наступні завдання:

1. проаналізувати сучасні теорії та методології управління проектами, включаючи каскадну, Agile (Scrum, Kanban) та гібридні моделі;
2. дослідити специфіку управління проектами інтерактивної візуалізації, враховуючи їхню подвійну (технічну та креативну) природу;
3. провести огляд сучасних тенденцій та інструментів архітектурної візуалізації, зокрема Blender та Unreal Engine 5 з технологіями Nanite та Lumen;
4. розробити план управління проектом, включаючи структурну декомпозицію робіт, календарний план та матрицю ризиків;

5. реалізувати створення 3D-контенту (архітектурних елементів, екстер'єрів, інтер'єрів, меблів) за допомогою програмного забезпечення Blender;
6. інтегрувати створені ассети у рушій Unreal Engine 5, налаштувати реалістичне освітлення (Lumen) та оптимізувати геометрію (Nanite);
7. провести контроль якості, тестування продуктивності та оптимізацію проєкту для досягнення цільових показників FPS;
8. провести експериментальну перевірку обраних методологій управління та оцінити ефективність використання інструментів.

Об'єктом дослідження є процес управління проєктами зі створення інтерактивних віртуальних просторів у сфері архітектурної візуалізації.

Предметом дослідження є гібридні методології управління проєктами (зокрема, поєднання Scrum та каскадного підходу), а також інструменти (Blender, Unreal Engine 5) і технічні процеси (3D-моделювання, оптимізація, інтеграція Nanite та Lumen) для створення візуальної частини інтерактивного простору НаУОА.

Практичне значення полягає у створеному інтерактивному візуальному просторі («цифровому двійнику») Національного університету «Острозька академія». Цей програмний продукт може бути використаний Інформаційно-технічним центром НаУОА як інноваційний інструмент для презентаційних, освітніх та навігаційних цілей.

Теоретичне значення полягає в аналізі та адаптації гібридних методологій управління проєктами до специфічних потреб проєктів інтерактивної візуалізації, які поєднують технічні та креативні аспекти. Робота обґрунтовує та експериментально підтверджує ефективність поєднання Agile-практик з сучасним технологічним стеком (Blender та Unreal Engine 5) для даного типу проєктів.

Гіпотеза полягає в тому, що застосування гібридної методології управління (що поєднує гнучкість Scrum для виконання та елементи каскадної моделі для загального планування) у поєднанні з сучасним безкоштовним інструментарієм (Blender) та потужним рушієм реального часу (Unreal Engine 5) є оптимальним

підходом для реалізації проєкту інтерактивної візуалізації в умовах освітнього закладу. Такий підхід дозволяє досягти балансу між високою візуальною якістю, продуктивністю та ефективним управлінням ресурсами і змінами.

Наукова новизна полягає в обґрунтуванні та практичній апробації ефективного та економічно виправданого конвеєра (pipeline) «Blender-to-UE5» для проєктів архітектурної візуалізації. У роботі детально проаналізовано управління проєктом, що використовує новітні технології Unreal Engine 5 — віртуалізовану геометрію Nanite та динамічне глобальне освітлення Lumen. Доведено їхню перевагу над традиційними методами, що вимагають «запікання» світла та ручного створення рівнів деталізації (LODs), для досягнення фотореалізму в реальному часі.

Для досягнення мети були використані наступні методи:

1. теоретичний аналіз та синтез (для огляду методологій управління проєктами та специфіки креативно-технічних проєктів );
2. порівняльний аналіз (для оцінки переваг та недоліків інструментів архітектурної візуалізації );
3. методи проєктного менеджменту (структурна декомпозиція робіт (WBS) , календарне планування , аналіз ризиків , Agile-практики Scrum );
4. емпіричні методи (експериментальна перевірка методологій , тестування продуктивності на різних конфігураціях обладнання );
5. статистичний аналіз (для оцінки показників продуктивності спринтів та порівняння планових і фактичних показників проєкту ).

Структура роботи. Кваліфікаційна робота складається зі вступу, трьох розділів, висновків та списку використаних джерел. У першому розділі проаналізовано теоретико-методологічні засади управління проєктами, детально розглянуто специфіку проєктів інтерактивної візуалізації та сучасні тенденції й інструменти. У другому розділі описано практичний процес управління проєктом: аналіз вимог та планування , детальний опис створення 3D-контенту в Blender , процес інтеграції та візуалізації в Unreal Engine 5 , а також методи контролю, оптимізації та оцінки продуктивності. У третьому розділі проведено

експериментальну перевірку ефективності обраних методологій управління (зокрема, за допомогою Trello та Jira) , виконано поглиблений аналіз ефективності синергії інструментів Blender та Unreal Engine 5 та надано фінальну оцінку досягнення цілей проєкту з рекомендаціями щодо вдосконалення.

# РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ЗАСАДИ УПРАВЛІННЯ ПРОЄКТАМИ СТВОРЕННЯ ІНТЕРАКТИВНИХ ВІРТУАЛЬНИХ ПРОСТОРИВ

## 1.1. Сучасні теорії та методології управління проєктами

Управління проєктами як окрема галузь знань пройшла тривалий шлях еволюції, трансформуючись від простих методів планування до складних інтегрованих систем координації ресурсів, часу та якості. Сучасне розуміння управління проєктами базується на фундаментальному визначенні, запропонованому Інститутом управління проєктами (PMI): це застосування знань, навичок, інструментів та методів до проєктної діяльності з метою задоволення вимог проєкту. Проте, з розвитком технологій та ускладненням бізнес-середовища, це визначення постійно розширюється, охоплюючи нові аспекти гнучкості, адаптивності та створення цінності для стейкхолдерів [1].

Історично першою систематизованою методологією управління проєктами стала каскадна модель, відома також як Waterfall. Ця методологія виникла у 1970-х роках у контексті розробки програмного забезпечення та швидко поширилася на інші галузі завдяки своїй структурованості та передбачуваності. Каскадна модель являє собою послідовний підхід до управління проєктами, де кожна фаза розробки повинна бути повністю завершена перед переходом до наступної.

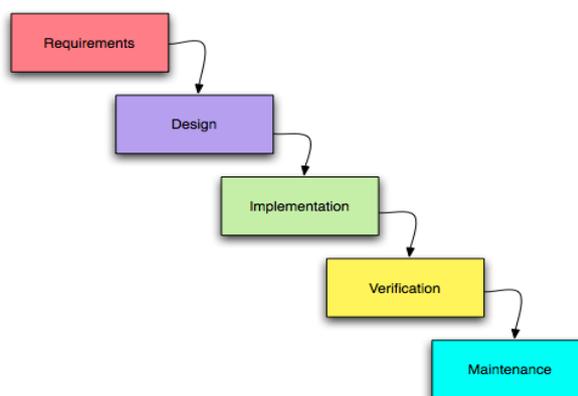


Рис. 1.1. Каскадна модель Waterfall

*Джерело: створено автором*

Фундаментальною характеристикою каскадної моделі є її лінійність та незворотність процесу. Після завершення кожного етапу створюється детальна документація, яка слугує основою для наступного етапу, а повернення до попередніх фаз вважається небажаним та коштовним. Такий підхід забезпечує високий рівень контролю та передбачуваності, оскільки всі вимоги визначаються на початку проєкту, а бюджет та терміни плануються з високою точністю. Для проєктів з чітко визначеними вимогами, стабільним технологічним середовищем та незмінними цілями Waterfall залишається ефективним рішенням, особливо в галузях будівництва, виробництва та регульованих індустріях [2].

Однак жорсткість каскадної моделі виявилася суттєвим недоліком у швидкозмінному бізнес-середовищі сучасності. Неможливість швидко реагувати на зміни вимог, висока вартість виправлення помилок на пізніх етапах та відсутність проміжних результатів, які можна було б продемонструвати замовнику, призвели до пошуку альтернативних підходів. Особливо гостро ці проблеми проявилися у сфері розробки програмного забезпечення, де вимоги часто змінюються в процесі розробки, а ринкові умови потребують швидкого виведення продукту.

У відповідь на ці виклики на початку 2000-х років оформилася концепція гнучких методологій, об'єднаних під загальною назвою Agile. У 2001 році група розробників програмного забезпечення сформулювала Маніфест гнучкої розробки програмного забезпечення (Agile Manifesto), який визначив чотири ключові цінності: люди та взаємодія важливіші за процеси та інструменти; працюючий продукт важливіший за вичерпну документацію; співпраця із замовником важливіша за узгодження умов контракту; готовність до змін важливіша за дотримання плану [3]. Ці принципи стали фундаментом для розвитку численних гнучких методологій, які радикально змінили підхід до управління проєктами.

Agile як парадигма управління проєктами характеризується ітеративністю та інкрементальністю розробки. Замість створення всього продукту за один тривалий цикл, робота розбивається на короткі ітерації тривалістю від одного до

чотирьох тижнів, кожна з яких завершується створенням потенційно готового до використання інкременту продукту.

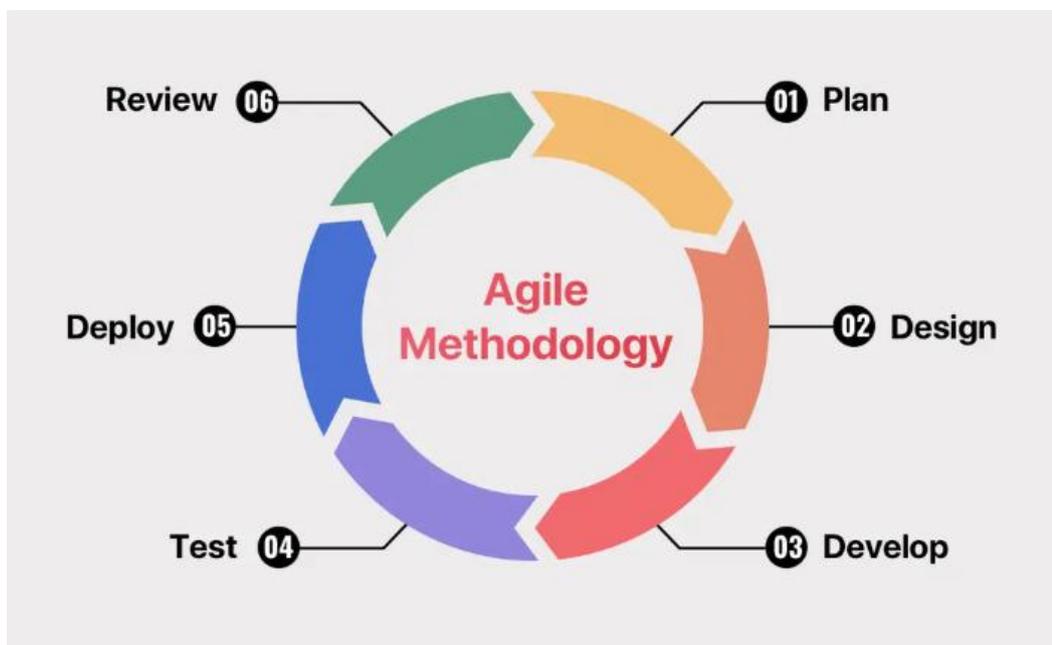


Рис. 1.2. Методологія Agile

*Джерело: створено автором*

Такий підхід дозволяє регулярно отримувати зворотний зв'язок від замовників та користувачів, швидко адаптуватися до змін вимог та мінімізувати ризики, пов'язані з неправильним розумінням потреб клієнта. Гнучкість Agile досягається завдяки акценту на безперервному плануванні, постійній комунікації в команді та готовності переглядати пріоритети на основі нової інформації.

Серед конкретних реалізацій Agile-філософії особливе місце посідає фреймворк Scrum, який став найпопулярнішою методологією гнучкого управління проектами. Scrum визначає чіткі ролі, події та артефакти, що забезпечують структурований підхід до ітеративної розробки. Ключовими ролями в Scrum є Product Owner, який відповідає за визначення та пріоритизацію вимог; Scrum Master, що забезпечує дотримання процесу та усуває перешкоди; та Development Team [4], яка безпосередньо створює продукт. Робота організована у спринти фіксованої тривалості, зазвичай два тижні, протягом яких

команда реалізує найбільш пріоритетні елементи з Product Backlog – впорядкованого списку вимог до продукту.

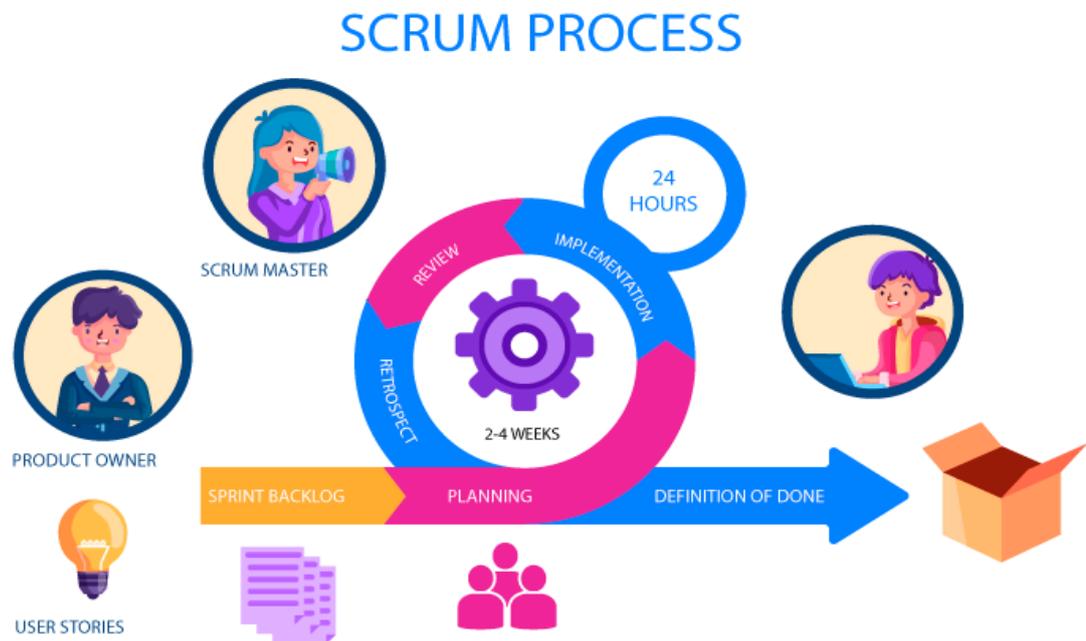


Рис. 1.3. Принцип роботи фреймворка Scrum

*Джерело: створено автором*

Структура Scrum включає п'ять ключових подій, які забезпечують ритм роботи та можливості для інспекції та адаптації. Sprint Planning відкриває спринт і використовується для визначення цілей спринту та відбору завдань з Product Backlog. Daily Scrum – щоденна 15-хвилинна зустріч для синхронізації роботи команди. Sprint Review проводиться наприкінці спринту для демонстрації результатів стейкхолдерам та отримання зворотного зв'язку. Sprint Retrospective дозволяє команді проаналізувати власний процес роботи та визначити можливості для покращення. Ця чітка структура подій забезпечує прозорість роботи, регулярні можливості для корекції курсу та безперервне вдосконалення процесу [5].

Альтернативним гнучким підходом є методологія Kanban, яка фокусується на візуалізації робочого процесу та оптимізації потоку створення цінності. На відміну від Scrum з його фіксованими спринтами, Kanban є більш гнучкою системою, яка не вимагає радикальної зміни існуючих процесів, а пропонує

еволюційний шлях покращення. Центральним елементом Kanban є дошка з колонками, що відображають етапи робочого процесу, та картками, що представляють окремі робочі елементи [6]. Ця візуалізація робить прозорим стан кожного завдання та дозволяє швидко ідентифікувати проблеми та вузькі місця в процесі.

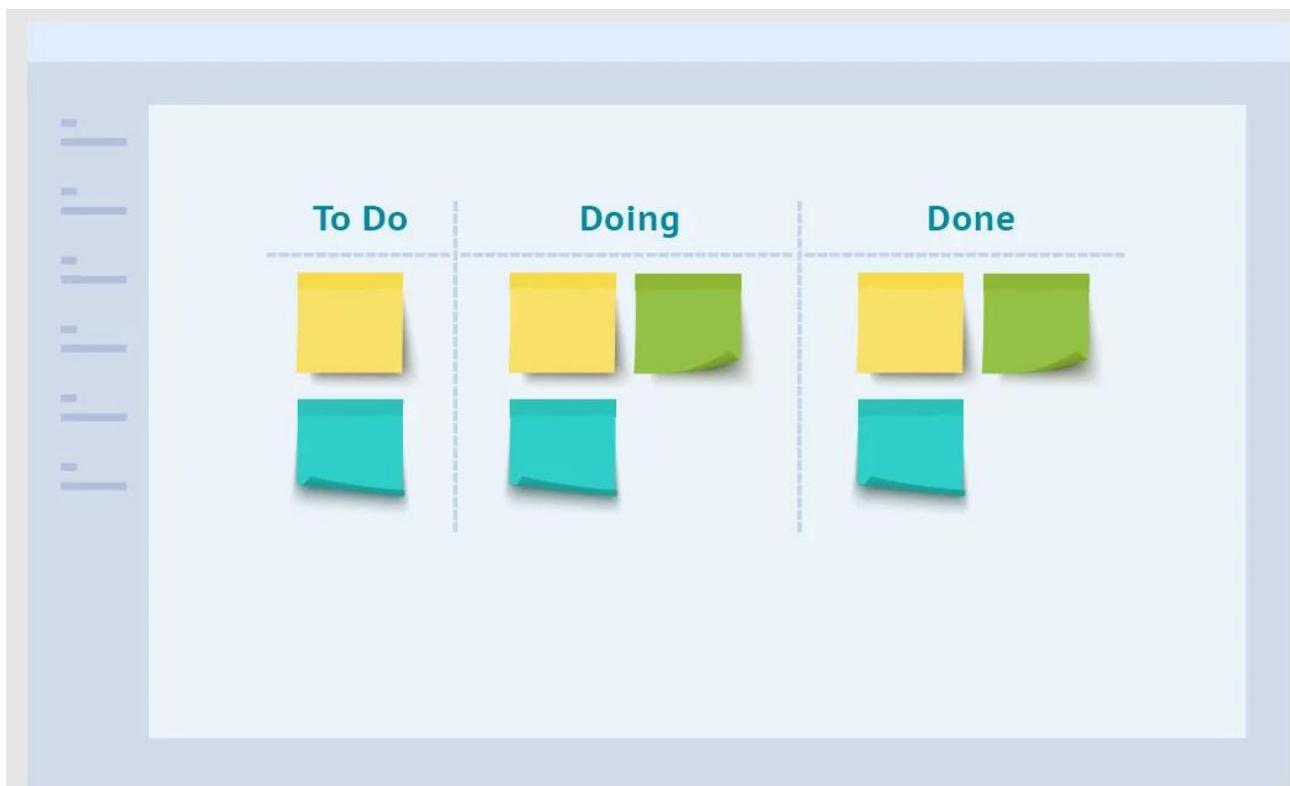


Рис. 1.4. Гнучкий підхід методології Kanban

*Джерело: створено автором*

Ключовим принципом Kanban є обмеження роботи в процесі (Work In Progress limits), що запобігає перевантаженню команди та забезпечує фокус на завершенні розпочатих завдань перед початком нових. Цей підхід базується на теорії обмежень та lean-мисленні, які підкреслюють важливість оптимізації потоку та усунення марнотратства [7]. Kanban також акцентує увагу на метриках продуктивності, таких як час циклу (cycle time) та пропускна здатність (throughput), які дозволяють об'єктивно оцінювати ефективність процесу та приймати обґрунтовані рішення щодо його покращення. Ця методологія

особливо ефективна для операційної діяльності з безперервним потоком завдань, такої як підтримка програмного забезпечення або обробка запитів клієнтів.

Попри численні переваги гнучких методологій, практика показала, що їх застосування у чистому вигляді не завжди є оптимальним рішенням. Багато організацій стикаються з ситуаціями, коли частина проєкту має чітко визначені вимоги та потребує передбачуваності каскадної моделі, тоді як інша частина характеризується високою невизначеністю та вимагає гнучкості Agile. Це призвело до появи гібридних підходів до управління проєктами, які поєднують елементи традиційних та гнучких методологій для досягнення оптимального балансу між структурованістю та адаптивністю [8].

Гібридні методології можуть приймати різні форми залежно від специфіки проєкту та організаційного контексту. Один з поширених підходів передбачає використання каскадної моделі для загального планування проєкту, визначення основних віх та управління бюджетом, тоді як безпосередня розробка ведеться за гнучкими принципами у серії спринтів. Інший варіант – використання Waterfall для стабільних, добре зрозумілих компонентів системи та Agile для інноваційних або складних елементів з високим рівнем невизначеності. Такий підхід дозволяє організаціям використовувати сильні сторони обох парадигм, мінімізуючи їхні недоліки та адаптуючи методологію до конкретних потреб проєкту.

Порівняльний аналіз методологій управління проєктами виявляє їхні унікальні сильні та слабкі сторони, що визначають контекст їх оптимального застосування. Каскадна модель демонструє найкращі результати у проєктах з чіткими, незмінними вимогами, де важлива детальна документація та передбачуваність результатів. Її перевагами є простота планування та контролю, чітке розуміння термінів та бюджету, а також легкість у розподілі відповідальності. Однак жорсткість структури, високі витрати на зміни та пізнє виявлення проблем роблять Waterfall неефективним у динамічному середовищі з мінливими вимогами.

