

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

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня бакалавра

на тему: **«Розробка 3D контенту та інтерактивного інтерфейсу для мобільного ігрового додатку Libtake з використанням сучасних інструментів моделювання, текстуровання та інтеграції»**

Виконав: студент 4 курсу, групи КН-41
першого (бакалаврського) рівня вищої освіти
спеціальності 122 Комп'ютерні науки
освітньо-професійної програми «Комп'ютерні науки»
Дейнека Валерій Володимирович

Керівник: викладач кафедри інформаційних технологій та
аналітики даних
Місай Володимир Віталійович

Рецензент: Розробник програмного забезпечення в ТОВ
ОБНОВА-ЄВРОШОП, викладач кафедри інформаційних
технологій та аналітики даних НаУОА
Шведюк Володимир Володимирович

РОБОТА ДОПУЩЕНА ДО ЗАХИСТУ

Завідувач кафедри інформаційних технологій та аналітики даних _____
(проф., д.е.н. Кривицька О.Р.)
Протокол №10 від 28 травня 2025 р.

Острог, 2025

АНОТАЦІЯ
кваліфікаційної роботи
на здобуття освітнього ступеня бакалавра

Тема: Розробка 3D контенту та інтерактивного інтерфейсу для мобільного ігрового додатку Libtake з використанням сучасних інструментів моделювання, текстурювання та інтеграції

Автор: Дейнека Валерій Володимирович

Науковий керівник: викладач Місай Володимир Віталійович

Захищена «.....»..... 20__ року.

Пояснювальна записка до кваліфікаційної роботи: 60 с., 31 рис., 0 табл., 0 додатків, 17 джерел.

Короткий зміст праці:

У кваліфікаційній роботі розглянуто етапи створення 3D-контенту та інтерактивного інтерфейсу для мобільного ігрового додатку Libtake. Визначено основні завдання: розробка стилістики гри, створення 3D-моделей та анімацій з урахуванням обмежень мобільних платформ, розробка UX/UI-дизайну, інтеграція контенту в ігрове середовище Unity та оптимізація продуктивності. Проведено аналіз сучасних програмних засобів для моделювання, текстурювання та створення інтерфейсів. Встановлено роль технічного художника у процесі розробки та тестування. Результатом роботи стало функціональне ігрове середовище з оптимізованим 3D-контентом та зручним інтерфейсом.

Ключові слова: 3D-КОНТЕНТ, ІГРОВИЙ ІНТЕРФЕЙС, МОБІЛЬНИЙ ДОДАТОК, UNITY, UX/UI-ДИЗАЙН, ТЕХНІЧНИЙ ХУДОЖНИК, МОДЕЛЮВАННЯ, ТЕКСТУРУВАННЯ, ОПТИМІЗАЦІЯ, АНІМАЦІЯ

Summary of the work:

This qualification thesis takes into account the process of developing 3D content and an interactive interface for the mobile game application Libtake. Main tasks include the design of the style of the game, creating 3D models and animations with consideration for mobile platform limitations, developing UX/UI design, implementing assets in the Unity engine, and optimization of performance. The paper covers modern tools for modeling, texturing, and designing the interface. The role of a technical artist in the development and testing process is defined. As a result, a functional game environment with optimized 3D content and user-friendly interface was created.

Keywords: 3D CONTENT, GAME INTERFACE, MOBILE APPLICATION, UNITY, UX/UI DESIGN, TECHNICAL ARTIST, MODELING, TEXTURING, OPTIMIZATION, ANIMATION

(підпис автора)