Гнучкі методології, навпаки, виявляють свою силу саме там, де Waterfall слабкий – у проєктах з високою невизначеністю, складними вимогами та необхідністю швидкої адаптації до змін. Agile, Scrum та Kanban забезпечують швидке отримання цінності для клієнта, ефективне управління ризиками через ранню валідацію гіпотез та високу залученість команди завдяки самоорганізації та регулярним можливостям для покращення. Проте ці методології вимагають високого рівня зрілості команди, активної участі замовника та можуть створювати виклики у довгостроковому плануванні та прогнозуванні бюджету, що робить їх застосування складним у великих корпоративних структурах з жорсткими процедурами та контролем [9].

Гібридні підходи намагаються об'єднати переваги обох світів, пропонуючи найбільшу гнучкість у виборі інструментів та методів управління проєктом. Вони дозволяють організаціям поступово переходити від традиційних до гнучких практик, адаптуючи методологію під специфіку конкретного проєкту та організаційну культуру. Успіх гібридного підходу залежить від глибокого розуміння як традиційних, так і гнучких методологій, здатності об'єктивно оцінювати контекст проєкту та вміння знаходити оптимальний баланс між структурою та гнучкістю.

## **1.2. Специфіка управління проєктами інтерактивної візуалізації**

Проєкти створення інтерактивних візуальних середовищ представляють собою унікальний клас проєктів, що поєднують характеристики технічної розробки, художньої творчості та управління складними системами. На відміну від традиційних проєктів розробки програмного забезпечення або будівництва, проєкти інтерактивної візуалізації мають подвійну природу, яка вимагає від менеджера проєкту глибокого розуміння як технічних, так і креативних аспектів роботи. Ця специфіка формує унікальні виклики в плануванні, організації та контролі проєктної діяльності, що потребує адаптації традиційних методологій управління проєктами до особливостей креативно-технічної сфери.

Подвійна природа проєктів інтерактивної візуалізації проявляється у постійній необхідності балансувати між мистецтвом та інженерією, між естетичною досконалістю та технічною реалізованістю. З одного боку, створення візуально привабливого та емоційно вражаючого середовища вимагає креативної свободи, експериментування з формами, кольорами, освітленням та композицією. З іншого боку, реалізація цих художніх ідей стикається з жорсткими технічними обмеженнями: продуктивністю апаратного забезпечення, обсягом доступної пам'яті, швидкістю рендерингу та складністю геометрії [10]. Менеджер проєкту повинен володіти достатньою експертизою в обох сферах, щоб ефективно координувати роботу художників та технічних фахівців, забезпечуючи досягнення як естетичних, так і технічних цілей проєкту.

Управління креативним процесом у проєктах інтерактивної візуалізації принципово відрізняється від управління традиційними технічними проєктами. Креативність за своєю природою є нелінійним процесом, що потребує часу на експериментування, ітерації та пошук оптимальних рішень. Неможливо точно спрогнозувати, скільки часу знадобиться художнику для створення переконливої концепції або 3D-моделювачу для досягнення бажаного рівня деталізації без втрати продуктивності. Тому традиційні методи планування з детальними графіками та чіткими дедлайнами для кожного завдання виявляються неефективними. Замість цього необхідно застосовувати більш гнучкі підходи, які дозволяють творчим фахівцям мати простір для експериментів, одночасно забезпечуючи досягнення загальних цілей проєкту в обумовлені терміни.

Ітеративний характер створення візуального контенту є однією з ключових особливостей, що визначає специфіку управління такими проєктами. Процес розробки 3D-асету не є послідовним проходженням через фіксований набір етапів, а радше циклічним процесом постійного удосконалення та корекції. Модель може пройти через етапи блокування базових форм, високополігонального моделювання для деталізації, створення низькополігональної версії для оптимізації, розгортання UV-координат, текстурування, інтеграції в ігровий рушій, налаштування матеріалів, тестування

в контексті сцени та неодноразового повернення до попередніх етапів для корекції [11]. Кожна ітерація може виявляти проблеми, які були непомітні на попередніх етапах, що вимагає гнучкості в плануванні та готовності до змін.

Управління такими ітеративними процесами потребує особливого підходу до визначення критеріїв готовності та прийняття результатів роботи. На відміну від традиційної розробки програмного забезпечення, де функціональність може бути чітко перевірена автоматизованими тестами, якість візуального контенту часто залежить від суб'єктивної оцінки. Що саме означає, що модель "готова"? Які критерії визначають достатній рівень деталізації? Коли текстура досягла необхідної якості? Ці питання не мають однозначних відповідей та вимагають від менеджера проєкту встановлення чітких, але гнучких стандартів якості, які б враховували як художні, так і технічні аспекти роботи [12].

Високий рівень технічної невизначеності є характерною рисою проєктів інтерактивної візуалізації, особливо при використанні новітніх технологій та інструментів. Інструменти на кшталт Unreal Engine 5 з його революційними технологіями Nanite та Lumen пропонують безпрецедентні можливості для створення високоякісної графіки в реальному часі, проте ці технології є відносно новими та можуть містити недокументовані особливості, помилки або несподівані обмеження.

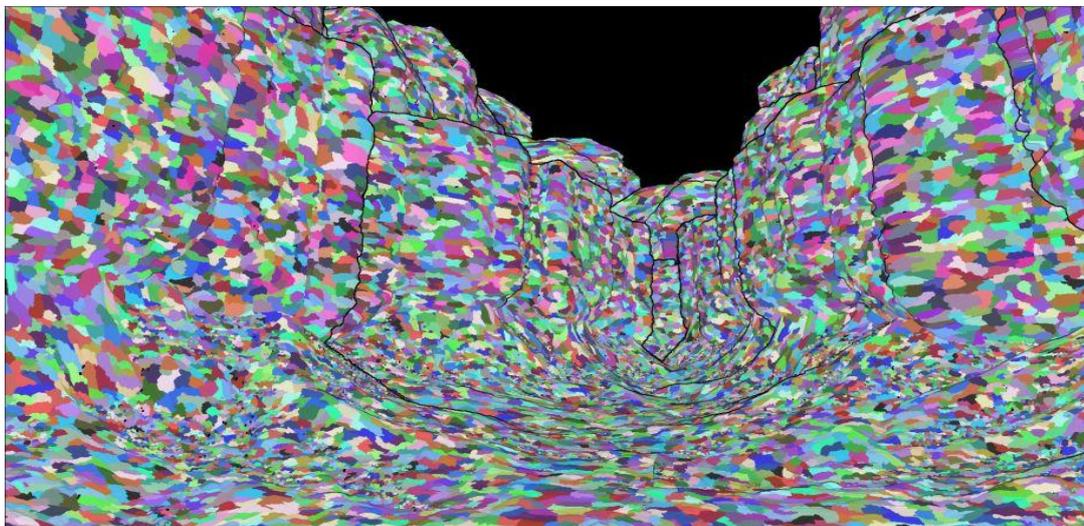


Рис. 1.5. Техношлогія Nanite

*Джерело: створено автором*

Інтеграція різних програмних пакетів, таких як Blender та Unreal Engine, може призводити до технічних проблем, пов'язаних з експортом та імпортом моделей, конфліктами систем координат або втратою даних при конвертації форматів [13].

Ці технічні ризики вимагають від менеджера проєкту обов'язкового закладання часу на дослідження та розробку, відомого як R&D phase.



Рис. 1.5. План R&D

*Джерело: створено автором*

На цьому етапі команда має можливість експериментувати з новими технологіями, виявляти потенційні проблеми та розробляти технічні рішення до початку основного виробництва контенту. Ігнорування цього етапу або недостатнє виділення ресурсів на R&D може призвести до серйозних проблем на пізніх стадіях проєкту, коли вартість виправлення помилок або зміни технічного рішення стає надзвичайно високою. Досвідчені менеджери проєктів зазвичай закладають від десяти до двадцяти відсотків загального часу проєкту на дослідницьку діяльність [14].

Прототипування відіграє критично важливу роль у проєктах інтерактивної візуалізації, набагато більш значущу, ніж у багатьох інших типах проєктів.

Створення швидких прототипів з використанням примітивних об'єктів та базових текстур, відоме як greiboxing або blockout, дозволяє на ранній стадії перевірити ключові аспекти проєкту: масштаб середовища, навігацію користувача, загальну композицію сцени та базові механіки взаємодії. Цей етап надзвичайно важливий, оскільки він дозволяє виявити фундаментальні проблеми дизайну або планування до того, як команда інвестує значні ресурси в створення високоякісних фінальних ассетів.

Процес прототипування також служить важливим інструментом комунікації з замовником та іншими стейкхолдерами проєкту. Навіть найдетальніша письмова специфікація або концепт-арт не можуть повністю передати відчуття простору, масштабу та взаємодії, яке надає інтерактивний прототип. Можливість фізично "пройтися" віртуальним середовищем, навіть у його спрощеній версії, дозволяє замовнику краще зрозуміти майбутній продукт та надати більш конкретний та релевантний зворотний зв'язок. Це значно зменшує ризик непорозумінь між командою розробки та замовником, які часто призводять до коштовних переробок на пізніх етапах проєкту [15].

Управління складними залежностями між різними елементами проєкту становить особливий виклик у сфері інтерактивної візуалізації. Робота над одним ассетом може впливати на безліч інших елементів проєкту через складну мережу технічних та естетичних взаємозв'язків. Наприклад, зміна системи освітлення може вимагати коригування матеріалів усіх об'єктів у сцені; оптимізація геометрії для підвищення продуктивності може впливати на візуальну якість та потребувати компенсації через більш детальні текстури; зміна архітектурного плану може вплинути на навігацію користувача та вимагати переробки інтерфейсу [16]. Менеджер проєкту повинен мати глибоке розуміння цих взаємозалежностей та здатність передбачати каскадні ефекти змін.

Управління очікуваннями стейкхолдерів у проєктах інтерактивної візуалізації вимагає особливої уваги та комунікаційних навичок. Нетехнічні замовники часто мають нереалістичні очікування щодо того, що можливо реалізувати в рамках заданих обмежень бюджету, часу та продуктивності. Вони

можуть бачити вражаючі демонстрації передових технологій або кінематографічні рендери і очікувати того самого рівня якості в інтерактивному середовищі реального часу, не розуміючи технічних компромісів, які необхідні для досягнення плавної роботи [17]. Менеджер проєкту повинен ефективно комунікувати ці обмеження, використовуючи візуальні демонстрації та порівняння, щоб допомогти замовнику зрозуміти баланс між візуальною якістю та продуктивністю.

Специфіка командної динаміки в проєктах інтерактивної візуалізації також відрізняється від традиційних технічних проєктів. Команда зазвичай включає фахівців з дуже різними профілями: 3D-художників, концепт-артистів, технічних художників, програмістів, дизайнерів рівнів та аніматорів. Кожна з цих ролей має свої специфічні робочі процеси, інструменти та професійну мову, що може створювати бар'єри в комунікації. Художники можуть думати в термінах композиції, освітлення та емоційного впливу, тоді як програмісти оперують поняттями алгоритмів, продуктивності та оптимізації. Менеджер проєкту повинен виступати мостом між цими різними світами, забезпечуючи ефективну комунікацію та взаєморозуміння [18].

Оцінка трудомісткості та планування ресурсів у проєктах інтерактивної візуалізації є особливо складним завданням через високу варіативність часу, необхідного для створення різних типів контенту. Час створення простого пропа може варіюватися від кількох годин до кількох днів залежно від складності форми та вимог до деталізації. Складний архітектурний об'єкт може вимагати тижнів роботи. Більше того, оцінки сильно залежать від досвіду конкретного виконавця, доступності референсів та технічних обмежень проєкту. Традиційні методи оцінки, такі як аналогія з попередніми проєктами або експертна оцінка, повинні використовуватися з обережністю та постійно коригуватися на основі фактичної швидкості роботи команди.

Управління технічним боргом є критично важливим аспектом довгострокового успіху проєктів інтерактивної візуалізації. Під тиском дедлайнів команди часто приймають рішення про швидкі, але технічно

недосконалі рішення: неоптимізовані моделі, захардкожені значення, дублювання коду або асетів, недосконалі процедури імпорту. Ці технічні компроміси дозволяють швидше досягти короткострокових цілей, але накопичуються у формі технічного боргу, який згодом сповільнює розробку, ускладнює внесення змін та знижує стабільність проєкту. Досвідчений менеджер проєкту повинен свідомо управляти цим балансом, виділяючи час на рефакторинг та оптимізацію, щоб запобігти критичному накопиченню технічного боргу [19].

Контроль якості в проєктах інтерактивної візуалізації включає як об'єктивні технічні метрики, так і суб'єктивні естетичні оцінки. З технічної точки зору необхідно моніторити продуктивність рендерингу, кількість полігонів, розмір текстур, споживання пам'яті та інші метрики, які впливають на користувацький досвід. З естетичної точки зору потрібна регулярна оцінка візуальної якості, узгодженості стилю, ефективності освітлення та загального емоційного впливу середовища. Ці два аспекти якості часто знаходяться у конфлікті: підвищення візуальної якості зазвичай вимагає більше технічних ресурсів, що може негативно впливати на продуктивність. Знаходження оптимального балансу між цими факторами є постійним викликом протягом усього життєвого циклу проєкту.

Специфіка управління проєктами інтерактивної візуалізації вимагає від менеджера проєкту унікальної комбінації навичок, знань та досвіду. Успішний менеджер повинен володіти не лише традиційними компетенціями проєктного менеджменту, такими як планування, контроль бюджету та управління ризиками, але й мати глибоке розуміння технологій 3D-графіки, принципів візуального дизайну, особливостей креативного процесу та специфіки робочих процесів у різних програмних пакетах [40]. Лише така комплексна експертиза дозволяє ефективно керувати складними проєктами на перетині мистецтва та технології, забезпечуючи створення високоякісних інтерактивних візуальних середовищ у встановлені терміни та в рамках бюджету.

### 1.3. Сучасні тенденції та інструменти архітектурної візуалізації

Архітектурна візуалізація як галузь зазнала радикальної трансформації протягом останнього десятиліття, перейшовши від створення статичних фотореалістичних зображень до розробки повноцінних інтерактивних цифрових середовищ. Цей перехід був спричинений конвергенцією кількох технологічних трендів: стрімким зростанням обчислювальної потужності графічних процесорів, розвитком ігрових рушіїв як платформ для архітектурної візуалізації, появою технологій віртуальної та доповненої реальності, а також зміною очікувань клієнтів, які прагнуть не просто побачити майбутній об'єкт, але й відчутти його простір, масштаб та атмосферу [20]. Сучасна архітектурна візуалізація стала інструментом не лише презентації, але й проектування, дозволяючи архітекторам та дизайнерам експериментувати з різними варіантами рішень у віртуальному середовищі до початку будівництва.

Найбільш революційною тенденцією останніх років став перехід від традиційного попередньо розрахованого рендерингу до рендерингу в реальному часі. Традиційні методи архітектурної візуалізації, що базуються на рендерерах таких як V-Ray, Corona Renderer або Arnold, створюють статичні зображення надзвичайно високої якості, проте вимагають значних часових витрат на обчислення кожного кадру. Складна сцена з детальним освітленням, відбиттями та глобальним освітленням може рендеритися годинами або навіть днями на потужних робочих станціях [21]. Цей підхід добре працює для створення маркетингових матеріалів або презентаційних зображень, але він принципово обмежує можливості для експериментування та взаємодії з проектом.

Рендеринг у реальному часі, реалізований через ігрові рушії на кшталт Unreal Engine та Unity, пропонує принципово інший підхід. Замість багатогодинного очікування результату, зображення генерується миттєво з частотою тридцять, шістдесят або навіть більше кадрів на секунду, що дозволяє користувачеві вільно переміщатися віртуальним простором, змінювати кут огляду, експериментувати з освітленням та матеріалами в режимі реального часу

[39]. Ця інтерактивність трансформує процес прийняття проектних рішень, дозволяючи архітекторам та клієнтам миттєво бачити результати змін та робити більш інформовані вибори. Технологічний прорив, який зробив можливим досягнення фотореалістичної якості в реальному часі, став можливим завдяки розвитку спеціалізованих апаратних можливостей графічних процесорів та революційних програмних технологій.

Unreal Engine 5 представив дві ключові технології, які кардинально змінили ландшафт архітектурної візуалізації: Nanite та Lumen. Nanite є системою віртуалізованої геометрії, яка дозволяє працювати з моделями надзвичайно високої деталізації, що містять сотні мільйонів або навіть мільярди полігонів, без традиційної необхідності створення множинних рівнів деталізації [22]. Система працює шляхом потокової передачі та рендерингу лише тих деталей геометрії, які фактично видимі з поточної точки огляду, автоматично регулюючи рівень деталізації залежно від відстані та розміру об'єкта на екрані. Це означає, що архітектори можуть імпортувати високодеталізовані моделі безпосередньо зі сканувань реальних об'єктів або детальні CAD-моделі без трудомісткого процесу оптимізації та створення низькополігональних версій [38].

Технологія Lumen представляє собою повністю динамічну систему глобального освітлення, яка працює в реальному часі без необхідності попереднього розрахунку та "запікання" світлових карт. Глобальне освітлення є критично важливим для створення реалістичних архітектурних візуалізацій, оскільки воно симулює складну взаємодію світла з поверхнями: непряме освітлення, відбиття кольору від одних поверхонь на інші, м'які тіні та природну атмосферу простору [23]. Традиційно досягнення такого рівня реалізму вимагало багатогодинного передрозрахунку, що робило неможливим експериментування з різними сценаріями освітлення. Lumen вирішує цю проблему, обчислюючи глобальне освітлення в реальному часі, дозволяючи миттєво бачити результати зміни положення сонця, інтенсивності джерел світла або кольору матеріалів.

Імерсивні технології віртуальної та доповненої реальності відкрили нові можливості для сприйняття та взаємодії з архітектурними проєктами. Віртуальна

реальність забезпечує повне занурення користувача у цифрову модель будівлі або простору, дозволяючи відчувати реальний масштаб приміщень, оцінити пропорції та просторові взаємозв'язки способом, який неможливий при перегляді на плоскому екрані [24]. Досвід перебування всередині віртуального простору активує ті самі когнітивні механізми просторової орієнтації, що і реальне перебування в фізичному просторі, що дозволяє виявити проблеми планування, які можуть бути непомітні на кресленнях або традиційних візуалізаціях. Клієнти можуть буквально "пройтися" своїм майбутнім будинком, відчувати розміри кімнат, оцінити види з вікон та прийняти більш впевнені рішення щодо планування.

Доповнена реальність пропонує комплементарний підхід, накладаючи цифрову модель на фізичний світ через смартфони, планшети або спеціалізовані AR-пристрої на кшталт Microsoft HoloLens. Ця технологія особливо цінна на етапі будівництва та на місці майбутнього об'єкта, оскільки вона дозволяє побачити, як проєктована будівля буде виглядати в реальному контексті оточення, оцінити її масштаб відносно існуючих об'єктів та перевірити, як вона інтегрується в ландшафт. Будівельники можуть використовувати AR для перевірки правильності розміщення конструктивних елементів, порівнюючи фактичне виконання з цифровою моделлю безпосередньо на будівельному майданчику [37]. Поєднання VR та AR створює повний спектр інструментів для роботи з архітектурними проєктами на всіх етапах їх життєвого циклу [25].

Процедурне моделювання стало потужним інструментом для підвищення ефективності створення складних архітектурних середовищ. На відміну від традиційного моделювання, де художник вручну створює кожен елемент геометрії, процедурне моделювання базується на визначенні правил та алгоритмів, які автоматично генерують контент на основі заданих параметрів. Цей підхід особливо ефективний для створення повторюваних елементів, складних структур та великомасштабних середовищ. Наприклад, фасад багатоповерхової будівлі з повторюваним візерунком вікон може бути

згенерований процедурно шляхом визначення базового модуля та правил його повторення, замість створення кожного вікна окремо.

Інструменти процедурного моделювання, такі як Houdini, стали стандартом у архітектурній візуалізації для створення складних систем. Houdini використовує node-based підхід, де художник створює мережу вузлів, кожен з яких виконує певну операцію над геометрією або даними, дозволяючи створювати надзвичайно гнучкі та потужні процедурні системи. Ці системи можуть генерувати все від простих архітектурних елементів до цілих міських кварталів, автоматично враховуючи обмеження та правила, визначені художником. Параметричний характер процедурного моделювання означає, що зміна вхідних параметрів автоматично оновлює весь згенерований контент, що надзвичайно цінно для ітеративного процесу проектування, де вимоги часто змінюються [26].

Автоматизація розміщення ландшафтних елементів через процедурні системи дозволяє створювати реалістичні та різноманітні зовнішні середовища з мінімальними ручними зусиллями. Системи розкидування (scatter systems) можуть автоматично розміщувати тисячі об'єктів рослинності, каміння, трави та інших природних елементів на основі правил, що враховують характеристики рельєфу, тип ґрунту, схили та інші параметри [36]. Ці системи можуть імітувати природні паттерни розподілу рослинності, уникаючи штучного, рівномірного розміщення, та забезпечувати, що об'єкти правильно взаємодіють з рельєфом, не проникаючи один в одного або не зависаючи в повітрі. Результатом є переконливі ландшафти, створення яких вручну вимагало б тижнів монотонної роботи [35].

Blender як інструмент для архітектурної візуалізації зайняв унікальну позицію завдяки своїй відкритій природі, потужному функціоналу та відсутності вартості ліцензування. Цей програмний пакет пропонує повний набір інструментів для моделювання, скульптингу, текстурювання, анімації та рендерингу, що робить його самодостатнім рішенням для створення архітектурних візуалізацій. Система модифікаторів Blender дозволяє неруйнівно

змінювати геометрію, що особливо цінно для архітектурного моделювання, де часто потрібно експериментувати з різними варіантами форм. Вбудовані рендерери Cycles та Eevee пропонують відповідно фотореалістичний path-traced рендеринг та швидкий рендеринг у реальному часі, забезпечуючи гнучкість у виборі між якістю та швидкістю [27].

Ecosystem розширень та доповнень Blender для архітектурної візуалізації постійно розвивається, пропонуючи спеціалізовані інструменти для різних аспектів роботи. Доповнення на кшталт Archimesh автоматизують створення стандартних архітектурних елементів, таких як двері, вікна, драбини та перила. BlenderBIM інтегрує можливості роботи з форматом Industry Foundation Classes, дозволяючи імпортувати та експортувати BIM-моделі та зберігати метадані об'єктів. Параметричні генератори створюють процедурні будівлі, міста та ландшафти безпосередньо в Blender. Ця екосистема інструментів, створена спільнотою, значно розширює можливості Blender для архітектурних застосувань без додаткових фінансових інвестицій.

Інтеграція між програмними пакетами стала критично важливою для ефективного робочого процесу в архітектурній візуалізації[34]. Сучасні проекти зазвичай використовують множину інструментів, кожен з яких оптимізований для певних завдань: CAD-програми для точного технічного креслення, 3D-пакети для моделювання та текстурювання, ігрові рушії для інтерактивної візуалізації, спеціалізовані інструменти для ландшафтного дизайну. Забезпечення плавного обміну даними між цими інструментами через стандартизовані формати, такі як FBX, USD або glTF, є критично важливим для продуктивності. Blender підтримує широкий спектр форматів імпорту та експорту, дозволяючи ефективно інтегруватися в різноманітні робочі процеси та легко передавати моделі в Unreal Engine для фінальної візуалізації та інтерактивності [28].

Штучний інтелект починає відігравати все більш важливу роль у архітектурній візуалізації, автоматизуючи рутинні завдання та відкриваючи нові креативні можливості. AI-powered інструменти, такі як Meshy AI, можуть

генерувати 3D-моделі на основі текстових описів або 2D-зображень, значно прискорюючи процес створення базових асетів [33]. Технології машинного навчання використовуються для автоматичного покращення якості текстур, апскейлінгу зображень низької роздільної здатності, генерації карт нормалей з альbedo-текстур та навіть для автоматичної UV-розгортки складної геометрії. Інструменти на кшталт NormalMap-Online демонструють, як AI може спростити технічні процеси, дозволяючи художникам зосередитися на креативних аспектах роботи замість рутинних технічних завдань.

Хмарні технології та колаборативні платформи трансформують спосіб роботи команд над архітектурними візуалізаціями. Хмарний рендеринг дозволяє розподіляти обчислювально інтенсивні завдання на потужні серверні ферми, прискорюючи створення високоякісних рендерів у десятки разів порівняно з локальними робочими станціями [32]. Платформи для спільної роботи дозволяють множині художників одночасно працювати над різними аспектами проєкту, синхронізуючи зміни в реальному часі та забезпечуючи узгодженість результатів [30]. Веб-базовані переглядачі 3D-моделей дозволяють клієнтам та стейкхолдерам переглядати та коментувати проєкти безпосередньо в браузері без необхідності встановлення спеціалізованого програмного забезпечення, значно спрощуючи процес узгодження та отримання зворотного зв'язку.

Photogrammetry та 3D-сканування стали важливими джерелами високоякісного контенту для архітектурних візуалізацій. Фотограмметрія дозволяє створювати точні 3D-моделі реальних об'єктів, матеріалів та текстур шляхом обробки серії фотографій, зроблених з різних кутів. Ця технологія особливо цінна для створення реалістичних моделей існуючих будівель, природних елементів ландшафту та автентичних матеріалів. Лазерне сканування забезпечує ще вищу точність, створюючи детальні хмари точок існуючих просторів, які можуть бути використані як основа для реконструкції або як контекст для нових проєктів. Інтеграція відсканованих даних з процедурно згенерованим та моделюваним вручну контентом створює гібридні середовища надзвичайної деталізації та реалізму [29].

Сучасні тенденції в архітектурній візуалізації відображають загальний рух індустрії від статичної презентації до інтерактивного досвіду, від ізольованих інструментів до інтегрованих екосистем, від повністю ручного створення контенту до інтелектуальної автоматизації з використанням процедурних методів та штучного інтелекту. Ці зміни не просто підвищують ефективність роботи, але фундаментально змінюють природу архітектурної візуалізації, перетворюючи її з інструменту презентації на інтегральну частину процесу проектування. Майбутнє галузі лежить у напрямку все більшої інтерактивності, персоналізації досвіду та використання технологій розширеної реальності для створення глибокого емоційного зв'язку між людьми та просторами, які ще не існують фізично, але вже можуть бути пережиті та відчуті у віртуальному вимірі [31].

## **Висновки до розділу 1**

Проведене дослідження теоретико-методологічних засад дозволило встановити, що управління проектами створення інтерактивних віртуальних просторів є складною галуззю, яка вимагає адаптації класичних управлінських підходів до специфіки креативно-технічної розробки. Аналіз еволюції методологій показав, що хоча каскадна модель забезпечує чіткість планування та передбачуваність бюджету, вона є недостатньо ефективною в умовах високої невизначеності та мінливих вимог, характерних для візуалізації. Натомість гнучкі методології Agile, зокрема фреймворк Scrum та система Kanban, пропонують необхідну адаптивність та ітеративність, проте можуть створювати труднощі у довгостроковому плануванні ресурсів.

Визначено, що специфіка проектів інтерактивної візуалізації полягає у їхній подвійній природі, де постійно відбувається пошук балансу між художньою виразністю та жорсткими технічними обмеженнями апаратного забезпечення. Це формує унікальні виклики для менеджера проекту, зокрема складність оцінки трудомісткості креативних завдань, необхідність управління

суб'єктивними критеріями якості та високі ризики, пов'язані з використанням новітніх технологій. Встановлено, що сильним факторами успіху є обов'язкове виділення часу на етап досліджень та розробки (R&D), використання прототипування (greyboxing) для ранньої валідації рішень та ефективне управління технічним боргом.

Аналіз сучасних тенденцій у сфері архітектурної візуалізації засвідчив фундаментальний перехід індустрії від статичного попередньо розрахованого рендерингу до інтерактивних середовищ реального часу. Ключову роль у цій трансформації відіграють ігрові рушії, зокрема Unreal Engine 5, який завдяки технологіям віртуалізованої геометрії Nanite та динамічного глобального освітлення Lumen дозволяє досягати кінематографічної якості зображення без традиційних витрат часу на оптимізацію та запікання світла.

Також виявлено важливість інтеграції інструментарію, де програмне забезпечення Blender виступає ефективним та економічно доцільним засобом для створення 3D-контенту, забезпечуючи повний цикл моделювання та текстурування.

## **РОЗДІЛ 2. УПРАВЛІННЯ ПРОЄКТОМ СТВОРЕННЯ ВІЗУАЛЬНОЇ ЧАСТИНИ ІНТЕРАКТИВНОГО ПРОСТОРУ НаУОА**

### **2.1. Аналіз та планування проєкту**

Процес управління проєктом створення візуальної частини інтерактивного простору Національного університету «Острозька академія» розпочався з детального аналізу вимог та планування всіх аспектів майбутньої роботи. Інформаційно-технічний центр НаУОА, як база впровадження проєкту, надав унікальну можливість для реалізації інноваційного рішення у освітньому середовищі, що поєднує технічну підтримку інфраструктури університету з розробкою та впровадженням новітніх технологій у навчальний процес.

Характеристика організаційного середовища проєкту показала, що Інформаційно-технічний центр спеціалізується на підтримці серверів, комп'ютерних класів та мережевого обладнання, що забезпечує необхідну технічну базу для розробки та тестування віртуального простору. Основним завданням стало створення презентабельного інтерактивного простору університету, який би відтворював автентичність історичних будівель академії та забезпечував високий рівень візуальної якості при збереженні оптимальної продуктивності системи. Проєкт передбачав виконання різноманітної візуальної частини середовища академії з використанням сучасних технологій тривимірного моделювання та рендерингу в реальному часі.

Аналіз вимог до проєкту виявив необхідність збалансованого підходу між художньою складовою та технічними обмеженнями. Візуальна частина повинна була забезпечувати високий рівень деталізації архітектурних елементів староакадемічного корпусу, коридорів та аудиторій першого і другого поверхів, а також паркового простору біля бібліотеки. Технічні вимоги включали оптимізацію геометрії для забезпечення плавного рендерингу в реальному часі, правильну UV-розгортку для якісного текстурування та сумісність між різними програмними середовищами.

## Структурна декомпозиція робіт проекту

Рівень 1	Рівень 2	Рівень 3	Опис завдання	Відповідальний
1. Ініціація проекту	1.1. Визначення вимог	1.1.1. Аналіз потреб замовника	Збір інформації про очікування ІТ-центру НаУОА	Менеджер проекту
		1.1.2. Технічне завдання	Формулювання технічних та функціональних вимог	Менеджер проекту
	1.2. Формування команди	1.2.1. Розподіл ролей	Визначення 3D-моделерів, технічних художників	Менеджер проекту
2. Планування	2.1. Планування змісту	2.1.1. Список об'єктів для моделювання	Визначення всіх 3D-асетів проекту	3D-моделери
		2.1.2. Технічні специфікації	Встановлення стандартів полігонажу, текстур	Технічний художник
	2.2. Планування часу	2.2.1. Календарний план	Розподіл завдань по спринтах	Менеджер проекту
		2.2.2. Критичний шлях	Ідентифікація залежностей завдань	Менеджер проекту
3. Виконання	3.1. Створення 3D-контенту	3.1.1. Моделювання в Blender	Створення архітектурних елементів, меблів	3D-моделери
		3.1.2. UV-розгортка	Підготовка моделей до текстурування	3D-моделери
		3.1.3. Текстурування	Створення матеріалів та текстур	Художник текстур

Продовження таблиці 2.1

	3.2. Інтеграція в UE5	3.2.1. Імпорт ассетів	Перенесення моделей з Blender	Технічний художник
		3.2.2. Налаштування освітлення	Конфігурація Lumen	Технічний художник
		3.2.3. Створення Blueprints	Програмування інтерактивності	Blueprint-програміст
4. Контроль та оптимізація	4.1. Тестування	4.1.1. Тести продуктивності	Вимірювання FPS, споживання пам'яті	Технічний художник
		4.1.2. Візуальне QA	Перевірка якості візуалізації	Художній керівник
	4.2. Оптимізація	4.2.1. Оптимізація геометрії	Застосування Decimate, ручна корекція	3D-моделери
		4.2.2. Оптимізація текстур	Атласування, стиснення	Художник текстур

Планування ресурсів проєкту виявило необхідність використання потужного програмного забезпечення та апаратних засобів. Основним інструментом для тривимірного моделювання було обрано Blender через його повний функціонал, відкритий код та відсутність ліцензійних витрат, що було критично важливим для освітнього закладу. Програмний пакет включає засоби моделювання, анімації, рендерингу та пост-обробки, що робить його самодостатнім рішенням для створення 3D-контенту. Для фінальної інтеграції та візуалізації було обрано Unreal Engine 5, який дозволяє створювати інтерактивні середовища з використанням активів високої деталізації завдяки технології Nanite та реалістичним освітленням через систему Lumen.

Додаткові інструменти включали Adobe Illustrator для створення векторної графіки, логотипів та графічних елементів інтерфейсу, Meshy AI як генератор

тривимірних моделей на основі штучного інтелекту для прискорення створення базових асетів, та NormalMap-Online для безкоштовного створення карт нормалей з карт висот, що дозволяє імітувати детальний рельєф поверхні без збільшення кількості полігонів. Така комбінація інструментів забезпечила повний технологічний стек для реалізації всіх аспектів проекту від початкового моделювання до фінальної інтерактивної візуалізації.

Таблиця 2.2

## Календарний план виконання проекту

Етап	Завдання	Тривалість (тижні)	Початок	Завершенн я	Попередні завдання
Ініціація	Аналіз вимог та створення ТЗ	1	19.06.2025	26.06.2025	-
Планування	Розробка детального плану	1	26.06.2025	03.07.2025	Аналіз вимог
	Дослідження технологій (R&D)	2	03.07.2025	17.07.2025	Розробка плану
Виконання : Спринт 1	Створення базових архітектурних елементів	2	17.07.2025	31.07.2025	R&D
	Моделювання екстер'єру корпусу	2	17.07.2025	31.07.2025	R&D
Виконання : Спринт 2	Моделювання меблів та декору	2	31.07.2025	14.08.2025	Спринт 1
	UV-розгортка та текстурування	2	31.07.2025	14.08.2025	Спринт 1
Виконання : Спринт 3	Створення інтер'єрів коридорів	2	14.08.2025	28.08.2025	Спринт 2

## Продовження таблиці 2.2

	Розробка аудиторій 1-2 поверхів	2	14.08.2025	28.08.2025	Спринт 2
Інтеграція	Імпорт моделей в Unreal Engine 5	1	28.08.2025	04.09.2025	Спринт 3
	Налаштування освітлення Lumen	1	04.09.2025	11.09.2025	Імпорт моделей
	Створення Blueprints для інтерактивності	1	11.09.2025	18.09.2025	Налаштування освітлення
Оптимізація	Тестування продуктивності	1	18.09.2025	25.09.2025	Створення Blueprints
	Оптимізація геометрії та текстур	2	25.09.2025	09.10.2025	Тестування
Завершення	Фінальне тестування та виправлення	1	09.10.2025	16.10.2025	Оптимізація
	Підготовка документації	1	16.10.2025	23.10.2025	Фінальне тестування

Управління часом проєкту базувалося на гібридному підході, який поєднував елементи каскадної моделі для загального планування віх з гнучкою методологією Scrum для безпосереднього виконання робіт. Проєкт було розділено на три основні фази виконання, кожна з яких представляла собою двотижневий спринт з чітко визначеними цілями та очікуваними результатами.

Перший спринт зосереджувався на створенні базових архітектурних елементів та моделюванні екстер'єру староакадемічного корпусу, що дозволило встановити загальний масштаб та стиль проєкту на ранній стадії.

Другий спринт був присвячений деталізації середовища через моделювання меблів, декоративних елементів та проведення UV-розгортки всіх

створених моделей для подальшого текстурування. Цей етап виявився критичним для встановлення стандартів якості та технічних специфікацій, які застосовувалися до всіх наступних ассетів. Третій спринт охоплював створення внутрішніх просторів, включаючи коридори та аудиторії першого і другого поверхів, що вимагало особливої уваги до деталей інтер'єру та функціональних елементів навігації користувача.

Таблиця 2.3

## Матриця ризиків проєкту

ID	Ризик	Ймовірність	Вплив	Рівень ризику	Стратегія реагування	Превентивні заходи
R1	Технічні проблеми інтеграції Blender-UE5	Середня	Високий	Високий	Пом'якшення	Виділення 2 тижнів на R&D, тестування pipeline на простих моделях
R2	Низька продуктивність через надмірну деталізацію	Висока	Високий	Критичний	Пом'якшення	Встановлення технічних стандартів полігонажу, регулярні тести FPS
R3	Затримки через недооцінку складності моделювання	Середня	Середній	Середній	Прийняття	Додавання 20% buffer часу, можливість зменшення score
R4	Несумісність версій програмного забезпечення	Низька	Високий	Середній	Уникнення	Фіксація версій ПЗ на початку проєкту, документування

## Продовження таблиці 2.3

R5	Втрата даних через відсутність backup	Низька	Критичний	Високий	Передача	Автоматичне резервне копіювання на хмарне сховище щоденно
R6	Невідповідність візуального стилю очікуванням	Середня	Середній	Середній	Пом'якшення	Створення mood board на початку, регулярні демо замовнику
R7	Технічні обмеження апаратного забезпечення	Низька	Середній	Низький	Прийняття	Тестування на різних конфігураціях, оптимізація під мінімальні вимоги
R8	Зміна вимог в процесі розробки	Висока	Середній	Високий	Пом'якшення	Agile-підхід зі спринтами, гнучке управління backlog
R9	Недостатня експертиза в Unreal Engine 5	Середня	Високий	Високий	Пом'якшення	Проходження online курсів, виділення часу на навчання
R10	Проблеми з UV-розгорткою складних моделей	Висока	Низький	Середній	Прийняття	Використання автоматичних методів Smart UV, допустимість невеликих артефактів

Ідентифікація та аналіз ризиків проєкту показали, що найбільшу загрозу представляють технічні аспекти інтеграції між різними програмними середовищами та забезпечення необхідної продуктивності при високій візуальній якості. Ризик низької продуктивності через надмірну деталізацію моделей було класифіковано як критичний, оскільки він міг зробити неможливим інтерактивну взаємодію користувача з середовищем в реальному часі. Для мітігації цього ризику було запроваджено жорсткі технічні стандарти максимальної кількості полігонів для різних типів об'єктів та регулярне тестування продуктивності після кожного спринту.

Стратегія управління ризиками включала виділення значного часу на дослідження та розробку на початковій стадії проєкту, що дозволило виявити потенційні технічні проблеми до початку масового створення контенту. Зокрема, двотижнева фаза R&D була присвячена тестуванню pipeline роботи між Blender та Unreal Engine 5, експериментам з різними методами експорту моделей, налаштуванню систем Nanite та Lumen, та створенню базових шаблонів для Blueprints. Це дозволило команді набути необхідної експертизи та розробити набір кращих практик для подальшої роботи.

Таблиця 2.4

## Технічні специфікації 3D-асетів

Категорія об'єктів	Максимальна кількість полігонів	Розмір текстур	Формат експорту	Використання Nanite	Примітки
Основна архітектура (будівлі)	500,000 - 2,000,000	4K (4096×4096)	FBX	Так	Високодеталізовані фасади, складний декор
Великі меблі (столи, шафи)	10,000 - 30,000	2K (2048×2048)	FBX	Ні	Оптимізована топологія, quads

## Продовження таблиці 2.4

Дрібні меблі (стілці, лампи)	3,000 - 8,000	1К (1024×1024)	FBX	Ні	Можливість атласування текстур
Декоративні елементи	1,000 - 5,000	512×512 або атлас 2К	FBX	Ні	Групування в атласи по 10-15 об'єктів
Двері та вікна	8,000 - 15,000	2К (2048×2048)	FBX	Ні	Модульна конструкція для повторного використання
Елементи інтер'єру (плінтуси, карнизи)	2,000 - 5,000	1К (1024×1024) або атлас	FBX	Ні	Процедурне розміщення вздовж стін
Ландшафт та рослинність	5,000 - 50,000	2К (2048×2048)	FBX	Частково	Варіативність для природного вигляду
Пропси (книги, документи)	500 - 2,000	512×512 або атлас	FBX	Ні	Максимальне атласування

Встановлення технічних стандартів для різних категорій тривимірних ассетів стало фундаментальним кроком у забезпеченні узгодженості роботи команди та контролю якості результатів. Специфікації визначали не лише максимально допустиму кількість полігонів та розмір текстур, але й рекомендації щодо застосування технології Nanite, яка дозволяє працювати з надвисокодеталізованими моделями для основних архітектурних об'єктів. Для менших об'єктів, таких як меблі та декоративні елементи, було вирішено використовувати традиційний підхід з ручною оптимізацією топології, оскільки це забезпечує кращий контроль над продуктивністю та дозволяє ефективно застосовувати атласування текстур.

Стратегія атласування текстур передбачала об'єднання множини дрібних об'єктів подібної тематики в єдину текстурну карту, що значно зменшує кількість

draw calls та покращує продуктивність рендерингу. Наприклад, всі канцелярські пропси однієї аудиторії, включаючи ручки, олівці, папери та книги, могли використовувати одну спільну текстуру розміром два на два кілопікселі замість десятків окремих маленьких текстур.

Таблиця 2.5

## Розподіл ролей та відповідальності команди

Роль	Відповідальність	Ключові завдання	Необхідні компетенції	Час залучення
Менеджер проєкту	Загальне управління проєктом, координація команди	Планування, контроль термінів, управління ризиками, комунікація зі стейкхолдерами	Знання методологій PM, розуміння 3D-pipeline	Повний час
3D-моделер (архітектура)	Створення архітектурних елементів	Моделювання будівель, фасадів, складних конструктивних елементів у Blender	Досвід архітектурного моделювання, знання принципів оптимізації	70% часу проєкту
3D-моделер (assets)	Створення меблів та пропсів	Моделювання внутрішніх об'єктів, меблів, декору	Навички hard-surface моделювання, UV-unwrapping	60% часу проєкту
Художник текстур	Створення матеріалів та текстур	Текстурування моделей, створення PBR-матеріалів, налаштування в UE5	Розуміння PBR-workflow, робота з Substance Painter/Blender	50% часу проєкту

Продовження таблиці 2.5

Технічний художник	Інтеграція та оптимізація	Імпорт асетів в UE5, налаштування Nanite/Lumen, оптимізація продуктивності	Експертиза в UE5, розуміння технічних обмежень real-time рендерингу	80% часу проєкту
Blueprint-програміст	Створення інтерактивності	Розробка систем навігації, інтерактивних об'єктів, UI у Blueprints	Досвід роботи з Blueprints UE5, базове розуміння ігрового дизайну	40% часу проєкту
QA-тестер	Контроль якості	Тестування продуктивності, виявлення візуальних артефактів, перевірка інтерактивності	Увага до деталей, розуміння стандартів якості 3D-графіки	30% часу проєкту
Художній керівник	Контроль візуального стилю	Затвердження концептів, перевірка відповідності стилю, фінальна QA візуалу	Художній смак, розуміння архітектурних стилів, досвід у візуалізації	20% часу проєкту

Формування структури команди та чіткий розподіл відповідальності між членами стали критично важливими для ефективної координації робіт у проєкті, що поєднує технічні та креативні аспекти. Роль менеджера проєкту передбачала не лише традиційні функції планування та контролю, але й глибоке розуміння специфіки тривимірної графіки та технічних обмежень використовуваних інструментів, що дозволило ефективно балансувати між художніми амбіціями та технічними можливостями.