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	4
ВСТУП.....	6
РОЗДІЛ 1. Теоретичні аспекти створення 3D-контенту та інтерфейсів для мобільних ігрових додатків.....	8
1.1. Актуальність і перспективи розвитку мобільної ігрової індустрії.....	8
1.2. Особливості створення 3D-контенту для мобільних платформ.....	10
1.3. Аналіз та порівняння сучасних інструментів для розробки 3D-графіки та інтерактивного інтерфейсу.....	11
1.4. Роль технічного художника в процесі розробки ігор.....	14
1.5. UX/UI-дизайн у створенні інтерактивних інтерфейсів.....	17
1.6. Постановка задачі та вимоги до графічного контенту.....	19
Висновок до розділу 1.....	21
РОЗДІЛ 2. Розробка 3D-контенту та ігрового інтерфейсу, інтеграція в ігрове середовище.....	22
2.1. Проектування стилістики гри: персонажі, оточення, текстури.....	22
2.2. Створення 3D-моделей з урахуванням мобільних платформ.....	24
2.3. Розробка анімацій для взаємодії гравця із об'єктами гри.....	31
2.4. Створення UX/UI-дизайну з використанням сучасних інструментів.....	35
2.5. Інтеграція контенту в Unity.....	36
2.6. Освітлення та його оптимізація.....	38
Висновок до розділу 2.....	41
РОЗДІЛ 3. Оновлення ігрового додатку, оцінка результатів та перспективи розвитку проекту.....	42
4.1. Створення оновлення з новою глобальною ціллю для ігрового додатку.....	42
4.2. Доповнення контентом на основі зворотного зв'язку.....	48
4.3. Оптимізація продуктивності.....	51
Висновок до розділу 3.....	55
ВИСНОВКИ.....	57

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

Анімація	Створення руху об'єктів та персонажів
ASTC	Adaptive Scalable Texture Compression (адаптивне масштабоване стиснення текстур)
Canvas	Спеціальний об'єкт в Unity, який відповідає за відображення графічного інтерфейсу користувача (UI)
Compression	Стиснення даних для зменшення розміру файлів
Edge Detection	Метод виявлення контурів об'єктів зображення або сцени
ETC2	Ericsson Texture Compression 2 (технологія стиснення текстур)
FPS	Frames Per Second (кадри на секунду)
GPU	Graphics Processing Unit (графічний процесор)
Inverted Hull	Метод створення контурів шляхом вивертання геометрії моделі
Light Probes	Точки в просторі, що зберігають дані освітлення для динамічних об'єктів
Lightmap	Текстура, що зберігає інформацію про освітлення сцени
Lowpoly	Модель з малою кількістю полігонів
Mesh	Геометрична сітка 3D-моделі
Motion capture	Захоплення руху для анімації персонажів
Normals (нормалі)	Вектор, перпендикулярний поверхні, що використовується для освітлення
Outline	Контур об'єкта візуально
Particle Effects	Ефекти з частинок (вогонь, дим, дощ тощо)
Pipeline (Пайплайн)	Послідовність етапів обробки графіки у рушії або візуалізаційному процесі
Prefab	Попередньо налаштований об'єкт у Unity
Рендеринг	Процес виведення зображення на екран шляхом обчислення освітлення, тіней, текстур та матеріалів сцени

Ретаргетинг анімації	Перенесення анімації з одного скелета або персонажа на інший, з урахуванням відмінностей їх структур
Rigging	Створення скелетної структури для анімації моделі
Screen Space	Простір, що відповідає екрану пристрою
Shader	Програма для візуалізації поверхонь у графічному рушії
Texel Density	Щільність текстурних пікселів на одиницю площі моделі
Текстурування	Процес нанесення текстур на 3D-об'єкти
UI	User Interface (інтерфейс користувача)
UV	Двовимірна координатна система для розгортки текстур
UV Mapping	Розгортка 3D-моделі для застосування текстури
UV-Island	Окрема замкнена ділянка на UV-розгортці, не з'єднана з іншими
UX	User Experience (досвід користувача)

ВСТУП

У сучасному світі мобільні ігрові додатки відіграють важливу роль у сфері цифрових розваг і навчання, демонструючи стрімке зростання популярності та впливу. За останні роки ринок мобільних ігор перевищив половину світового ігрового сегменту, що свідчить про високий попит і актуальність створення якісного контенту для мобільних платформ. Особливої уваги заслуговують ігри з елементами едьютейнменту, які поєднують освітню цінність і розважальні механіки, що робить їх перспективним напрямом розробки.

Актуальність теми дослідження зумовлена потребою у створенні оптимізованого 3D-контенту для мобільних ігор, який забезпечує візуальну привабливість, ефективну взаємодію користувача з ігровим середовищем, а також стабільну продуктивність на широкому спектрі пристроїв. У рамках дослідження розглядається процес створення графічного контенту для мобільного ігрового додатку Libtake, що поєднає освітні та розважальні компоненти в стилізованому 3D-середовищі віртуальної бібліотеки.