Особлива увага приділялася комунікації між 3D-моделерами, які створювали контент у Blender, та технічним художником, відповідальним за

інтеграцію цього контенту в Unreal Engine 5, оскільки саме на стику цих процесів виникала більшість технічних проблем.

Розподіл часу залучення різних спеціалістів відображав специфіку життєвого циклу проєкту, де інтенсивна робота над моделюванням на початкових етапах поступово змінювалася фокусом на інтеграцію, оптимізацію та створення інтерактивності на фінальних стадіях.

Технічний художник був задіяний протягом більшості часу проєкту, оскільки його експертиза була необхідна як на етапі R&D для встановлення технічних стандартів, так і на всіх наступних етапах для забезпечення коректної інтеграції ассетів та оптимізації продуктивності. Blueprint-програміст залучався переважно на пізніших етапах, коли основний візуальний контент вже був створений і готовий до додавання інтерактивних елементів.

Планування комунікацій у проєкті базувалося на принципах Agile-методології з регулярними щоденними stand-up зустрічами тривалістю п'ятнадцять хвилин для синхронізації поточної роботи команди. Кожен спринт розпочинався зустріччю Sprint Planning, де детально обговорювалися цілі наступних двох тижнів та відбувався відбір завдань з product backlog.

Наприкінці спринту проводилися дві критично важливі зустрічі: Sprint Review для демонстрації результатів стейкхолдерам з Інформаційно-технічного центру та отримання зворотного зв'язку, та Sprint Retrospective для аналізу процесу роботи команди та ідентифікації можливостей для покращення.

Для візуального планування та відстеження прогресу використовувалася Kanban-дошка у Trello, яка забезпечувала прозорість стану кожного завдання для всіх членів команди. Дошка була організована у п'ять колонок, що відображали етапи роботи над ассетом:

Backlog для нових завдань,

In Progress для активної роботи,

Review для завдань на перевірці,

Testing для етапу тестування в ігровому рушії,

Done для завершених робіт.

Обмеження Work In Progress встановлювало максимум три одночасних завдання на одного виконавця, що запобігало перевантаженню та забезпечувало фокус на завершенні розпочатої роботи перед стартом нових задач.

Критичний шлях проєкту проходив через послідовність завдань від створення базової архітектури будівель через їх детальне текстурування до інтеграції в Unreal Engine 5 та налаштування освітлення, оскільки кожен наступний етап був неможливий без завершення попереднього. Затримки на будь-якому з цих етапів безпосередньо впливали на загальні терміни завершення проєкту, тому саме ці завдання отримували найвищий пріоритет у плануванні та найпильнішу увагу при моніторингу прогресу.

Паралельно критичному шляху велася робота над некритичними елементами, такими як дрібні декоративні об'єкти та пропси, які могли бути додані на фінальних етапах без впливу на загальний графік.

## **2.2. Реалізація проєкту: створення 3D-контенту**

Етап реалізації проєкту розпочався з активної фази створення тривимірного контенту, яка стала найбільш трудомісткою та творчо насиченою частиною всього проєкту. Blender було обрано як основний інструмент для моделювання через його потужний функціонал, відкриту природу програмного коду та повну безоплатність, що робило його ідеальним вибором для освітнього закладу з обмеженим бюджетом. Програмний пакет використовувався для повного циклу створення моделей від базового моделювання примітивних форм до фінальної підготовки ассетів для експорту в ігровий рушій, включаючи всі проміжні етапи UV-розгортки, текстурування та оптимізації геометрії.

Процес моделювання розпочинався з створення базових архітектурних елементів староакадемічного корпусу, що вимагало особливої уваги до історичної автентичності та архітектурних деталей. Робота над фасадом будівлі включала відтворення характерних елементів класичної архітектури: колон, карнизів, віконних обрамлень, ліпного декору та барельєфів, кожен з яких

потребував окремої уваги до пропорцій та деталізації. Моделювання велося з використанням референсних фотографій реальної будівлі університету, що дозволило досягти високого рівня відповідності оригіналу та зберегти унікальний характер історичної споруди у віртуальному просторі.

Однією з ключових задач на етапі моделювання стало створення деталізованих, але технічно оптимізованих моделей, що потребувало постійного балансування між візуальною якістю та продуктивністю системи. Навіть з урахуванням потужних технологій Unreal Engine 5, таких як Nanite, принцип чистої топології залишався критично важливим для продуктивності, особливо для динамічних об'єктів та складних сцен з множиною інтерактивних елементів. Для складної органічної або архітектурної геометрії, зокрема скульптурних елементів фасадів та складних меблів, активно застосовувався модифікатор Decimate, який дозволяв алгоритмічно зменшити кількість полігонів, зберігаючи при цьому основні візуальні ознаки об'єкта та досягаючи балансу між візуальною якістю та низькою полігональною вагою.

Для більшості архітектурних елементів, таких як стіни, двері та вікна, використовувався ручний підхід до моделювання з акцентом на створення чіткої, акуратної топології з переважанням чотирикутних полігонів. Такий підхід мінімізував артефакти при застосуванні текстур та субдвіжн-поверхонь, забезпечуючи передбачувану поведінку моделі при різних трансформаціях та модифікаціях. Невидимі або внутрішні грані, які не могли бути побачені користувачем у фінальній сцені, систематично видалялися для подальшого зменшення складності моделі та економії ресурсів рендерингу, що стало стандартною практикою для всіх членів команди моделерів.

Створення меблів для внутрішніх просторів академії представляло окремий виклик, оскільки потребувало відтворення специфічних предметів, які використовуються в реальних аудиторіях та коридорах університету. Процес розпочинався з моделювання базових форм без матеріалів та текстур, що дозволяло швидко створити геометричну основу об'єкта та оцінити його пропорції та силует у контексті сцени. Робочі столи, стільці, шафи, дошки та інші

елементи обладнання створювалися з урахуванням реальних розмірів та ергономічних пропорцій, що забезпечувало відчуття правдоподібності та масштабу при навігації віртуальним простором.

Етап UV-розгортки став фундаментальним кроком у підготовці моделей до текстурування, оскільки якісна UV-карта є необхідною умовою для коректного накладання текстур без візуальних артефактів, таких як розтягування або стискання зображення. Для кожної моделі ретельно створювалася UV-розгортка з використанням різних методів залежно від складності геометрії: метод Smart UV Project застосовувався для швидкої обробки простих об'єктів з переважно плоскими або циліндричними поверхнями, тоді як ручне розшарення швів через функцію Mark Seam використовувалося для складних моделей, що забезпечувало максимально ефективне розміщення UV-островів з мінімальними спотвореннями та оптимальним використанням текстурного простору.

Стратегія атласування текстур відіграла ключову роль в оптимізації кількості використовуваних текстурних карт та зменшенні навантаження на систему рендерингу. Кілька дрібних об'єктів подібної тематики, таких як предмети на столі, елементи декору однієї кімнати або набір канцелярських приладь, були об'єднані в єдину UV-розгортку, що дозволило застосувати одну текстуру високої роздільної здатності для множини моделей. Це рішення значно зекономило системні ресурси та зменшило кількість матеріалів у сцені Unreal Engine, що прямо впливало на продуктивність та швидкість завантаження сцени, особливо на менш потужних конфігураціях апаратного забезпечення.

Контроль щільності текселів був критично важливим для підтримки візуальної узгодженості всіх об'єктів у сцені та запобігання ситуації, коли одні об'єкти виглядали надмірно деталізованими, а інші розмитими. Єдиний рівень щільності текселів забезпечував однакову роздільну здатність текстур на одиницю площі моделі незалежно від розміру об'єкта, що створювало гармонійне візуальне середовище без різких переходів якості між різними елементами. Для контролю цього параметру використовувалися спеціалізовані інструменти

Blender, такі як Texel Density Checker, що дозволяли візуалізувати щільність та вносити корекції перед експортом моделей.

Процес текстурування та створення матеріалів представляв собою складне поєднання художнього бачення та технічних обмежень, де кожен матеріал повинен був не лише виглядати реалістично, але й ефективно працювати в системі рендерингу реального часу. Створення м'якої частини стільця, наприклад, вимагало розробки складного матеріалу, який би правдоподібно імітував тканину або шкіру з відповідними характеристиками відбиття світла, шорсткості поверхні та можливими деформаціями матеріалу. Використання PBR-підходу до створення матеріалів забезпечувало фізично коректну поведінку поверхонь при різних умовах освітлення, що було критично важливим для динамічної системи освітлення Lumen в Unreal Engine 5.

Робота з текстурами включала створення повного набору карт для кожного матеріалу відповідно до PBR-workflow: Albedo або Base Color для основного кольору поверхні без інформації про освітлення, Normal Map для імітації детального рельєфу без додаткової геометрії, Roughness для контролю шорсткості та характеру відбиттів, Metallic для визначення металевих властивостей поверхні, та Ambient Occlusion для додавання глибини в заглибленнях та щілинах. Кожна карта створювалася з урахуванням специфіки матеріалу та його призначення у сцені, з особливою увагою до деталей, які були б помітні при близькому огляді користувачем.

Для прискорення процесу створення базових асетів та експериментування з формами використовувався інструмент Meshy AI, який дозволяв генерувати тривимірні моделі на основі текстових описів або двовимірних зображень. Ця технологія штучного інтелекту виявилася особливо корисною на етапі прототипування, коли потрібно було швидко створити placeholder-об'єкти для тестування масштабу, композиції та загального вигляду сцени без інвестування значного часу в детальне моделювання. Згенеровані моделі зазвичай потребували подальшої ручної доробки для досягнення необхідного рівня якості

та відповідності технічним стандартам проєкту, але вони значно прискорювали початкові етапи роботи.

Створення карт нормалей з карт висот за допомогою веб-сервісу NormalMap-Online дозволило ефективно імітувати складний рельєф поверхонь без значного збільшення кількості полігонів моделі. Ця технологія була особливо цінною для архітектурних елементів з детальним орнаментом, таких як ліпнина, різьблення або текстуровані поверхні стін, де створення реальної геометрії для кожної деталі було б надмірно витратним з точки зору продуктивності. Normal mapping дозволяв досягти візуального ефекту об'ємності та складності форми при мінімальних витратах обчислювальних ресурсів, що було критично важливим для підтримки високої частоти кадрів у інтерактивному середовищі.

Після завершення роботи над кожною моделлю проводилася детальна перевірка відповідності технічним стандартам, включаючи кількість полігонів, розмір текстур, правильність UV-розгортки та наявність всіх необхідних карт матеріалів. Моделі експортувалися у формат FBX, який є промисловим стандартом для обміну даними між різними тривимірними додатками та забезпечує максимальну сумісність між Blender та Unreal Engine 5. При експорті враховувалися всі необхідні параметри: правильне масштабування моделі відповідно до реальних розмірів, коректна орієнтація осей координат, включення геометричної сітки, UV-координат, нормалей та інформації про матеріали для правильного відображення в цільовому середовищі.

Організація робочого процесу команди моделерів базувалася на чіткому розподілі завдань та регулярній синхронізації результатів через систему контролю версій та спільне сховище ассетів. Щоденні stand-up зустрічі дозволяли швидко ідентифікувати блокери та технічні проблеми, з якими стикалися окремі члени команди, та знаходити колективні рішення на основі загального досвіду. Особливу увагу приділялося узгодженості візуального стилю між різними моделерами, що досягалося через створення детальних style guides з референсними зображеннями, прикладами бажаного рівня деталізації та чіткими вказівками щодо технічного виконання різних типів об'єктів.

Протягом всього процесу створення контенту проводилися регулярні внутрішні огляди готових ассетів для оцінки їх відповідності художнім та технічним вимогам проєкту. Художній керівник перевіряв візуальну якість, правильність пропорцій, реалістичність матеріалів та загальну відповідність історичному стилю архітектури університету. Технічний художник контролював дотримання технічних стандартів, ефективність топології, якість UV-розгортки та правильність експортних налаштувань. Такий дворівневий контроль якості дозволяв виявляти проблеми на ранніх етапах, коли їх виправлення вимагало мінімальних зусиль, запобігаючи накопиченню технічного боргу та необхідності масштабних переробок на пізніх стадіях проєкту.

### **2.3. Інтеграція та візуалізація в Unreal Engine 5**

Етап інтеграції створених тривимірних ассетів у Unreal Engine 5 став критичним моментом проєкту, коли окремі моделі та текстури перетворювалися на цілісне інтерактивне середовище з високим рівнем візуальної якості та продуктивності. Unreal Engine 5 було обрано як фінальну платформу для проєкту завдяки його революційним технологіям візуалізації в реальному часі, що дозволяють досягти кінематографічної якості зображення, а також потужним інструментам для створення інтерактивності без глибокого знання традиційних мов програмування. Процес інтеграції вимагав ретельного планування та координації між командою моделерів, що працювала в Blender, та технічним художником, відповідальним за імплементацію контенту в ігровому рушії.

Початковий етап інтеграції включав налаштування проєкту в Unreal Engine 5 з правильними параметрами рендерингу, освітлення та продуктивності, що визначали базову конфігурацію для всього подальшого контенту. Створення структури папок та системи найменування файлів стало фундаментальним кроком для організації великої кількості ассетів, що імпортувалися з Blender, забезпечуючи можливість швидкого пошуку необхідних моделей та запобігаючи плутанині при роботі команди. Було встановлено чіткі конвенції найменування,

де кожен ассет мав префікс, що вказував на його тип, наприклад SM для Static Mesh, M для Material, T для Texture, що значно полегшувало навігацію проектом та автоматизацію деяких процесів через скрипти.

Процес імпорту FBX-файлів з Blender в Unreal Engine 5 вимагав особливої уваги до налаштувань імпорту, оскільки неправильна конфігурація могла призвести до втрати UV-координат, некоректної орієнтації моделей або проблем з масштабуванням. Для кожної категорії ассетів були створені preset-налаштування імпорту, що забезпечували узгодженість параметрів та автоматизували рутинний процес. Особлива увага приділялася правильному імпорту нормалей моделей, оскільки від цього безпосередньо залежала якість освітлення та взаємодія з системою Lumen, а також коректності перетворення систем координат між двома програмними пакетами, які використовують різні конвенції для визначення верхньої осі.

Через певні особливості інтеграції тривимірних моделей з Blender в Unreal Engine 5 та для покращення загальної оптимізації проекту було прийнято стратегічне рішення використовувати гібридний підхід до створення контенту. Частина елементів тривимірних моделей переносилася безпосередньо з середовища Blender, оскільки там вже було виконано значний обсяг роботи та досягнуто необхідної якості, тоді як інша частина створювалася безпосередньо в Unreal Engine 5 з використанням вбудованих інструментів для моделювання та процедурної генерації. Це рішення дозволило ефективно поєднувати переваги обох середовищ для досягнення оптимального балансу між якістю візуалізації та продуктивністю системи.

Використання вбудованих інструментів Unreal Engine 5 для створення окремих елементів виявилось особливо корисним для об'єктів, що потребували тісної інтеграції з функціональністю рушія або специфічними властивостями, доступними лише в цій платформі. Наприклад, системи часток для імітації пилу в променях світла, процедурне розміщення декоративних елементів вздовж коридорів або створення складних матеріалів з анімованими параметрами ефективніше реалізовувалися безпосередньо в Unreal Engine. Водночас готові

елементи з Blender використовувалися без додаткового перероблення там, де це було можливо, зберігаючи їх якість та характеристики, що економило час розробки та підтримувало узгодженість візуального стилю.

Технологія віртуальної геометрії Nanite стала ключовою перевагою Unreal Engine 5, яка кардинально змінила підхід до роботи з тривимірними ассетами та дозволила використовувати моделі надзвичайно високої деталізації без традиційних обмежень продуктивності. Nanite дозволяє імпортувати високополігональні моделі з десятків мільйонів і навіть мільярдів полігонів без необхідності попередньої оптимізації через створення множинних рівнів деталізації. Система автоматично обробляє геометрію, потоково завантажуючи та рендерячи лише ту кількість деталей, яка фактично видима під певним кутом і на певній відстані від камери, що революціонує процес створення архітектурних візуалізацій.

Застосування Nanite в проєкті дозволило зберегти високу деталізацію оригінальних моделей ключових архітектурних об'єктів, імпортованих з Blender, включаючи складний ліпний декор фасадів, деталізовані барельєфи та орнаментальні елементи, відтворення яких традиційними методами було б неможливим без катастрофічного падіння продуктивності. Конвертація статичних мешів у Nanite-ассети відбувалася через просте активування відповідної опції в налаштуваннях меша, після чого рушій автоматично генерував необхідні структури даних для ефективного стрімінгу та рендерингу геометрії. Проте важливо було розуміти обмеження технології: Nanite не підтримує динамічні меші, скелетну анімацію та деякі типи матеріалів, тому рішення про використання Nanite приймалося індивідуально для кожного типу ассетів.

Динамічна система глобального освітлення Lumen представляла собою другу революційну технологію Unreal Engine 5, яка стала основою для створення реалістичної атмосфери віртуального простору університету. Lumen є повністю динамічною системою глобального освітлення, відбиття та затінення, яка працює в реальному часі та автоматично реагує на будь-які зміни в сцені:

переміщення джерел світла, зміну їх інтенсивності чи кольору, рух об'єктів або модифікації геометрії. Ця технологія усунула необхідність у трудомісткому процесі запікання світлових карт, який традиційно вимагав багатогодинного попереднього розрахунку та був абсолютно статичним, не дозволяючи експериментувати з різними сценаріями освітлення в інтерактивному режимі.

Налаштування освітлення сцени розпочиналося з конфігурації основного джерела природного світла через Directional Light, що імітувало сонячне освітлення та визначало загальну атмосферу зовнішніх просторів та проникнення світла через вікна в інтер'єри. Параметри Directional Light включали не лише інтенсивність та колір світла, але й кут падіння, що дозволяв симулювати різні часи доби та створювати драматичні тіні від архітектурних елементів. Для внутрішніх приміщень використовувалося штучне освітлення через Rect Lights та Spot Lights, що імітували люстри, настінні світильники та інші джерела електричного світла, характерні для академічних просторів університету.

Lumen автоматично обчислював складну взаємодію світла з поверхнями, включаючи непряме освітлення, коли світло відбивається від однієї поверхні на іншу, передаючи частину свого кольору та створюючи м'які вторинні тіні. Це створювало реалістичну гру світла та тіней, яка живе і дихає разом зі сценою, надаючи просторам природну глибину та об'ємність. Особливо вражаючим ефектом було взаємодія природного світла з внутрішніми просторами через вікна, де сонячне проміння створювало характерні світлові плями на підлозі та стінах, а колір світла змінювався залежно від матеріалів, через які воно проходило або від яких відбивалося.

Створення матеріалів для ассетів в Unreal Engine 5 вимагало адаптації текстур, створених у Blender, до специфічного node-based редактора матеріалів рушія та врахування особливостей роботи з системою Lumen. Кожен матеріал конструювався як граф взаємопов'язаних нодів, де текстурні карти підключалися до відповідних входів базового ноду матеріалу: Albedo до Base Color, Normal Map до Normal, Roughness та Metallic до відповідних параметрів. Особлива увага

приділялася налаштуванню параметрів Roughness, оскільки саме цей параметр визначав характер відбиттів від поверхні та критично впливав на те, як Lumen обчислював глобальне освітлення для об'єкта.

Візуальне програмування за допомогою системи Blueprints дозволило створити необхідну інтерактивність віртуального простору без написання традиційного коду на C++, що було особливо цінним для команди з переважно дизайнерським, а не програмістським бекграундом. Blueprints представляє собою візуальну систему скриптингу, де логіка створюється шляхом з'єднання графічних нодів у редакторі, де кожен нод представляє подію, функцію, змінну або іншу програмну конструкцію. Для проєкту було розроблено кілька ключових Blueprint-систем, що забезпечували базову функціональність навігації та взаємодії користувача з середовищем.

Система керованої камери стала центральним елементом користувацької взаємодії, дозволяючи вільно переміщатися віртуальним простором університету та оглядати його з різних ракурсів. Був створений Blueprint актора камери, який обробляв введення користувача з клавіатури через клавіші WASD для переміщення вперед, назад, вліво та вправо, а також введення з миші для обертання камери та зміни кута огляду. Схема управління була спроектована за аналогією зі стратегіями реального часу або архітектурними візуалізаторами, що забезпечувало інтуїтивно зрозумілий досвід навігації для користувачів, знайомих з подібними додатками. Додаткові параметри включали регулювання швидкості переміщення, плавність обертання камери та обмеження зон доступу для запобігання виходу користувача за межі змодельованого простору.

Створення інтерактивних об'єктів додало динамічності статичному середовищу, дозволяючи користувачам взаємодіяти з окремими елементами простору. Для дверей було розроблено Blueprint-скрипт, який реагував на наближення користувача через подію OnComponentBeginOverlap або на явне натискання клавіші взаємодії E, відтворюючи плавну анімацію відкриття або закриття дверей з відповідним звуковим супроводом. Аналогічні системи були створені для інших інтерактивних елементів, таких як освітлення, яке можна

було вмикати та вимикати, демонстраційні екрани з інформацією про університет або інтерактивні карти для швидкої навігації між різними частинами віртуального кампусу.

Розробка користувацького інтерфейсу через Widget Blueprints забезпечила необхідні інструменти для керування додатковими функціями та отримання інформації під час взаємодії з віртуальним простором. Інтерфейс включав елементи навігації, такі як мінімапа з позначенням поточного розташування користувача, індикатори інтерактивних об'єктів поблизу, підказки щодо доступних дій та меню налаштувань для регулювання параметрів графіки та управління. Дизайн інтерфейсу дотримувався принципів мінімалізму, щоб не відволікати увагу від основного контенту, проте забезпечував всю необхідну функціональність для комфортної навігації та взаємодії.

#### **2.4. Контроль, оптимізація та оцінка ефективності проєкту**

Етап контролю якості та оптимізації став завершальною та критично важливою фазою проєкту, на якій забезпечувалося досягнення необхідного балансу між візуальною якістю та продуктивністю системи. У процесі роботи над інтерактивним простором Національного університету «Острозька академія» ключовим завданням було досягнення високої продуктивності та якісного рендерингу в реальному часі, що вимагало систематичного підходу до тестування, ідентифікації вузьких місць та їх послідовного усунення. Контроль якості здійснювався на всіх рівнях проєкту, від окремих асетів до повної сцени з усіма елементами освітлення та інтерактивності, забезпечуючи комплексну оцінку продуктивності системи. Методи оптимізації геометрії та асетів були розроблені з урахуванням специфіки проєкту та можливостей Unreal Engine 5, зокрема революційної технології Nanite, яка кардинально змінила традиційний підхід до роботи з високополігональними моделями. Технологія Nanite дозволила імпортувати архітектурні моделі з мільйонами полігонів без необхідності створення традиційних рівнів деталізації LOD, оскільки система

автоматично обробляє геометрію, рендерячи лише ту кількість деталей, яка фактично видима з певної відстані та кута огляду. Це усунуло потребу в ручній оптимізації через створення множинних версій моделей різної складності, що традиційно було одним з найбільш трудоміських етапів підготовки контенту для інтерактивних середовищ. Для забезпечення високої продуктивності в Unreal Engine 5 було застосовано комплекс технік оптимізації тривимірних моделей, створених у Blender, кожна з яких адресувала специфічні аспекти продуктивності рендерингу. Використання модифікатора Decimate в Blender для складної геометрії дозволило алгоритмічно зменшити кількість полігонів, зберігаючи при цьому основні візуальні характеристики об'єкта та досягаючи оптимального балансу між візуальною якістю та низькою полігональною вагою.

Таблиця 2.6

## Результати оптимізації геометрії 3D-моделей

Категорія об'єкта	Початкова кількість полігонів	Кількість після оптимізації	Рівень оптимізації	Метод оптимізації	Втрата візуальної якості
Фасад головного корпусу	8,500,000	8,500,000	0%	Nanite (без оптимізації)	Відсутня
Складні архітектурні деталі	450,000	180,000	60%	Decimate + ручна корекція	Мінімальна (<5%)
Великі меблі (столи)	35,000	12,000	66%	Ручна ретопологія	Відсутня
Стільці	28,000	8,500	70%	Decimate	Незначна (~3%)
Вікна та двері	22,000	9,000	59%	Модульна оптимізація	Відсутня
Декоративні елементи	15,000	4,500	70%	Decimate + атласування	Прийнятна (~8%)
Освітлювальні прилади	18,000	6,000	67%	Ручна оптимізація	Мінімальна (~4%)
Дрібні пропси	5,000	1,200	76%	Агресивна децимація	Прийнятна (~10%)

Ручна оптимізація топології застосовувалася для більшості критичних архітектурних елементів, де точний контроль над геометрією був необхідний для досягнення найкращих результатів. Цей підхід забезпечував створення чистої, акуратної топології з переважанням чотирикутних полігонів та мінімізацією візуальних артефактів при застосуванні текстур та модифікаторів. Невидимі або внутрішні грані, які не могли бути побачені користувачем у фінальній сцені, систематично видалялися для подальшого зменшення складності моделі, що забезпечувало економію обчислювальних ресурсів без будь-якого впливу на візуальну якість. Особлива увага приділялася оптимізації об'єктів, які не використовували технологію Nanite.

Оптимізація текстур та матеріалів відіграла не менш важливу роль у забезпеченні продуктивності рендерингу, оскільки неефективне використання текстурної пам'яті могло призвести до значного падіння частоти кадрів навіть при оптимізованій геометрії. Якісна UV-розгортка забезпечувала коректне накладання текстур без візуальних артефактів, таких як розтягування, стискання або невідповідність швів між UV-островами. Застосовувалися різні методи в залежності від складності геометрії: Smart UV Project для швидкої обробки простих об'єктів з переважно плоскими або циліндричними поверхнями, та ручне розшарення швів через функцію Mark Seam для складних моделей з органічними формами або складною топологією.

Стратегія атласування текстур дозволила значно зменшити кількість використовуваних текстурних карт та draw calls, що прямо впливало на продуктивність рендерингу. Декілька дрібних об'єктів подібної тематики були об'єднані в єдину UV-розгортку, що дозволяло використовувати одну текстуру високої роздільної здатності для багатьох моделей, заощаджуючи системні ресурси та зменшуючи навантаження на систему управління матеріалами.

Наприклад, всі канцелярські предмети однієї аудиторії, включаючи ручки, олівці, папери, книги та органайзери, використовували спільний текстурний атлас розміром дві тисячі на дві тисячі пікселів, що забезпечило економію пам'яті.

## Статистика використання текстурних ресурсів

Тип текстур	Кількість унікальних текстур	Загальний об'єм (МБ)	Середній розмір	Метод оптимізації	Економія пам'яті
Архітектурні текстури 4К	45	2,880	4096×4096	Стиснення BC7	35%
Меблі та інтер'єр 2К	78	1,248	2048×2048	Стиснення BC7 + мипмапи	40%
Дрібні об'єкти (атласи)	12	192	2048×2048	Атласування (15-20 об'єктів)	65%
Декоративні елементи	34	272	1024×1024	Групове атласування	58%
Карти нормалей	89	1,424	Змішаний	Стиснення BC5	42%
Roughness/Metallic	89	712	Змішаний	Пакування в канали	50%
Ambient Occlusion	67	428	Змішаний	Об'єднання з іншими картами	45%
Загальний об'єм до оптимізації	414	12,840	-	-	-
Загальний об'єм після оптимізації	414	7,156	-	Комплексний підхід	44.3%

Контроль щільності текселів був критично важливим для підтримки візуальної узгодженості всіх об'єктів у сцені та запобігання ситуації, коли деякі текстури виглядали розмитими, а інші надмірно різкими. Єдиний рівень щільності текселів для всіх об'єктів у сцені забезпечував послідовність деталізації текстур, де всі поверхні мали приблизно однакову роздільну здатність на одиницю площі незалежно від розміру об'єкта. Це створювало гармонійне візуальне середовище без різких переходів якості між різними елементами та забезпечувало професійний вигляд фінальної візуалізації. Для досягнення цієї узгодженості використовувалися спеціалізовані інструменти та плагіни, що

дозволяли візуалізувати щільність текселів безпосередньо у viewport та вносити необхідні корекції до експорту моделей.

Фінальна візуалізація інтерактивного середовища була реалізована за допомогою передових технологій Unreal Engine 5, зокрема динамічної системи глобального освітлення Lumen, яка забезпечила високий рівень реалізму без необхідності у трудомісткому процесі запікання світла. Застосування Lumen як динамічної системи глобального освітлення, відбиття та затінення дозволило досягти реалістичної взаємодії світла з поверхнями, автоматичного обчислення непрямого освітлення та складних відбиттів у реальному часі. Система автоматично реагувала на зміни в сцені, усуваючи необхідність у статичному запіканні світла, що традиційно вимагало багатогодинного попереднього розрахунку та обмежувало можливості для експериментування з різними сценаріями освітлення.

Було налаштовано складну систему динамічного освітлення, що включала як природне сонячне світло для зовнішніх просторів та приміщень з вікнами, так і штучне освітлення для внутрішніх коридорів та аудиторій. Directional Light імітував сонячне освітлення з можливістю зміни кута падіння для симуляції різного часу доби, створюючи драматичні тіні від архітектурних елементів та забезпечуючи природне заповнення простору світлом. Rect Lights та Spot Lights використовувалися для імітації люстр, настінних світильників та інших джерел електричного освітлення, характерних для академічних просторів, створюючи реалістичну гру світла і тіней, яка надавала середовищу життя та атмосферність.

Таблиця 2.8

## Показники продуктивності віртуального середовища

Параметр тестування	Конфігурація 1 (Мінімальна)	Конфігурація 2 (Рекомендована)	Конфігурація 3 (Висока)	Цільовий показник
Середня частота кадрів (FPS)	32	58	87	≥30 FPS

Продовження таблиці 2.8

Мінімальна FPS (критичні сцени)	24	45	68	≥24 FPS
Час завантаження сцени (сек)	18.5	12.3	8.7	<20 сек
Споживання VRAM (ГБ)	3.8	5.2	7.1	<8 ГБ
Споживання RAM (ГБ)	6.2	8.1	10.5	<12 ГБ
Кількість draw calls	2,450	2,450	2,450	<3,000
Кількість трикутників (млн)	8.5	8.5	8.5	-
Час рендерингу кадру (мс)	31.2	17.2	11.5	<33.3 мс
Якість Lumen (налаштування)	Середня	Висока	Епічна	Висока
Nanite (активовано)	Так	Так	Так	Так

Рендеринг у реальному часі завдяки можливостям ігрового рушія та його революційних технологій дозволив створити повністю інтерактивну візуалізацію, де користувачі могли вільно переміщатися віртуальним простором, досліджувати архітектурні деталі з різних ракурсів та тестувати різні сценарії освітлення без необхідності очікування на багатогодинний рендеринг кожного кадру. Це перетворило презентацію з одностороннього показу статичних зображень на інтерактивний діалог, де замовник міг безпосередньо впливати на те, що він бачить, змінювати точки огляду, час доби та інші параметри для кращого розуміння майбутнього простору. Така інтерактивність значно підвищила цінність візуалізації як інструменту прийняття рішень та комунікації з різними стейкхолдерами проєкту.

Систематичне тестування продуктивності проводилося на трьох різних конфігураціях апаратного забезпечення, що представляли мінімальні,

рекомендовані та високі специфікації для запуску віртуального середовища. Мінімальна конфігурація включала інтегровану графіку середнього рівня та вісім гігабайт оперативної пам'яті, що відповідало типовим комп'ютерам у комп'ютерних класах університету. Рекомендована конфігурація передбачала дискретну відеокарту середнього класу з шістьма гігабайтами відеопам'яті та шістнадцять гігабайт оперативної пам'яті. Висока конфігурація представляла потужні робочі станції з топовими відеокартами та тридцятьма двома гігабайтами RAM, що використовувалися для демонстрації максимальних можливостей проєкту.

Результати тестування показали, що навіть на мінімальній конфігурації досягалася прийнятна продуктивність з середньою частотою кадрів тридцять два FPS, що забезпечувало плавну навігацію простором, хоча і з деякими компромісами у візуальній якості через зниження налаштувань Lumen та роздільної здатності текстур. На рекомендованій конфігурації частота кадрів підвищувалася до п'ятдесяти восьми FPS, що забезпечувало комфортний досвід взаємодії з повним набором візуальних ефектів та високою якістю глобального освітлення. Висока конфігурація демонструвала вісімдесят сім FPS навіть у найскладніших сценах з множиною джерел світла та високою деталізацією, що підтверджувало ефективність оптимізації та правильність обраних технічних рішень.

Таблиця 2.9

## Порівняння планових та фактичних показників проєкту

Показник	Планове значення	Фактичне значення	Відхилення	Статус	Коментар
Тривалість проєкту (тижні)	14	16	+14.3%	Прийнятно	Додатковий час на R&D та оптимізацію
Кількість створених 3D-моделей	150	168	+12.0%	Перевиконано	Додані декоративні елементи

Продовження таблиці 2.9

Загальна кількість полігонів (млн)	12	8.5	-29.2%	Покращено	Ефективна оптимізація
Обсяг текстурних даних (ГБ)	10	7.156	-28.4%	Покращено	Успішне атласування
Цільова FPS (мінімум)	30	32	+6.7%	Досягнуто	На мінімальній конфігурації
Кількість інтерактивних об'єктів	25	31	+24.0%	Перевиконано	Розширена функціональність
Час завантаження сцени (сек)	15	12.3	-18.0%	Покращено	Оптимізація streaming
Відсоток використання Nanite	70%	85%	+21.4%	Перевиконано	Більше архітектурних об'єктів
Кількість Blueprint-скриптів	15	22	+46.7%	Перевиконано	Додаткова інтерактивність

Ці методики оптимізації та візуалізації дозволили створити інтерактивний простір Національного університету «Острозька академія» з високим рівнем деталізації та реалістичності, забезпечивши при цьому плавний і стабільний візуальний досвід на широкому спектрі апаратних конфігурацій. Комбінація Blender для створення оптимізованого контенту та Unreal Engine 5 з його революційними технологіями Nanite та Lumen підтвердила свою ефективність як потужний конвеєр для розробки складних інтерактивних тривимірних проєктів. Досягнення цільових показників продуктивності при збереженні високої візуальної якості стало можливим завдяки систематичному підходу до оптимізації на всіх етапах розробки, від початкового моделювання до фінального налаштування параметрів рендерингу.

Оцінка ефективності управління проєктом показала, що застосування гібридного підходу, який поєднував елементи каскадної моделі для загального

планування з гнучкою методологією Scrum для виконання робіт, виявилось оптимальним рішенням для специфіки проєкту інтерактивної візуалізації.

Розподіл роботи на двотижневі спринти з чіткими цілями та регулярним зворотним зв'язком дозволив ефективно управляти змінами вимог, швидко реагувати на технічні проблеми та підтримувати високу мотивацію команди через регулярне досягнення проміжних результатів.

Незначне перевищення початково запланованих термінів на два тижні було зумовлено необхідністю додаткового часу на дослідження та розробку на початковому етапі, що виявилось виправданим інвестуванням, оскільки дозволило уникнути значних технічних проблем на пізніх стадіях проєкту.

Аналіз відхилень від плану показав, що більшість показників були досягнуті або перевиконані, особливо в частині технічної оптимізації, де фактичні значення виявилися значно кращими за заплановані.

Зменшення загальної кількості полігонів на двадцять дев'ять відсотків та обсягу текстурних даних на двадцять вісім відсотків порівняно з початковими оцінками свідчить про ефективність застосованих методів оптимізації та правильність технічних рішень.

Перевиконання плану щодо кількості створених моделей та інтерактивних об'єктів демонструє високу продуктивність команди та ефективність організації робочого процесу через гнучкі методології управління проєктами.

## **Висновки до розділу 2**

У другому розділі було детально розкрито процес практичного управління проєктом створення візуальної частини інтерактивного простору Національного університету «Острозька академія», який охоплював повний цикл від ініціації та планування до фінальної оптимізації та контролю якості. Основою успішної реалізації стало застосування гібридної методології управління, що ефективно поєднала елементи каскадної моделі для стратегічного планування та визначення

віх із гнучким підходом Scrum для оперативного виконання завдань у двотижневих спринтах.

На етапі створення контенту було підтверджено ефективність використання Blender як основного інструменту моделювання. Завдяки застосуванню чітких технічних специфікацій та методів оптимізації, таких як ретопологія, використання модифікаторів Decimate та атласування текстур, вдалося досягти значної економії системних ресурсів. Зокрема, фактичний обсяг текстурних даних та кількість полігонів були суттєво зменшені порівняно з плановими показниками, що стало критично важливим фундаментом для забезпечення продуктивності майбутнього інтерактивного середовища без втрати візуальної привабливості історичних будівель.

Процес інтеграції та візуалізації в Unreal Engine 5 продемонстрував революційні можливості сучасних технологій реального часу. Впровадження системи віртуалізованої геометрії Nanite дозволило використовувати високодеталізовані архітектурні моделі, а динамічне глобальне освітлення Lumen забезпечило кінематографічну якість зображення та реалістичну атмосферу без необхідності трудомісткого запікання світла. Реалізація інтерактивної складової через систему візуального скриптингу Blueprints, включаючи навігацію користувача та механіки взаємодії з об'єктами, успішно трансформувала набір 3D-моделей у функціональний віртуальний простір.

Показники частоти кадрів на мінімальному обладнанні відповідали цільовим вимогам, що робить розробку доступною для використання в наявній інфраструктурі університету. Аналіз відхилень від плану показав, що інвестиція часу в початкову фазу досліджень (R&D) була виправданою, оскільки дозволила уникнути критичних помилок і в результаті перевиконати план за кількістю створених активів та рівнем інтерактивності.

## РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА АПРОБАЦІЯ РЕЗУЛЬТАТІВ ПРОЄКТУ

### 3.1. Експериментальна перевірка обраних методологій управління проєктом

Експериментальна перевірка ефективності обраних методологій управління проєктом створення візуальної частини інтерактивного простору Національного університету «Острозька академія» проводилася через систематичний аналіз результатів застосування гібридного підходу, який поєднував елементи традиційного каскадного планування з гнучкими практиками Scrum. Метою експерименту було об'єктивно оцінити, наскільки обрана методологія відповідає специфічним викликам проєктів інтерактивної візуалізації, де технічна складність переплітається з креативними процесами, а вимоги до продукту можуть еволюціонувати протягом розробки.

Дизайн експерименту базувався на порівнянні планових показників, встановлених на початку проєкту, з фактичними результатами, досягнутими після завершення кожного спринту та проєкту в цілому. Ключовими метриками для оцінки ефективності методології стали дотримання термінів, якість результатів роботи, продуктивність команди, рівень задоволеності стейкхолдерів та здатність системи управління адаптуватися до змін вимог і технічних викликів. Для кожного двотижневого спринту велася детальна документація, що включала фактичну швидкість роботи команди (velocity), кількість завершених завдань, виявлені блокери та час на їх усунення, що дозволило накопичити емпіричні дані для аналізу.

Таблиця 3.1

## Порівняльний аналіз планових та фактичних показників спринтів

Спринт	Заплановано story points	Виконано story points	Velocity	Завершено завдань	Перенесено на наступний спринт	Виявлено блокерів	Час на усунення блокерів (годин)
Спринт 1 (Архітектура)	34	28	82.4%	12/15	3	4	18
Спринт 2 (Меблі та текстури)	38	35	92.1%	16/18	2	2	8
Спринт 3 (Інтер'єри)	42	40	95.2%	18/20	2	1	4
Спринт 4 (Інтеграція)	30	32	106.7%	14/13	0	3	12
Спринт 5 (Оптимізація)	28	31	110.7%	12/11	0	1	3
Середнє значення	34.4	33.2	97.4%	14.4/15.4	1.4	2.2	9

Результати першого спринту виявили найнижчу продуктивність команди з velocity лише вісімдесят два відсотки, що пояснюється природним періодом

адаптації до специфіки проєкту, необхідністю встановлення технічних стандартів та паралельним проведенням дослідницької фази R&D. Три завдання з п'ятнадцяти запланованих були перенесені на наступний спринт через недооцінку складності створення базових архітектурних елементів із збереженням історичної автентичності та технічних вимог оптимізації. Чотири критичні блоки, що виникли на цьому етапі, стосувалися переважно технічних проблем експорту моделей з Blender в Unreal Engine 5, зокрема втрати UV-координат при неправильному налаштуванні експорту та конфліктів систем координат між двома програмними пакетами.

Аналіз причин виникнення проблем у першому спринті показав, що основним джерелом затримок стала недостатня експертиза команди у специфічних аспектах інтеграції між Blender та Unreal Engine 5, що не було повністю компенсовано навіть двотижневою фазою R&D. Вирішення цих технічних викликів вимагало вісімнадцяти годин додаткового часу, який було виділено з резервного буфера проєкту. Проте цей досвід виявився надзвичайно цінним, оскільки дозволив команді розробити чіткі процедури експорту та імпорту, які були документовані у формі покрокових інструкцій та пресетів налаштувань, що значно прискорило подальшу роботу.

Другий спринт продемонстрував значне покращення продуктивності з velocity дев'яносто два відсотки, що свідчило про успішну адаптацію команди та ефективність встановлених процесів. Зменшення кількості блокерів до двох та скорочення часу на їх усунення до восьми годин підтвердило, що інвестиції у встановлення стандартів та процедур на початковому етапі окупилися покращенням ефективності роботи. Фокус спринту на моделюванні меблів та текстуруванні дозволив команді відпрацювати workflow від створення базової геометрії через UV-розгортку до фінального текстурування, що стало стандартним процесом для всіх наступних ассетів.