Метою дослідження є розробка оптимізованого тривимірного контенту та інтерактивного інтерфейсу для мобільного додатку Libtake із використанням сучасних інструментів моделювання, текстурування, анімації та інтеграції в ігрове середовище Unity.

Для досягнення мети були визначені такі завдання дослідження:

1. аналіз предметної області мобільної розробки з акцентом на створення 3D-графіки;
2. проектування структури інформаційної системи мобільного додатку;
3. вибір інструментальних засобів для реалізації, оптимізації та тестування тривимірного ігрового контенту.

Об'єктом дослідження є мобільний ігровий додаток Libtake як приклад інформаційної системи з ігровою механікою та освітнім контентом.

Предмет дослідження – інструментальні засоби розробки та оптимізації 3D-графіки для мобільних платформ, включаючи математичні моделі, методи оптимізації, інформаційні технології візуалізації та ігрові рушії.

Методами дослідження є:

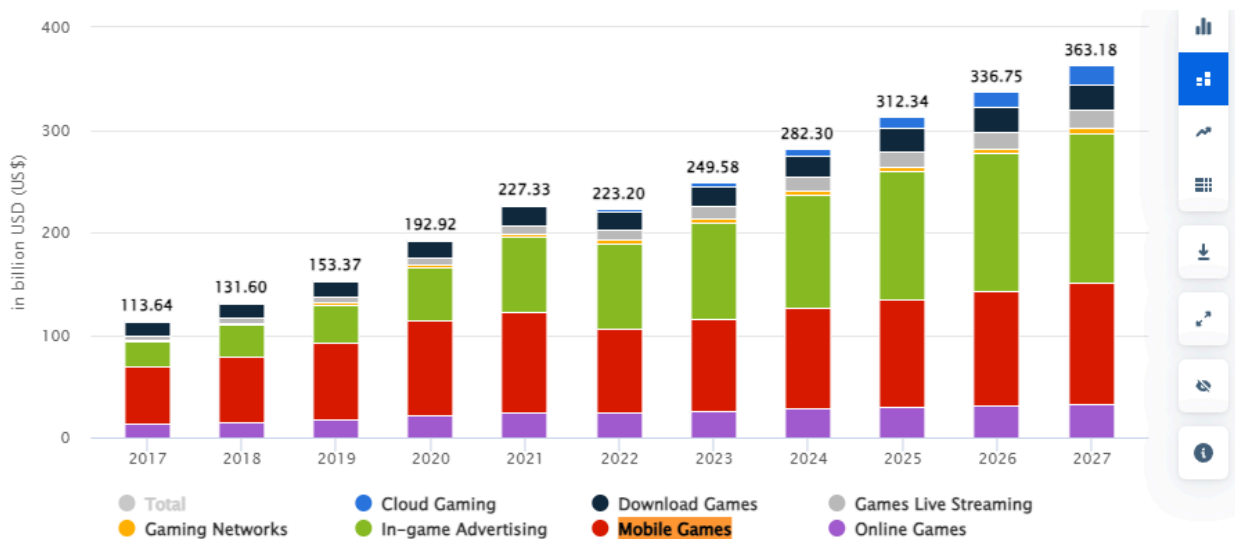
1. аналіз і порівняння сучасних графічних інструментів (Blender, GIMP, Unity, Figma);
2. моделювання та розробка 3D-об'єктів із застосуванням принципів оптимізації;
3. створення та покращення інтерактивного інтерфейсу для мобільного додатку;

Усі ці етапи дозволяють створити ефективний графічний контент для освітньо-розважального застосунку, здатного працювати на широкому спектрі мобільних платформ з урахуванням технічних обмежень.

РОЗДІЛ 1. Теоретичні аспекти створення 3D-контенту та інтерфейсів для мобільних ігрових додатків

1.1. Актуальність і перспективи розвитку мобільної ігрової індустрії

Мобільна ігрова індустрія є однією з найбільш динамічних і швидкозростаючих сфер цифрових розваг. За останні десятиліття мобільні ігри перетворилися з простих додатків на складні інтерактивні проекти, які можуть конкурувати за якістю з продуктами для ПК і консолей. Актуальність розвитку цього напрямку зумовлена стрімким поширенням мобільних пристроїв, постійним вдосконаленням їхньої апаратної складової та доступністю інтернету, що створює нові можливості для розробників і гравців.