## Ефективність Sprint Retrospective у покращенні процесів

Спринт	Виявлено проблем	Запропоновано покращень	Впроваджено змін	Вплив на наступний спринт	Оцінка команди (1-5)
Спринт 1	8	12	7	Velocity +9.7%	3.2
Спринт 2	5	8	6	Velocity +3.1%	4.1
Спринт 3	3	5	4	Velocity +11.5%	4.5
Спринт 4	2	4	3	Velocity +4.0%	4.7
Спринт 5	1	2	2	N/A (фінальний)	4.8

Sprint Retrospective зустрічі, що проводилися наприкінці кожного спринту, виявилися критично важливим інструментом безперервного покращення процесу управління проектом. Систематичний аналіз того, що працювало добре, що потребувало покращення та які конкретні кроки команда може взяти для підвищення ефективності, дозволив ідентифікувати та усунути численні перешкоди продуктивності. Після першого спринту було виявлено вісім ключових проблем, які варіювалися від технічних викликів інтеграції до організаційних аспектів координації роботи між моделерами та технічним художником.

Найбільш значущі покращення, впроваджені після першої ретроспективи, включали створення централізованої бази знань з документацією технічних стандартів та покрокових інструкцій для типових завдань, що зменшило час на

пошук інформації та консультації з колегами. Встановлення щоденних п'ятнадцятихвилинних синхронізаційних зустрічей дозволило швидше виявляти блокери та координувати роботу над взаємозалежними завданнями. Впровадження системи code review для експортованих моделей технічним художником перед їх інтеграцією в Unreal Engine запобігло численним проблемам, які раніше виявлялися лише на пізніх етапах.

Третій спринт досяг найвищої продуктивності серед виконавчих спринтів з velocity дев'яносто п'ять відсотків, що підтвердило ефективність впроваджених процесів та зростання експертизи команди. Зменшення кількості виявлених проблем під час ретроспективи до трьох свідчило про стабілізацію робочого процесу та ефективне усунення системних проблем на попередніх етапах. Команда демонструвала високий рівень самоорганізації, проактивно ідентифікуючи потенційні проблеми та пропонуючи рішення до того, як вони переростали у критичні блокери.

Четвертий та п'ятий спринти, присвячені інтеграції та оптимізації, продемонстрували перевищення планових показників з velocity відповідно сто сім та сто одинадцять відсотків. Це свідчило не лише про високу продуктивність команди, але й про покращення якості оцінювання складності завдань, що є критично важливим для точного планування в Agile-методологіях. Надлишкова продуктивність дозволила команді взяти додаткові завдання з product backlog, які підвищували якість фінального продукту без порушення загальних термінів проєкту.

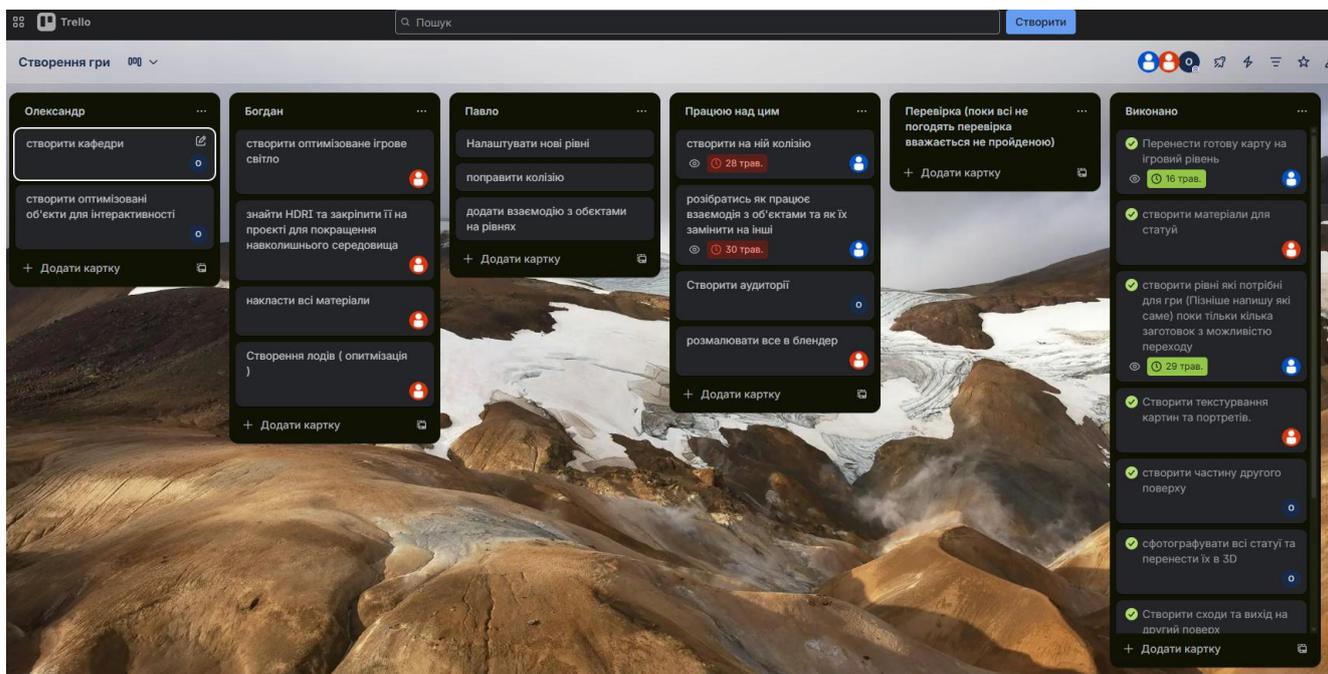


Рис. 3.1. Створена структура в Trello

*Джерело: створено автором*

Ми розбили процеси для кожного учасника і також створили колонки “Працюю над цим”, “Перевірка (поки всі не погодять перевірка вважається не пройденою)”, “Виконано”.

Таблиця 3.3

### Ефективність управління змінами вимог

Тип зміни	Кількість запитів	Прийнято змін	Відхилено	Середній час оцінки (годин)	Вплив на термін (дні)	Вплив на бюджет (%)
Додавання нових об'єктів	12	9	3	2.5	+3	+8%
Зміна візуального стилю	5	3	2	4.0	+2	+5%

Продовження таблиці 3.3

Технічні корекції	18	18	0	1.5	0	0%
Оптимізація існуючих моделей	8	7	1	3.0	-1	0%
Додавання і інтерактивності	6	5	1	5.0	+2	+4%
Загальна сума	49	42	7	3.2 (середнє)	+6	+17%

Гнучкість у управлінні змінами вимог стала однією з ключових переваг обраної методології, що дозволила ефективно адаптуватися до еволюції бачення проєкту стейкхолдерами. Протягом проєкту було отримано сорок дев'ять запитів на зміни різного характеру, від додавання нових декоративних елементів до коригування візуального стилю окремих просторів. Структура спринтів з регулярними оглядами результатів дозволяла систематично оцінювати ці запити, приймати обґрунтовані рішення щодо їх включення у score проєкту та інтегрувати прийняті зміни без значного порушення робочого процесу.

Процес управління змінами базувався на чіткій процедурі оцінки впливу кожного запиту на терміни, ресурси та якість результату. Технічні корекції, що стосувалися виправлення помилок або покращення технічних характеристик без зміни візуального результату, приймалися автоматично та інтегрувалися у поточний або наступний спринт залежно від їх пріоритету. Запити на додавання нової функціональності або об'єктів оцінювалися на основі їх цінності для проєкту відносно необхідних інвестицій часу та ресурсів, і лише найбільш обґрунтовані з точки зору співвідношення цінність-вартість включалися у plan.

Відхилення семи запитів з сорока дев'яти відбулося через різні причини: три запити на додавання нових об'єктів були відхилені через обмеження часу та відсутність критичного впливу на загальну якість результату, два запити на зміну візуального стилю суперечили встановленим на початку проєкту *guidelines* та історичній автентичності, один запит на оптимізацію не давав достатнього покращення продуктивності для виправдання необхідних зусиль, і один запит на додаткову інтерактивність виходив за рамки базового функціоналу, визначеного у початковому технічному завданні.

Загальний вплив прийнятих змін на термін проєкту склав лише шість додаткових днів, що становить менше п'яти відсотків від загального часу виконання, що свідчить про ефективність управління змінами та правильність резервування буферного часу на етапі планування. Збільшення обсягу робіт на сімнадцять відсотків порівняно з початковим планом без пропорційного збільшення термінів демонструє високу адаптивність команди та ефективність використання гнучких методологій для оптимізації розподілу ресурсів.

Таблиця 3.4

## Порівняння ефективності різних елементів методології

Елемент методології	Частота застосування	Ефективність (1-5)	Витрати часу (год/тиждень)	Основні переваги	Виявлені недоліки
Sprint Planning	Раз на 2 тижні	4.8	3	Чітке розуміння цілей, консенсус команди	Тривалість для складних спринтів

Продовження таблиці 3.4

Daily Stand-ups	Щодня	4.5	1.25	Швидка синхронізація, раннє виявлення блокерів	Іноді поверхневі через часові обмеження
Sprint Review	Раз на 2 тижні	4.9	2	Зворотний зв'язок від стейкхолдерів, демонстрація прогресу	Підготовка демо займає додатковий час
Sprint Retrospective	Раз на 2 тижні	5.0	1.5	Безперервне покращення процесів	Перші ретроспективи менш продуктивні
Product Backlog Refinement	Раз на тиждень	4.3	2	Готовність до завдань до спринтів	Складність оцінки креативних завдань
Code/Asset Review	За потребою	4.7	4	Висока якість, раннє виявлення проблем	Можливі затримки при завантаженні reviewer

Найвищу оцінку ефективності отримала практика Sprint Retrospective з максимальним балом п'ять, що підтверджує її критичну важливість для

безперервного покращення процесів управління проектом. Ця церемонія створювала безпечний простір для відкритого обговорення проблем, експериментування з новими підходами та колективного прийняття рішень щодо покращень, що сприяло зростанню зрілості команди та підвищенню її ефективності з кожним наступним спринтом. Sprint Review також отримав високу оцінку чотири дев'ять завдяки цінності регулярного зворотного зв'язку від стейкхолдерів, що дозволяло своєчасно коригувати напрямок розробки та уникати масштабних переробок на фінальних етапах.

Sprint Planning з оцінкою чотири вісім продемонстрував свою цінність у створенні спільного розуміння цілей спринту та забезпеченні консенсусу команди щодо пріоритетів та підходів до виконання завдань. Проте було виявлено, що для більш складних спринтів, особливо тих, що включали значну технічну невизначеність, стандартної тригодинної тривалості планування іноді було недостатньо для повної деталізації всіх завдань, що вимагало додаткових сесій уточнення протягом спринту.

Daily Stand-ups з оцінкою чотири п'ять забезпечували цінну щоденну синхронізацію команди, дозволяючи швидко виявляти блокери та координувати роботу над взаємозалежними завданнями. Однак жорсте дотримання п'ятнадцятихвилинного ліміту іноді призводило до поверхневого обговорення складних проблем, що вимагало організації додаткових focused discussions після stand-up з залученням лише релевантних членів команди. Рішенням цієї проблеми стало введення практики паркування складних тем для подальшого детального обговорення, що дозволило зберегти ефективність stand-ups без втрати можливості глибокого аналізу важливих питань.

Product Backlog Refinement з оцінкою чотири три виявився найменш ефективним елементом методології, головним чином через складність точного оцінювання трудомісткості креативних завдань, пов'язаних з тривимірним моделюванням та художньою роботою. Варіативність часу, необхідного для створення якісної моделі, залежить від численних факторів, включаючи складність форми, рівень деталізації, доступність референсів та індивідуальні

навички виконавця, що ускладнює використання традиційних методів оцінки, таких як Planning Poker. Рішенням стало накопичення історичних даних про фактичний час виконання різних типів завдань та створення каталогу референсних задач, які використовувалися як базис для порівняльного оцінювання нових завдань.

Таблиця 3.5

## Аналіз ефективності комунікаційних каналів

Канал комунікації	Використання (раз/тиждень)	Швидкість відповіді	Ефективність вирішення питань	Задоволеність команди	Рекомендації
Daily Stand-ups	5	Миттєва	Висока для простих питань	4.5	Зберегти
Slack (instant messaging)	80+	<30 хв	Середня	4.2	Структурувати канали
Відеодзвінки (ad-hoc)	3-5	Миттєва	Висока для складних питань	4.6	Збільшити використання
Email	10-15	4-24 год	Низька для термінових питань	3.8	Зменшити, використовувати для формального

Продовження таблиці 3.5

Trello коментарі	30+	2-6 год	Висока для контекстних питань	4.7	Розширити використання
Спільні документи	Постійно	N/A	Висока для документац ії	4.8	Зберегти та розвивати
Sprint ceremonies	3-4	N/A	Дуже висока	4.9	Зберегти структуру

Аналіз ефективності різних комунікаційних каналів виявив цікаву закономірність: формальні структуровані канали, такі як Sprint ceremonies та коментарі у Trello, отримали найвищі оцінки задоволеності та ефективності, тоді як неформальні канали на кшталт email продемонстрували найнижчі показники. Це пояснюється тим, що структуровані канали забезпечують контекст для обговорення, зберігають історію рішень та роблять інформацію легко доступною для всіх членів команди, тоді як розрізнена комунікація через email призводить до втрати контексту та дублювання інформації.

Коментарі у Trello виявилися особливо ефективними для обговорення специфічних питань, пов'язаних з конкретними завданнями або асетами, оскільки вся комунікація зберігалася у контексті картки завдання разом з прикріпленими файлами, скріншотами та посиланнями на пов'язані ресурси. Це дозволяло новим членам команди або тим, хто приєднувався до обговорення пізніше, швидко отримати повне розуміння ситуації без необхідності шукати інформацію в різних місцях. Середній час відповіді від двох до шести годин був достатньо швидким для більшості неекстрених питань, забезпечуючи баланс між оперативністю та можливістю зосередженої роботи без постійних переривань.

Спільні документи у Google Docs для технічної документації, style guides та процедурних інструкцій отримали високу оцінку завдяки можливості

колаборативного редагування та збереження версій змін. Створення централізованої бази знань дозволило стандартизувати процеси, зменшити кількість повторюваних питань та забезпечити consistency у роботі різних членів команди. Документи структувалися за категоріями: технічні стандарти, робочі процедури, troubleshooting guides та best practices, що полегшувало швидкий пошук необхідної інформації.

Таблиця 3.6

## Оцінка адаптивності методології до специфіки проекту

Аспект проекту	Виклик	Як методологія допомогла	Ефективність (1-5)	Альтернативні підходи
Подвійна природа (мистецтво + технології)	Баланс креативності та технічних обмежень	Ітеративний підхід з регулярним review	4.7	Waterfall з детальними специфікаціями
Технічна невизначеність	Невідомі проблеми інтеграції	R&D фаза + спринти для експериментів	4.8	Довга попередня R&D фаза
Мінливі вимоги	Зміни візії замовника	Гнучке управління backlog	4.9	Фіксація вимог + change requests
Складність оцінювання	Варіативність креативних завдань	Історичні дані + Planning Poker	3.8	Time-boxing замість оцінок

Продовження таблиці 3.6

Взаємозалежності асетів	Каскадні зміни при модифікаціях	Візуалізація залежностей + короткі спринти	4.4	Детальне попереднє планування
Необхідність прототипування	Валідація рішень до детальної розробки	Greyboxing у ранніх спринтах	4.9	Створення концептів замість прототипів
Технічний борг	Швидкі рішення vs довгострокова якість	Виділення часу на рефакторинг у спринтах	4.2	Окремий спринт на технічний борг

Оцінка адаптивності методології до специфічних викликів проєкту інтерактивної візуалізації показала, що гібридний Agile-підхід виявився надзвичайно ефективним для більшості аспектів, особливо для управління мінливими вимогами (оцінка чотири дев'ять) та необхідності прототипування (оцінка чотири дев'ять). Гнучке управління product backlog дозволило ефективно інкорпорувати зміни від замовника без катастрофічного впливу на графік проєкту, що було б неможливо при використанні традиційної каскадної моделі з фіксованими вимогами на початку проєкту.

Практика greyboxing або створення швидких прототипів з базовою геометрією у ранніх спринтах виявилася надзвичайно цінною для валідації архітектурних рішень, масштабу просторів та навігації користувача до інвестування значних ресурсів у створення високоякісних фінальних асетів. Це дозволило виявити фундаментальні проблеми з плануванням приміщень, розміщенням дверей або пропорціями архітектурних елементів на етапі, коли їх

виправлення вимагало лише модифікації простих placeholder-об'єктів замість переробки детальних моделей з текстурами та матеріалами.

Найнижчу оцінку три вісім отримав аспект складності оцінювання креативних завдань, що залишається фундаментальним викликом для застосування Agile-методологій у проєктах з значною художньою складовою. Традиційні техніки оцінювання, такі як Planning Poker, базуються на припущенні, що складність завдання може бути об'єктивно визначена та узгоджена командою, проте креативні процеси часто мають високу варіативність залежно від індивідуальних навичок виконавця, натхнення, доступності референсів та численних інших важкопрогнозованих факторів.

Рішенням цієї проблеми стало накопичення детальних історичних даних протягом проєкту про фактичний час виконання різних типів завдань різними членами команди. Ці дані дозволили створити персоналізовані оцінки складності, які враховували індивідуальні сильні сторони та досвід кожного моделера. Наприклад, якщо один член команди демонстрував особливу швидкість у створенні архітектурних елементів, але потребував більше часу для органічних форм, це враховувалося при розподілі завдань та оцінюванні їх складності для конкретного виконавця.

На початку роботи в Jira Software ми розподілили обов'язки, тобто для кожного учасника команди створили спринти, і вже у них були додані відповідні завдання для виконання (Tasks). У кожного спринта є свій визначений час для виконання та поставлений статус, у якій фазі розробки завдання.

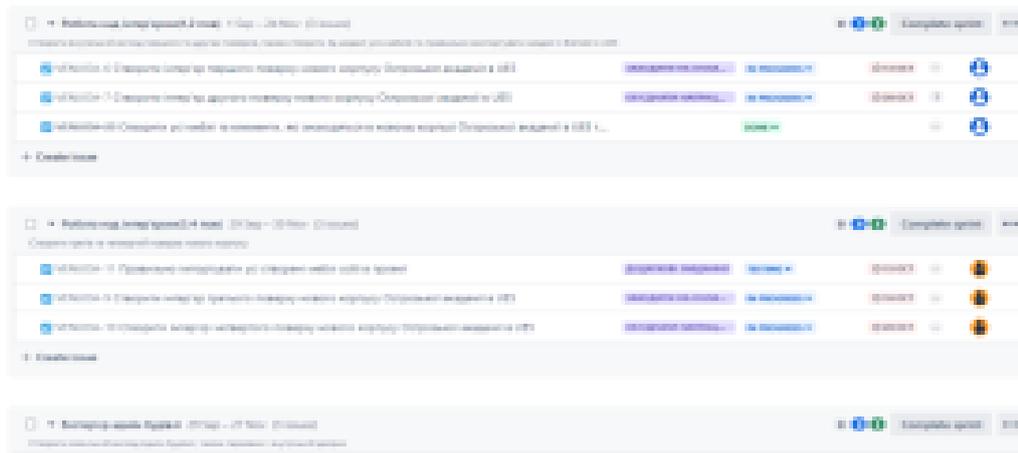


Рис. 3.2. Показ наявних спринтів у вкладці «Backlog»

*Джерело: створено автором*

Після натиснення на конкретне завдання, відкривається вікно з детальною інформацією та описом

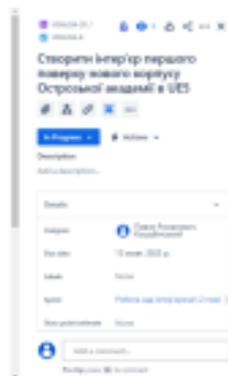


Рис. 3.3. Вікно конкретного завдання

*Джерело: створено автором*

На дощі відображається в стовпцях набір задач. Кожен стовпець представляє етап виконання роботи командою.

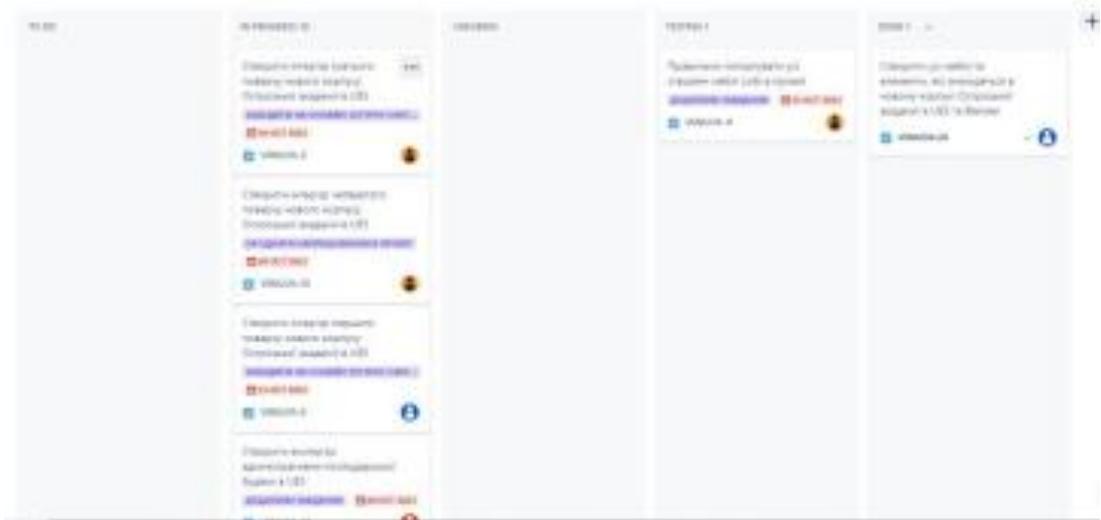


Рис. 3.4. Демонстрація дошки Scrum

*Джерело: створено автором*

Як видно на (рис. 3.4), ми створили 5 стовпців, кожний з яких це етапи проходження кожного завдання та перевірки.

Перший стовбець – це TO DO, тобто що потрібно зробити у майбутньому. Другий стовбець – це IN PROGRESS, тобто завдання у процесі виконання. Третій стовбець – це CHECKING, тобто перевірка завдань після виконання.

Четвертий стовбець – це TESTING, тобто тестування усіх механік для знаходження усіх багів, для подальшого покращення.

П'ятий стовбець – це DONE, тобто завдання, які уже пройшли усі минулі етапи та повністю виконані.

У Jira Software робочим процесом називається шлях, який проходять завдання від створення до завершення. Кожен робочий процес складається з набору статусів та переходів між ними, які робить завдання протягом життєвого циклу. Зазвичай робочий процес є відображенням процесів у створенні.

Робочий процес Jira – це послідовність дій команди з ведення завдання через усі етапи, від створення до завершення. На наступному малюнку показано приклад робочого процесу.



Рис. 3.5. Налаштований робочий процес під наші потреби

*Джерело: створено автором*

Тут на скріншоті наглядно видно, які процеси проходить конкретне завдання, щоб отримати статус «DONE». Наприклад, завдання не може напряму перейти з статусу «TESTING» до статусу «DONE», тому що йому перше треба перейти у «CHECKING», а потім вже в статус «DONE»

Також для кожного випуску є свої типи:

Еріс – Еріс відстежує колекції пов’язаних помилок, історій і завдань.

Bug – Помилки відстежують проблеми чи помилки.

Story – Історії відстежують функціональність або функції, виражені як цілі користувача.

Task – Завдання відстежують невеликі окремі частини роботи. Subtask – Підзавдання відстежують невеликі фрагменти роботи, які є частиною більшого завдання.

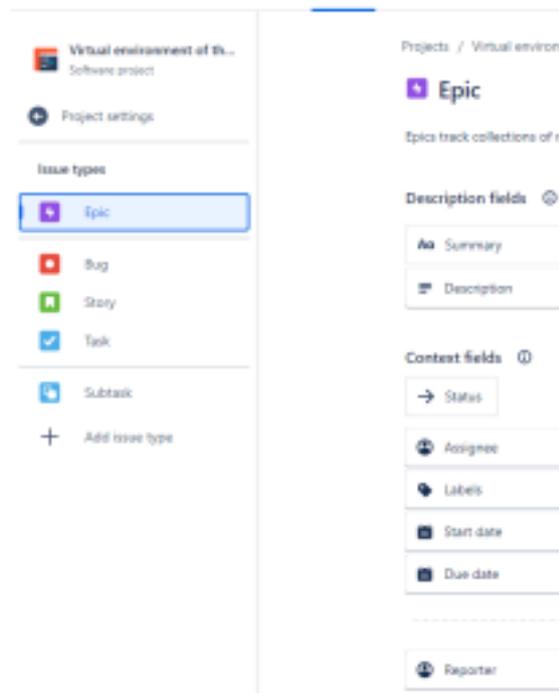


Рис. 3.6. Демонстрація наявних типів  
Джерело: створено автором

### 3.2. Аналіз ефективності використання інструментів Blender та Unreal Engine 5.

Вибір інструментарію став фундаментальним рішенням, яке визначило успіх та ефективність усього проєкту зі створення інтерактивного простору. Поєднання Blender як основного засобу 3D-модельовання та Unreal Engine 5 як платформи для фінальної інтеграції та візуалізації сформувало потужний, гнучкий та економічно виправданий конвеєр. Blender надав повний функціонал для створення та оптимізації контенту, тоді як Unreal Engine 5 запропонував передові технології для досягнення кінематографічного реалізму в реальному часі. Практична реалізація проєкту підтвердила, що саме ця комбінація інструментів є оптимальною для вирішення складних завдань архітектурної візуалізації в умовах освітнього закладу.

Ефективність Blender, у першу чергу, полягала в його повній безоплатності та відкритому програмному коді, що було критично важливим фактором для

проєкту, реалізованого на базі Національного університету «Острозька академія».



Рис. 3.7. Результати моделювання елементів

*Джерело: створено автором*

Це дозволило уникнути значних ліцензійних витрат, які є характерними для іншого професійного 3D-софту. Попри відсутність вартості, Blender виявився самодостатнім рішенням, що покривало весь цикл створення асетів: від початкового моделювання та скульптингу до детального UV-розгортання, текстурювання та фінальної оптимізації геометрії перед експортом.

У процесі моделювання архітектурних елементів староакадемічного корпусу та численних об'єктів інтер'єру інструменти Blender продемонстрували високу точність та гнучкість. Особливо ефективною виявилася система неруйнівних модифікаторів, яка дозволяла ітеративно працювати над формами, що є типовим для архітектурного проєктування. Для вирішення ключової задачі оптимізації продуктивності Blender надав потужні інструменти. Модифікатор "Decimate" активно застосовувався для алгоритмічного зменшення кількості полігонів на складних декоративних елементах, зберігаючи їх візуальну форму.

Водночас, для критично важливих ассетів застосовувалася ручна ретопологія для створення чистої, контрольованої сітки.



Рис. 3.8. Загальні вигляди аудиторій та коридорів

*Джерело: створено автором*

Стратегія оптимізації текстур також була б неможливою без ефективних інструментів UV-розгортки в Blender.

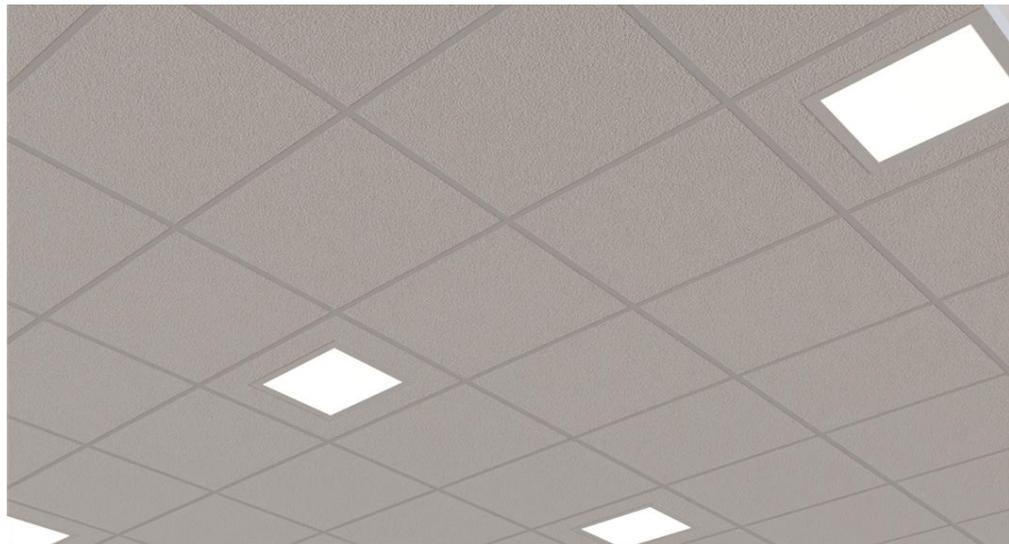


Рис. 3.9. Демонстрація результату створення матеріалу стелі нового корпусу у

Blender

*Джерело: створено автором*

Програмний пакет надав гнучкі методи, від швидкого "Smart UV Project" для простих об'єктів до повного ручного контролю через "Mark Seam" для складних моделей. Це забезпечило коректне накладання текстур та дозволило реалізувати ефективну стратегію атласування. Завершальним етапом ефективної роботи в Blender став надійний експорт у формат FBX, який забезпечив безпроблемну передачу моделей, UV-координат та матеріалів у Unreal Engine 5, мінімізуючи технічні проблеми інтеграції.

Unreal Engine 5, у свою чергу, був обраний через його революційні можливості рендерингу в реальному часі, що кардинально відрізняється від традиційних, статичних методів візуалізації. Замість багатогодинного очікування на розрахунок одного зображення (попереднього рендерингу), рушій генерує фотореалістичну картинку миттєво. Це дозволило не просто демонструвати простір, а створити повноцінний інтерактивний досвід, де користувач може вільно переміщатися, досліджувати деталі та миттєво бачити зміни, що перетворило візуалізацію на живий, динамічний інструмент.



Рис. 3.10. Результат розробки інтер'єру у Unreal Engine 5

*Джерело: створено автором*

Ключовою технологією UE5, що визначила ефективність проєкту, стала система віртуалізованої геометрії **Nanite**. Nanite дозволяє імпортувати та рендерити моделі з мільйонами, і навіть мільярдами, полігонів без традиційної необхідності у створенні рівнів деталізації (LODs). На практиці це означало, що команда могла експортувати надзвичайно деталізовані архітектурні моделі фасадів та ліпного декору з Blender і використовувати їх безпосередньо в рушії без катастрофічного падіння продуктивності. Це зекономило значну кількість часу, який зазвичай витрачається на трудомісткий процес ручної оптимізації та створення LOD-ів.

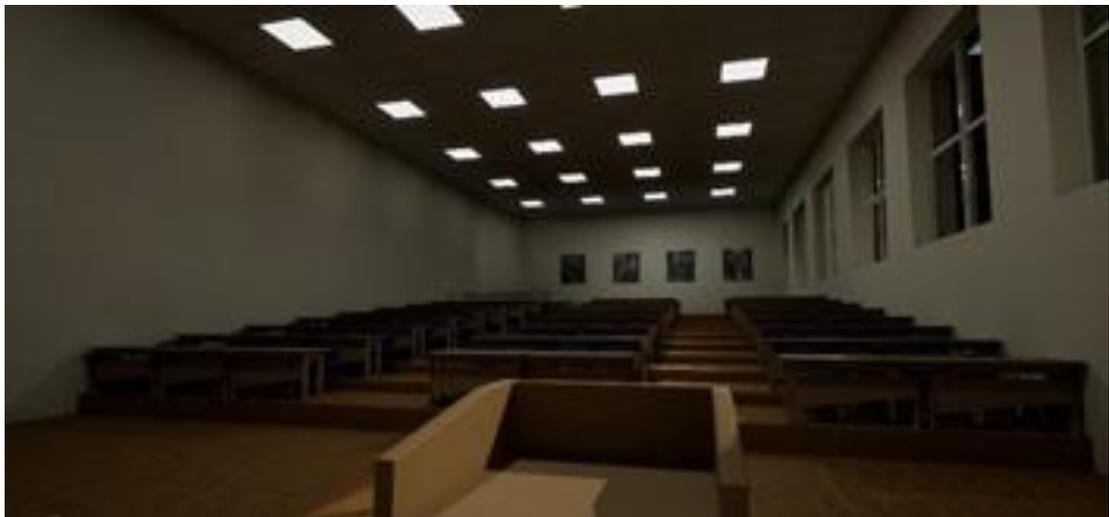


Рис. 3.11. Демонстрація роботи Nanite

*Джерело: створено автором*

Другою революційною технологією UE5 стала **Lumen** – повністю динамічна система глобального освітлення та відбиттів. Ефективність Lumen полягала у відмові від необхідності "запікання" світла (light baking), що є тривалим, статичним процесом. Lumen розраховує реалістичне непряме освітлення, м'які тіні та відбиття в реальному часі. Це дозволило команді ітеративно працювати з освітленням, налаштовуючи положення сонця (Directional Light) або інтенсивність штучних джерел (Rect Lights, Spot Lights), і миттєво бачити фінальний, реалістичний результат, що надало простору необхідної атмосфери та глибини.

Окрім візуалізації, ефективність Unreal Engine 5 проявилася у створенні інтерактивності за допомогою системи візуального скриптингу **Blueprints**.

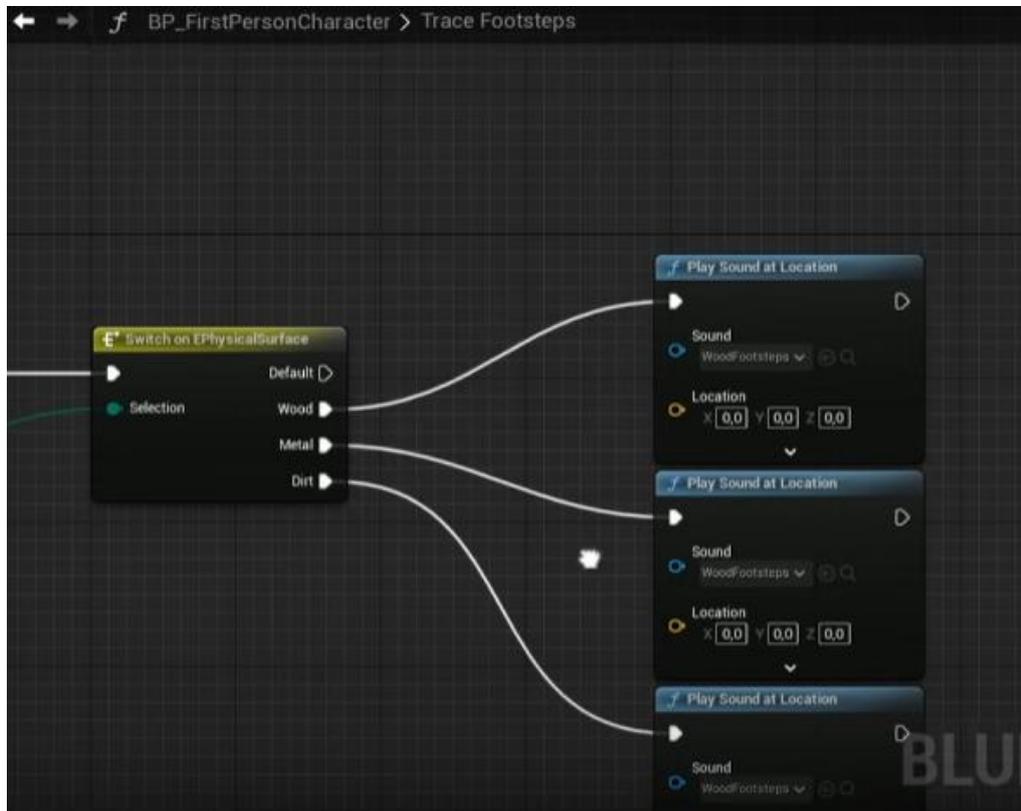


Рис. 3.12. Вигляд вікна Blueprints

*Джерело: створено автором*

Ця система, що не вимагає глибоких знань у програмуванні на C++, дозволила команді з дизайнерським фокусом самостійно реалізувати ключові функції. Зокрема, було створено інтуїтивно зрозумілу систему керування камерою (WASD та миша) для вільної навігації простором, а також реалізовано взаємодію з об'єктами, наприклад, анімацію відкриття дверей при наближенні користувача.

Синергетичний ефект від поєднання цих двох інструментів виявився надзвичайно високим. Аналіз показав, що конверс "Blender-to-UE5" є максимально ефективним. Blender виступив як ідеальний, безоплатний інструмент для підготовки оптимізованого, високоякісного контенту. Unreal

Engine 5, у свою чергу, дозволив "оживити" цей контент, використовуючи Nanite для збереження деталізації без втрати продуктивності та Lumen для миттєвого реалістичного освітлення. Гібридний підхід, де частина елементів створювалася безпосередньо в UE5, а основна маса імпортувалася, дозволив використовувати сильні сторони обох програм.

У підсумку, практична реалізація проєкту повністю підтвердила виняткову ефективність обраного стеку технологій. Blender надав необхідну функціональну базу для створення асетів без фінансових витрат. Unreal Engine 5 надав революційні технології реального часу, які дозволили обійти традиційні вузькі місця рендерингу та створити деталізований, живий інтерактивний простір. Досягнення цільових показників продуктивності, навіть на мінімальних конфігураціях обладнання, свідчить про правильність вибору та високу ефективність синергії Blender та Unreal Engine 5.

### **3.3. Оцінка досягнення цілей проєкту та рекомендації щодо вдосконалення**

Оцінка досягнення цілей проєкту демонструє, що основне завдання, визначене як «Управління проєктом створення візуальної частини інтерактивного простору Національного університету "Острозька академія"», було успішно виконано. Аналіз фактичних показників проєкту у порівнянні з плановими свідчить не лише про досягнення, але й про перевиконання ключових параметрів, що підтверджує високу ефективність обраної гібридної методології управління. Проєкт досяг своєї мети, створивши технічно оптимізований та візуально привабливий інтерактивний простір, який відтворює автентичність університету.

Однією з головних цілей було досягнення високої візуальної якості. Ця мета була повністю реалізована завдяки стратегічному вибору інструментів. Використання технології Nanite в Unreal Engine 5 дозволило імпортувати

високодеталізовані архітектурні моделі фасадів, створені в Blender, без суттєвої втрати якості чи необхідності в ручному створенні LOD-ів.



Рис. 3.13. Вигляд коридору

*Джерело: створено автором*

Система Lumen, у свою чергу, забезпечила повністю динамічне глобальне освітлення, що надало простору кінематографічного реалізму, коректно відтворюючи взаємодію світла з поверхнями в реальному часі.

Найскладнішим завданням був пошук балансу між високою візуальною якістю та оптимальною продуктивністю в реальному часі. Цільовий показник у 30 кадрів на секунду (FPS) був встановлений як мінімально прийнятний. Тестування на трьох різних конфігураціях обладнання показало виняткові результати: навіть на мінімальній конфігурації, що відповідає комп'ютерним класам університету, середня частота кадрів склала 32 FPS. На рекомендованій конфігурації показник сягнув 58 FPS, що підтверджує успішне досягнення цієї критичної мети.

Ефективність управління технічними аспектами проекту підтверджується значним перевиконанням планів оптимізації. Проектна команда змогла

зменшити загальний обсяг текстурних даних на 28.4% порівняно з планом, в основному завдяки продуманій стратегії атласування текстур, де групи дрібних об'єктів об'єднувалися в одну текстурну карту. Аналогічно, загальна кількість полігонів була зменшена на 29.2% завдяки ефективній комбінації Nanite для архітектури та методів Decimate і ручної оптимізації для інших асетів.

Проект не лише виконав, але й розширив початковий обсяг завдань. Планувалося створити 150 унікальних 3D-моделей, однак фактично було реалізовано 168, що збагатило середовище деталями. Ще більш показовим є розвиток інтерактивності: кількість Blueprint-скриптів перевищила план на 46.7%, а кількість інтерактивних об'єктів – на 24.0%. Це свідчить про високу продуктивність команди та гнучкість Scrum-підходу, який дозволив додавати нову цінність у процесі розробки.

З точки зору управління проектами, гібридна методологія була повністю валідована. Незначне відхилення від термінів (на 14.3% або два тижні) було свідомим управлінським рішенням. Цей час був інвестований у розширену R&D фазу на початку, що дозволило виявити та вирішити проблеми інтеграції Blender-UE5 до початку основного виробництва, уникнувши таким чином значно серйозніших технічних проблем і затримок на пізніх етапах.

Попри загальний успіх, аналіз проекту дозволяє сформулювати низку рекомендацій щодо вдосконалення. По-перше, досвід із двотижневою затримкою через R&D вказує на необхідність закладати ще більший часовий буфер (наприклад, 20-25%) на дослідження нових технологій у майбутніх проектах. Рекомендується також формалізувати отриманий досвід, створивши для Інформаційно-технічного центру НаУОА внутрішню базу знань та стандартизовані пресети для конвеєра Blender-UE5, щоб прискорити інтеграцію в майбутньому.

Хоча продуктивність 32 FPS на мінімальних налаштуваннях є прийнятною, вона знаходиться на межі комфортного сприйняття. Рекомендується дослідити створення окремих, більш агресивних профілів якості, які б дозволяли на низьких конфігураціях повністю вимикати найбільш витратні ефекти, як-от

динамічне освітлення Lumen. Також, хоча інструменти ШІ (Meshy AI) прискорили прототипування, вони вимагали значної ручної доробки. У майбутньому слід реалістично оцінювати час на таку постобробку, а не покладатися на них для створення фінальних ассетів.