Notes: Data was converted from local currencies using average exchange rates of the respective year.

Most recent update: Mar 2024

Source: Statista Market Insights

Рис 1.1. Графік обсягу ігрового ринку

Джерело: www.statista.com

Популярність мобільних ігор спричинена їхньою доступністю та зручністю. Смартфони стали невід'ємною частиною життя мільярдів людей, а їх використання охоплює всі вікові групи та соціальні категорії. Гравці мають змогу насолоджуватися іграми будь-де: вдома, у транспорті, на перерві під час роботи чи навчання. Крім

того, мобільні ігри пропонують широкий спектр жанрів – від головоломок до масштабних багатокористувацьких онлайн-ігор, що задовольняє різноманітні вподобання користувачів.

Економічна складова мобільних ігор також відіграє важливу роль у їхньому розвитку. За даними аналітичних агентств, мобільні ігри вже перевершують інші сегменти ігрової індустрії за прибутковістю. Це пояснюється поширенням моделі free-to-play, яка дозволяє гравцям безкоштовно встановлювати ігри з можливістю робити внутрішньоігрові покупки. Монетизація через рекламу, підписки та мікротранзакції стимулює розробників до створення якісного ігрового контенту.

Значний вплив на розвиток індустрії мають також соціальні мережі та стримінгові платформи. Мобільні ігри стають частиною популярної культури, а гравці діляться своїми досягненнями в соціальних мережах, обговорюють ігри у спільнотах і навіть транслюють свій ігровий процес у реальному часі. Це сприяє залученню нових користувачів та розширенню аудиторії.

Однак, разом із можливостями, мобільна ігрова індустрія стикається з викликами. Основні з них включають високу конкуренцію серед розробників, потребу у врахуванні різноманітних технічних характеристик пристроїв, питання захисту даних користувачів і етичні аспекти монетизації. Для того щоб залишатися конкурентоспроможними, розробникам потрібно не лише створювати привабливі ігри, але й забезпечувати високу продуктивність, зручність користування та естетичну привабливість.

Таким чином, мобільна ігрова індустрія має значний потенціал для подальшого розвитку завдяки постійному прогресу технологій, широкій аудиторії користувачів і економічній вигідності. Водночас успіх у цій галузі вимагає інноваційного підходу, уваги до деталей і здатності адаптуватися до швидкозмінних умов ринку.

1.2. Особливості створення 3D-контенту для мобільних платформ

Створення 3D-контенту для мобільних платформ є одним із найбільш складних, але водночас цікавих аспектів розробки мобільних ігор. Особливості цього процесу тісно пов'язані з технічними обмеженнями мобільних пристроїв, специфікою їхньої апаратної складової, а також потребою у забезпеченні оптимального балансу між візуальною якістю та продуктивністю.

На відміну від потужних ігрових ПК або консолей, мобільні пристрої мають обмежені обчислювальні ресурси, які накладають певні рамки на створення тривимірної графіки. З одного боку, сучасні смартфони оснащені високопродуктивними процесорами та графічними чіпами, які здатні обробляти складну графіку. З іншого боку, ці пристрої обмежені в енергоспоживанні, що не дозволяє використовувати всі ресурси на повну потужність без шкоди для автономності. Тому 3D-моделі, текстури, анімація й інші елементи контенту повинні бути ретельно оптимізовані.

Оптимізація 3D-моделей є ключовим етапом у створенні графічного контенту для мобільних платформ. Вона полягає у зменшенні кількості полігонів, що утворюють модель, без втрати візуальної якості. Цей процес потребує вміння знаходити баланс між деталізацією об'єкта і продуктивністю гри, адже чим більше полігонів містить сцена, тим більше ресурсів необхідно для її обробки.

Текстури для мобільних ігор також вимагають особливої уваги. Через обмеження обсягу пам'яті текстури зазвичай зберігаються у стислому вигляді та мають невеликий розмір. Часто використовуються методи, які дозволяють зберегти якість текстур, але зменшити їхню вагу, наприклад, розділення на канали чи використання текстурних атласів. Крім того, у мобільних іграх активно застосовуються процедурні текстури, які створюються в реальному часі, що дозволяє значно зменшити навантаження на пам'ять пристрою.