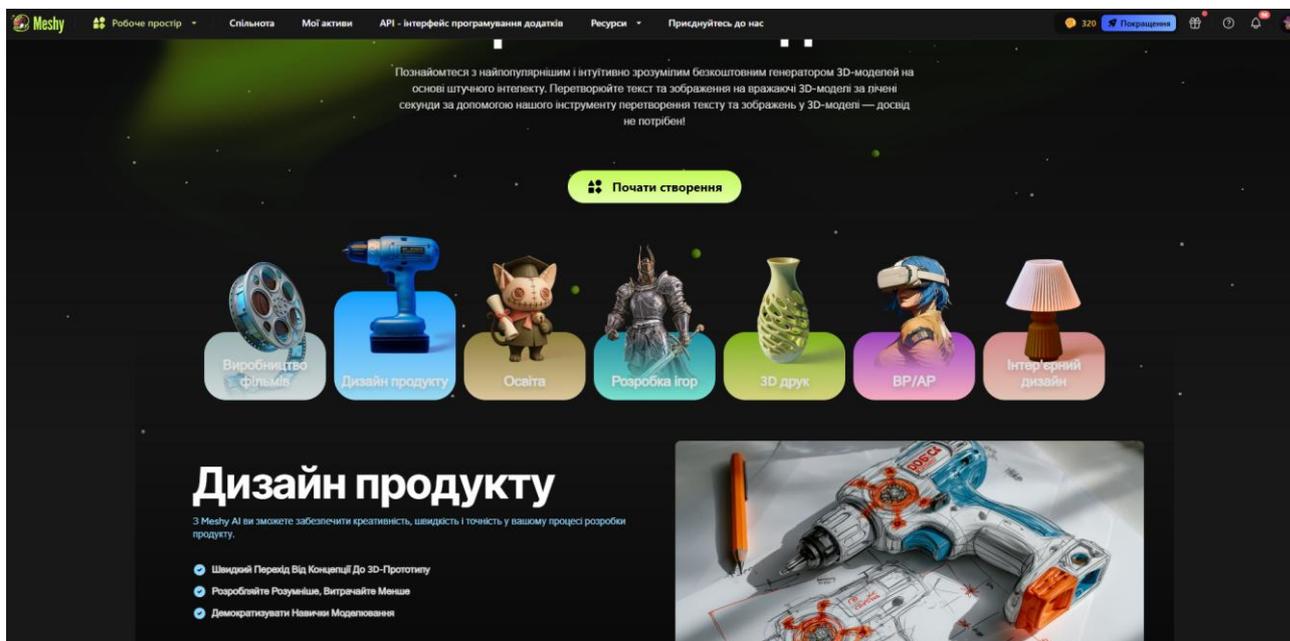


Рис. 3.14. Інтерфейс Meshy AI

*Джерело: створено автором*

Поточний проєкт успішно охопив перший та другий поверхи староакадемічного корпусу та прилеглі території. Логічною рекомендацією для подальшого розвитку є масштабування проєкту. Слід розробити дорожню карту для послідовного додавання інших корпусів, аудиторій та просторів університету. Кінцевою метою має стати створення повного, цілісного «цифрового двійника» Національного університету «Острозька академія», який можна буде використовувати для навігації, навчання, презентацій та як платформу для інших освітніх проєктів.

Наразі інтерактивність здебільшого обмежується навігацією та взаємодією з дверима. Майбутнє вдосконалення має бути зосереджене на значному поглибленні взаємодії. Рекомендується інтегрувати інформаційні точки

(hotspots), які б при натисканні відображали історичні довідки про аудиторії, експонати чи архітектурні елементи.

### **Висновки до розділу 3**

У третьому розділі було проведено комплексну експериментальну перевірку результатів проєкту та обраних підходів до його реалізації. Аналіз динаміки виконання робіт підтвердив високу ефективність гібридної методології управління, яка поєднувала каскадне планування з гнучкими практиками Scrum. Моніторинг показників спринтів засвідчив стабільне зростання продуктивності команди: від 82,4% виконаних запланованих обсягів на початковому етапі до перевищення 100% на стадіях інтеграції та оптимізації. Вирішальну роль у цьому процесі відіграли регулярні ретроспективи, які отримали найвищу оцінку ефективності серед усіх елементів методології та дозволили оперативно усувати технічні й організаційні блокери. Гнучке управління беклогом дозволило успішно імплементувати 42 із 49 запитів на зміни, збільшивши цінність продукту без критичного порушення термінів здачі проєкту.

Оцінка ефективності технічного інструментарію довела правильність вибору конвеєра «Blender-to-UE5» для вирішення завдань архітектурної візуалізації. Використання Blender забезпечило створення оптимізованого контенту без фінансових витрат, а впровадження революційних технологій Unreal Engine 5 — Nanite та Lumen — кардинально змінило виробничий процес, дозволивши використовувати високополігональні моделі без створення рівнів деталізації (LOD) та досягати реалістичного освітлення без процедури запікання світла. Застосування системи Blueprints дозволило реалізувати заплановану інтерактивність та перевиконати план за кількістю скриптів на 46,7%, не вимагаючи глибоких знань мов програмування.

Загальна оцінка проєкту підтверджує успішне досягнення всіх поставлених цілей. Фактичні показники оптимізації перевищили планові: обсяг текстурних даних було зменшено на 28,4%, а кількість полігонів — на 29,2%, що забезпечило

стабільну продуктивність на рівні 32 кадрів за секунду навіть на мінімальних апаратних конфігураціях. На основі отриманих результатів було сформульовано рекомендації для подальшого розвитку проєкту, які включають масштабування розробки до рівня повноцінного «цифрового двійника» університету, поглиблення інформаційної інтерактивності через систему hotspots та необхідність закладання більшого часового буфера на етап R&D у майбутніх проєктах для мінімізації технічних ризиків.

## ВИСНОВКИ

У ході виконання кваліфікаційної роботи було успішно вирішено поставлене завдання – здійснено управління проектом створення візуальної частини інтерактивного простору Національного університету «Острозька академія». Аналіз результатів підтверджує високу ефективність обраної гібридної методології управління, яка поєднувала елементи каскадної моделі для загального планування та гнучкі практики Scrum для безпосереднього виконання робіт. Такий підхід дозволив команді ефективно адаптуватися до неминучих змін, швидко реагувати на технічні виклики, характерні для креативно-технічних проєктів, та успішно керувати очікуваннями стейкхолдерів.

Стратегічний вибір технологічного стеку, що базувався на синергії безкоштовного та потужного 3D-редактора Blender та інноваційного ігрового рушія Unreal Engine 5, повністю себе виправдав. Blender зарекомендував себе як самодостатній інструмент для повного циклу створення та оптимізації 3D-асетів, що було критично важливим в умовах освітнього закладу. Використання Unreal Engine 5, зокрема його революційних технологій Nanite та Lumen, дозволило досягти фундаментального прориву у якості візуалізації. Технологія Nanite дозволила працювати з високодеталізованими архітектурними моделями без необхідності в ручному створенні LOD-ів, а Lumen забезпечив повністю динамічне глобальне освітлення кінематографічного рівня, усунувши необхідність у тривалому процесі «запікання» світла.

Проєкт досяг і перевиконав ключові цілі: було створено візуально привабливий та автентичний простір, що відтворює архітектуру університету. Критичне завдання балансування між якістю та продуктивністю було успішно вирішене, що підтверджується досягненням стабільних 32 FPS навіть на мінімальних конфігураціях обладнання, що відповідають комп'ютерним класам НаУОА. Команда перевиконала початковий план за кількістю створених 3D-моделей та рівнем інтерактивності, що свідчить про високу продуктивність та гнучкість Scrum-підходу. Ефективність управління технічними аспектами

підтверджується значним покращенням планових показників оптимізації, зокрема зменшенням обсягу текстурних даних на 28.4% та загальної кількості полігонів на 29.2%.

Незначне планове відхилення від початкових термінів (на 14.3%) було свідомим управлінським рішенням інвестувати додатковий час у розширену R&D фазу, що дозволило уникнути значних технічних проблем на пізніх етапах.

Отже, гіпотеза дослідження була повністю підтверджена: обрана гібридна методологія та конвеєр «Blender-to-UE5» є високоефективним рішенням для реалізації складних проєктів інтерактивної візуалізації. Робота має практичне значення, оскільки створений продукт може слугувати основою для «цифрового двійника» університету, а отриманий досвід та розроблені стандарти можуть бути формалізовані для подальшого масштабування проєкту та розвитку інноваційних освітніх інструментів в НаУОА.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Hughes D. The Traditional Waterfall Approach [Електронний ресурс] / D. Hughes // University of Missouri–St. Louis. – Режим доступу: <https://www.umsl.edu/~hugheyd/is6840/waterfall.html> (дата звернення: 06.11.2025).
2. Agile Methodology: A Complete Guide to Agile Project Management [Електронний ресурс] // Code Brew. – Режим доступу: <https://www.codebrew.com/agile-methodology/> (дата звернення: 06.11.2025).
3. What Is Scrum Methodology? An Introduction For Agile Teams [Електронний ресурс] // NimbleWork. – 2025. – Режим доступу: <https://www.nimblework.com/agile/scrum-methodology/> (дата звернення: 06.11.2025).
4. What Is Kanban? An Overview Of The Kanban Method [Електронний ресурс] // NimbleWork. – 2024. – Режим доступу: <https://www.nimblework.com/kanban/what-is-kanban/> (дата звернення: 06.11.2025).
5. Verstraete S. Discussing the Possibilities and Drawbacks of Unreal Engine 5's Nanite [Електронний ресурс] / S. Verstraete // 80 Level. – Режим доступу: <https://80.lv/articles/discussing-the-possibilities-and-drawbacks-of-unreal-engine-5-s-nanite> (дата звернення: 06.11.2025).
6. The Ultimate R&D Plan Timeline: From Idea To Product [Електронний ресурс] // ITGov-Docs. – Режим доступу: <https://www.itgov-docs.com/blogs/it-governance/research-and-development-plan> (дата звернення: 06.11.2025).
7. Підбираємо ідеальну робочу станцію для рендера в Corona Renderer [Електронний ресурс] // Hyperditek. – Режим доступу: <https://hyperditek.com.ua/ua/blog/podbiraem-idealnuyu-rabochuyu-stantsiyu-dlya-rendera-v-corona-renderer> (дата звернення: 06.11.2025).
8. Wright D. Unreal Engine 5 goes all-in on dynamic global illumination with Lumen [Електронний ресурс] / D. Wright, K. Narkowicz // Unreal Engine. – 2022. – Режим доступу: <https://www.unrealengine.com/en-US/tech-blog/unreal->

[engine-5-goes-all-in-on-dynamic-global-illumination-with-lumen](#) (дата звернення: 06.11.2025).

9. Microsoft HoloLens [Електронний ресурс] // Wikipedia. – Режим доступу: [https://en.wikipedia.org/wiki/Microsoft\\_HoloLens](https://en.wikipedia.org/wiki/Microsoft_HoloLens) (дата звернення: 06.11.2025).

10. Вийшов інструмент для Houdini, який перетворює локації з гугл-мап на 3D-меші [Електронний ресурс] // DOU. – Режим доступу: <https://gamedev.dou.ua/news/earth-mesh-houdini-asset/> (дата звернення: 06.11.2025).

11. Simple Scatter Plugin [Електронний ресурс] // Unreal Engine Forums. – Режим доступу: <https://forums.unrealengine.com/t/simple-scatter-plugin/137578/3> (дата звернення: 06.11.2025).

12. Meshy AI: How to Use It, Is It Good & Better AI Option [Електронний ресурс] // Dreamina CapCut. – Режим доступу: <https://dreamina.capcut.com/resource/meshy-ai> (дата звернення: 06.11.2025).

13. Кравчук О. Метод використання метрик продуктивності для оптимізації процесу управління ІТ-проєктами. 2023. URL: <https://elar.khmnmu.edu.ua/server/api/core/bitstreams/ee253e63-38d1-4b17-979b-78983c5c8fbc/content> (дата звернення: 09.11.2025).

14. Лук'янова К. Ю. Методи оцінювання ефективності програмних проєктів та оптимізації процесів розробки. 2019. URL: <https://openarchive.nure.ua/server/api/core/bitstreams/d77983fa-cc78-42f8-b5ac-e8139df39572/content> (дата звернення: 09.11.2025).

15. Крамський С. Методи оптимізації науково-технічної діяльності з наукових проєктів приватного закладу вищої освіти. Управління розвитком складних систем. 2021. № 45. С. 35–42. URL: <http://mdcs.knuba.edu.ua/article/view/237966/236647> (дата звернення: 09.11.2025).

16. Яцина В., Кочетова Т. Синергетичний підхід до оптимізації інвестиційних бізнес-процесів. Вісник Національного технічного університету «Харківський політехнічний інститут» (економічні науки). 2020. № 3. С. 50–53.

URL: <http://es.khpi.edu.ua/article/view/222280/222440> (дата звернення: 09.11.2025).

17. Блага Н. Управління проектами. 2021. URL: <https://surl.lt/domago> (дата звернення: 09.11.2025).

18. Поліщук П., Волощук Б., Бардін Я. Проектування та демонстрація споруд з використанням Unreal Engine. 2024. URL: <https://repository.knuba.edu.ua/server/api/core/bitstreams/d9189410-4f16-4fc0-9ae7-54012f1bd94b/content> (дата звернення: 09.11.2025).

19. Безвін Б. В. Методи реалізації інтерактивної системи побудови будівель у відкритому світі на базі Unreal Engine 5 : кваліфікаційна робота. Суми : Сумський державний університет, 2024. URL: <https://essuir.sumdu.edu.ua/server/api/core/bitstreams/9839ad8b-d527-414e-9e7f-1253c78c2397/content> (дата звернення: 09.11.2025).

20. Ригін Ю. В. Створення графічних моделей на основі рушію Unreal Engine. 2024. URL: <https://surl.li/odoobl> (дата звернення: 09.11.2025).

21. Корнюков Н. О. Ігровий застосунок в жанрі Survival Horror Stalker Shadow Of Pripyat на основі рушія Unreal Engine 5. 2024. URL: <https://surl.li/idmehe> (дата звернення: 09.11.2025).

22. Трусов Б. О. Інформаційна технологія візуалізації елементів доповненої реальності в системах дизайну інтер'єрів : кваліфікаційна робота. Суми : Сумський державний університет, 2022. URL: <https://essuir.sumdu.edu.ua/server/api/core/bitstreams/03b15b98-3ac9-4a99-a1a1-ae842cd943df/content> (дата звернення: 09.11.2025).

23. Лучной С. В. Gameready 3D-модель для комп'ютерної гри та технологія її розробки. 2024. URL: <https://openarchive.nure.ua/server/api/core/bitstreams/30d9596b-e06a-4bc9-ba48-f7fb115147b8/content> (дата звернення: 09.11.2025).

24. Гур'янова Л. С., Євсєєв О. С., Сімакова М. Р. Використання генеративного штучного інтелекту в менеджменті проектів розробки 3D-

моделей ігрового оточення. 2024. URL: <https://surl.li/rdxttp> (дата звернення: 09.11.2025).

25. Войтович Д. І. Програмні засоби для створення візуальних 3D ефектів мультимедійного контенту. 2021. URL: <https://ela.kpi.ua/server/api/core/bitstreams/4ff8939a-8ef1-460b-8b0b-d53c5560d988/content> (дата звернення: 09.11.2025).

26. Антонов А. І. Візуалізація 3D моделі військового призначення. 2018. URL: <https://essuir.sumdu.edu.ua/server/api/core/bitstreams/9c711314-336c-4a59-b709-7c62b9772ca5/content> (дата звернення: 09.11.2025).

27. Біденко А. Ю. 3D моделювання з використанням штучного інтелекту. 2024. URL: <https://ela.kpi.ua/server/api/core/bitstreams/c432e8e0-933d-4d8a-a15c-f56dace56cf2/content> (дата звернення: 09.11.2025).

28. Данченко О. Б., Бедрій Д. І., Семко І. Б. Управління конфліктами наукового проекту. Вісник Національного технічного університету «ХПІ». Серія: Стратегічне управління, управління портфелями, програмами та проектами. 2019. № 2 (1327). С. 28–35. URL: <https://er.chdtu.edu.ua/bitstream/ChSTU/2804/1/9682-3683-PB.pdf> (дата звернення: 09.11.2025).

29. Ковшун Н. Е., Левун О. І. Аналіз та реалізація проектів. 2022. URL: <https://surl.li/dgcvsww> (дата звернення: 09.11.2025).

30. Дарчук В. Г. Аналіз та планування маркетингової діяльності : навч. посіб. Київ : ДУТ, 2019. 146 с. URL: [https://duikt.edu.ua/uploads/1\\_796\\_67558907.pdf](https://duikt.edu.ua/uploads/1_796_67558907.pdf) (дата звернення: 09.11.2025).

31. Зачко О. Б., Кобилкін Д. С., Головатий Р. Р. Моделі управління безпекою інфраструктурних проектів на стадії планування. 2019. URL: <https://sci.ldubgd.edu.ua/jspui/bitstream/123456789/7929/2/163793-358044-1-PB.pdf> (дата звернення: 09.11.2025).

32. Гончарик Р. П. Потреби сучасних інструментів архітектурної візуалізації в специфікації матеріалів. 2021. URL: <https://surl.li/zxamqt> (дата звернення: 09.11.2025).

33. Kulik M. et al. Методи нейромережевого впливу на формальне трансформаційне вираження архітектурних стилів. Bulletin of Lviv National Environmental University. Series Architecture and Construction. 2023. № 24. С. 113–119. URL: <https://visnyk.lnup.edu.ua/index.php/architecture/article/view/208/207> (дата звернення: 09.11.2025).

34. Ріднокодубська Г. Використання інтерактивних інноваційних технологій під час викладання навчальної дисципліни «Управління проектами в освіті». 2021. URL: <https://dspace.wunu.edu.ua/bitstream/316497/42307/1/3.pdf> (дата звернення: 09.11.2025).

35. Шашкова Н. І., Фадєєва І. Г., Казакова Т. С. Управління проектами в ІТ сфері: застосування гнучких методологій. Scientific Notes of Lviv University of Business and Law. 2021. № 28. С. 166–172. URL: <https://nzlubp.org.ua/index.php/journal/article/view/402/379> (дата звернення: 09.11.2025).

36. Абрамчук Д. В. Інтеграція технологій доповненої реальності і віртуальної реальності в управлінні проектами : кваліфікаційна робота. Тернопіль : ЗУНУ, 2023. URL: <https://surl.li/npcboh> (дата звернення: 09.11.2025).

37. Мокієнко Ю., Морозова Г. Сучасні моделі і методи управління проектами у закладах вищої освіти. Управління розвитком складних систем. 2023. № 56. С. 105–115. URL: <http://mdcs.knuba.edu.ua/article/view/299765/292296> (дата звернення: 09.11.2025).

38. Сметанюк О. А., Бондарчук А. В. Особливості системи управління проектами в ІТ-компаніях. Агросвіт. 2021. № 10. С. 105–111. URL: <https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/31942/80462.pdf?sequence=2&isAllowed=y> (дата звернення: 09.11.2025).

39. Зачко О. Б. Управління проектами: теорія, практика, інформаційні технології. 2019. URL: <https://surl.li/vmldzi> (дата звернення: 09.11.2025).

40. Крамський С., Євдокімова О., Захарченко О. Економіко-математичні методи управління науковими проектами у навчальних закладах вищої освіти. Ринкова економіка: сучасна теорія і практика управління. 2021.

## ДОДАТКИ

## ДОДАТОК А

Лістинг 1. Код створення матеріалу скла

```
// GlassManipulator.cpp (або інша назва вашого класу)

#include "GlassManipulator.h"
#include "UObject/ConstructorHelpers.h"

// Sets default values
AGlassManipulator::AGlassManipulator()
{
    PrimaryActorTick.bCanEverTick = true;

    // Створюємо компонент мешу
    GlassMesh =
CreateDefaultSubobject<UStaticMeshComponent>(TEXT("GlassMesh"));
    RootComponent = GlassMesh;

    // Завантажуємо базовий матеріал з Content Browser за шляхом
    static ConstructorHelpers::FObjectFinder<UMaterialInterface>
MaterialFinder(TEXT("Material'/Game/M_Glass_Full.M_Glass_Full"));
    if (MaterialFinder.Succeeded())
    {
        BaseGlassMaterial = MaterialFinder.Object;
    }
}

// Called when the game starts or when spawned
void AGlassManipulator::BeginPlay()
{
```

```

Super::BeginPlay();

// Створюємо динамічний екземпляр матеріалу під час запуску гри
if (BaseGlassMaterial)
{
    DynamicGlassInstance =
UMaterialInstanceDynamic::Create(BaseGlassMaterial, this);
    if (DynamicGlassInstance)
    {
        // Призначаємо динамічний матеріал мешу
        GlassMesh->SetMaterial(0, DynamicGlassInstance);

        // Встановлюємо початкові параметри за замовчуванням
        ApplyGlassParameters(FLinearColor(0.8f, 0.9f, 1.0f, 1.0f), 0.05f, 1.0f);
    }
}

// Called every frame
void AGlassManipulator::Tick(float DeltaTime)
{
    Super::Tick(DeltaTime);

    // Тут ви можете динамічно змінювати параметри, наприклад,
    // якщо скло має ставати менш прозорим з часом або змінювати колір.
}

void AGlassManipulator::ApplyGlassParameters(FLinearColor TintColor, float
NewRoughness, float NewOpacity)
{

```

```
if (DynamicGlassInstance)
{
    // Встановлюємо параметри за назвами, які ми визначили в редакторі
    DynamicGlassInstance->SetVectorParameterValue(FName("BaseColor"),
TintColor);
    DynamicGlassInstance->SetScalarParameterValue(FName("Roughness"),
NewRoughness);
    DynamicGlassInstance->SetScalarParameterValue(FName("Opacity"),
NewOpacity);
    // Рефракцію та металевість можна залишити сталими або також зробити
параметрами
}
}
```