Анімації є ще одним важливим елементом, який впливає на продуктивність мобільної гри. Для оптимізації використовуються спрощені скелетні анімації, мінімізація кількості кісток та їхньої складності. Для ще більшої оптимізації використовують процедурні анімації, які написані кодом.

Освітлення і шейдери також відіграють важливу роль у створенні якісного візуального контенту. На мобільних платформах часто використовуються попередньо розраховані карти освітлення (lightmaps) замість реального часу, оскільки це значно знижує навантаження на графічний процесор. Шейдера для мобільних ігор зазвичай мають спрощену структуру, щоб забезпечити швидкість їх виконання.

Ще однією важливою особливістю є врахування різноманітності пристроїв, що працюють на різних платформах. Розробникам необхідно адаптувати контент до широкого спектра екранів із різними роздільними здатностями, співвідношеннями сторін і потужностями апаратного забезпечення. Це вимагає використання універсальних підходів до дизайну, таких як масштабування моделей та текстур, а також оптимізація під різні версії операційних систем.

Таким чином, створення 3D-контенту для мобільних платформ – це складний процес, який поєднує технічні знання, креативний підхід і вміння працювати з сучасними інструментами розробки. Водночас він відкриває широкі можливості для створення унікальних ігрових продуктів, які можуть залучати гравців у всьому світі завдяки своїй доступності та високій якості.

1.3. Аналіз та порівняння сучасних інструментів для розробки 3D-графіки та інтерактивного інтерфейсу

Blender є одним із найпопулярніших інструментів для 3D-моделювання завдяки відкритому вихідному коду, безкоштовній ліцензії та великій спільноті користувачів, яка активно розвиває програму. Він пропонує широкий спектр функцій: створення моделей, текстурування, анімація, симуляція фізики (динаміка

рідин, твердих тіл, тканин тощо), а також рендеринг за допомогою потужних рушіїв, таких як Cycles та Eevee. Blender підтримує розширення за допомогою плагінів, що дозволяє адаптувати його під специфічні потреби проекту.

Порівняно з конкурентами, такими як Autodesk Maya, Blender має кілька важливих переваг. Maya є галузевим стандартом у кінематографі та розробці AAA-ігор, пропонуючи передові інструменти для високополігонального моделювання, ригінгу та анімації. Однак її вартість і складність освоєння роблять Maya менш доступною для інді-розробників. Blender, натомість, пропонує всі базові та навіть частину професійних функцій Maya, при цьому залишається безкоштовним, що робить його ідеальним вибором для мобільних проектів.

Іншим важливим конкурентом є ZBrush, який спеціалізується на цифровому скульптингу та створенні високодеталізованих моделей. ZBrush є незамінним інструментом для створення персонажів та високоякісних активів, проте для мобільних ігор, де деталізація часто обмежена продуктивністю пристроїв, функціонал ZBrush є надлишковим. Blender забезпечує базові можливості для скульптингу, а також потужні інструменти для оптимізації моделей (зменшення полігональності, UV-розгортка), що є критично важливим для мобільних платформ.

Також варто згадати Cinema 4D, який відомий своєю простотою використання та інтеграцією з програмами для обробки відео, такими як Adobe After Effects. Проте Cinema 4D більше орієнтований на створення анімацій та візуальних ефектів для реклами та кінематографу, що обмежує його застосування у розробці ігрових моделей.

Unity став одним із найпоширеніших ігрових рушіїв у світі завдяки своїй універсальності, простоті освоєння та підтримці великої кількості платформ, включаючи мобільні пристрої, ПК, ігрові консолі та навіть VR/AR. Unity пропонує низький поріг входу, що дозволяє навіть новачкам швидко освоїти основи розробки, одночасно забезпечуючи розширені можливості для досвідчених розробників.

Однією з ключових переваг Unity є його модульна структура. Рушій дозволяє використовувати готові бібліотеки, плагіни та інструменти, що значно прискорює процес розробки. Наприклад, Asset Store пропонує безліч безкоштовних і платних ресурсів, від 3D-моделей до шейдерів та анімацій. Unity також підтримує кілька рендеринг-пайплайнів, серед яких Universal Render Pipeline (URP), розроблений спеціально для оптимізації графіки на мобільних пристроях.

У порівнянні з Unreal Engine, Unity поступається у графічній потужності. Unreal Engine пропонує передові технології, такі як Lumen (глобальне освітлення) та Nanite (віртуалізація полігонів), що дозволяє створювати фотореалістичну графіку. Проте ці функції часто є зайвими для мобільних проектів через обмеження ресурсів пристроїв. Unity фокусується на оптимізації та продуктивності, що робить його ідеальним для мобільної розробки.

Ще одним конкурентом є Godot Engine, який активно розвивається як відкритий ігровий рушій. Godot пропонує інтуїтивний інтерфейс, особливо зручний для 2D-ігор, а також базові можливості для 3D. Проте у порівнянні з Unity, функціонал Godot у сфері 3D-графіки обмежений, і його спільнота значно менша.

GIMP є безкоштовним редактором растрової графіки, який забезпечує широкий набір інструментів для обробки зображень, включаючи підтримку шарів, масок, фільтрів, та інтеграцію з іншими програмами через плагіни. GIMP використовується для створення та редагування текстур, що є важливим етапом у розробці будь-якої гри.

Основним конкурентом GIMP є Adobe Photoshop, який є галузевим стандартом у професійній обробці графіки. Photoshop пропонує значно ширший набір функцій, інтуїтивний інтерфейс і тісну інтеграцію з іншими продуктами Adobe, такими як Illustrator та After Effects. Проте вартість підписки на Photoshop робить його менш доступним для невеликих команд і інді-розробників, тоді як GIMP забезпечує необхідний функціонал без додаткових витрат.

Ще один цікавий конкурент – Krita, орієнтований на цифровий живопис. Krita має перевагу у створенні концепт-артів та ілюстрацій, проте для редагування текстур та растрової графіки її можливості обмежені. Paint.NET, інший популярний інструмент, є простим у використанні, але поступається GIMP за функціональністю.

Figma – це сучасний інструмент для UI/UX-дизайну, який відрізняється своєю зручністю для командної роботи. Вона працює в браузері, що дозволяє легко співпрацювати над проектами в реальному часі. Figma ідеально підходить для створення інтерфейсів, прототипів та макетів мобільних ігор.

Порівняно з Adobe XD, Figma має перевагу завдяки простоті доступу з будь-якого пристрою без необхідності встановлення. Adobe XD, зі свого боку, пропонує глибоку інтеграцію з іншими продуктами Adobe, що може бути зручним для професійних дизайнерів.

Інший конкурент, Sketch, був популярним серед дизайнерів до появи Figma, але його обмеження платформною macOS і відсутність браузерної версії значно звужують можливості співпраці. Figma, завдяки своїй універсальності, стала одним із найкращих інструментів для дизайну інтерфейсів у багатоплатформених командах.

Blender, Unity, GIMP і Figma – це інструменти, які виділяються завдяки своїй доступності, універсальності та відповідності вимогам мобільної розробки. У порівнянні з конкурентами, ці платформи забезпечують оптимальний баланс між функціоналом, простотою використання та вартістю, що робить їх чудовим вибором для інді-розробників і малих студій. Однак їхній успіх значною мірою залежить від потреб проекту, навичок команди та наявних ресурсів.

1.4. Роль технічного художника в процесі розробки ігор

Роль технічного художника в процесі розробки ігор є ключовою та надзвичайно багатогранною. Цей фахівець виступає своєрідним мостом між творчою і технічною командами, забезпечуючи оптимальне поєднання художнього

задуму й ефективності роботи гри на різних платформах, зокрема на мобільних. У сучасній ігровій індустрії, де вимоги до візуальної якості та продуктивності постійно зростають, технічний художник стає незамінним учасником проекту.

Основна відповідальність технічного художника полягає у впровадженні технологій, які дозволяють досягти найвищої візуальної якості при оптимальному використанні ресурсів. Він займається створенням і оптимізацією графічних активів, таких як моделі, текстури, шейдери та ефекти, щоб вони відповідали вимогам мобільних платформ. Наприклад, технічний художник може працювати над зменшенням кількості полігонів у моделях без значної втрати якості або над стискуванням текстур для зменшення обсягу пам'яті, необхідної для гри.

Ще однією важливою задачею технічного художника є створення шейдерів – програм, які визначають, як саме буде відображатися світло і текстури на об'єктах у грі. Для мобільних платформ шейдери повинні бути максимально простими й ефективними, оскільки складні обчислення можуть суттєво впливати на продуктивність пристрою. Технічний художник займається оптимізацією існуючих шейдерів або розробляє власні, які відповідають потребам проекту.

Крім роботи з графікою, технічний художник бере участь у налаштуванні систем освітлення. Освітлення в іграх відіграє важливу роль у створенні атмосфери та реалістичності сцени, але водночас є одним із найскладніших елементів для обчислення. Технічний художник може впроваджувати техніки попередньо розрахованого освітлення (lightmaps) або застосовувати простіші моделі реального часу для забезпечення гарного вигляду при мінімальному навантаженні на процесор.

Однією з ключових сфер відповідальності технічного художника є забезпечення стабільної роботи гри на всіх підтримуваних пристроях. Оскільки мобільні платформи відрізняються великою кількістю моделей пристроїв із різними характеристиками, технічний художник працює над адаптацією контенту для кожного з них. Це включає тестування гри на різних пристроях, аналіз

продуктивності та виправлення можливих проблем, таких як низький FPS або артефакти графіки. Він також може створювати інструменти автоматизації цього процесу, що дозволяє економити час і зусилля команди.

Технічний художник часто працює у тісній співпраці з програмістами, щоб інтегрувати графічний контент у ігровий рушій, а також із дизайнерами, щоб забезпечити відповідність художньої реалізації їхньому задуму. Наприклад, якщо дизайнери створюють концепт складного об'єкта з великою кількістю деталей, технічний художник допомагає адаптувати цей концепт для реального використання в грі, розробляючи ефективні рішення, які зберігають основну ідею, але знижують навантаження на систему.

Крім основних обов'язків, технічний художник займається дослідженнями і впровадженням нових технологій у проект. Це може включати використання процедурного моделювання, генерацію текстур або анімацію з використанням штучного інтелекту. Завдяки цьому команда отримує інструменти, які не лише покращують якість кінцевого продукту, але й спрощують робочі процеси.

Важливим аспектом роботи технічного художника є його здатність комунікувати з різними відділами команди, пояснюючи складні технічні концепції простими словами. Він також виступає в ролі наставника для інших членів команди, навчаючи їх ефективним методам створення контенту, який відповідає технічним стандартам. Наприклад, художникам і дизайнерам можуть бути надані рекомендації щодо оптимального формату текстур або складності моделей.

У сучасному середовищі роль технічного художника стає дедалі важливішою, особливо в умовах, коли ігри повинні працювати на широкому спектрі пристроїв із різними характеристиками. Завдяки своїм знанням і досвіду він забезпечує баланс між творчістю і технічними обмеженнями, що дозволяє створювати красиві, але водночас продуктивні ігри. Його робота не лише впливає на візуальну якість

продукту, але й гарантує, що гравці отримають позитивний досвід незалежно від того, на якому пристрої вони грають.

1.5. UX/UI-дизайн у створенні інтерактивних інтерфейсів

UX/UI-дизайн відіграє одну з ключових ролей у створенні інтерактивних інтерфейсів, особливо у мобільних ігрових додатках. Цей напрямок зосереджений на розробці зручних, естетично привабливих і функціональних інтерфейсів, які забезпечують позитивний досвід для користувачів. Успішний UX/UI-дизайн не лише впливає на візуальне сприйняття продукту, а й формує загальне враження гравців від гри, що робить його надзвичайно важливим у процесі розробки.

UX-дизайн стосується загального досвіду взаємодії користувача з грою. Основною метою UX-дизайнера є створення логічної, зрозумілої і захопливої взаємодії між гравцем і додатком. У контексті мобільних ігор UX-дизайн включає побудову структури меню, навігаційних елементів, кнопок та ігрових механік, які не викликають плутанини у користувачів. Наприклад, перехід між екранами має бути інтуїтивно зрозумілим і швидким, а інформація – доступною без зайвих зусиль з боку гравця. У цьому процесі важливо враховувати особливості мобільних пристроїв, такі як розмір екрана, сенсорне керування та обмеження обчислювальних ресурсів.

UI-дизайн відповідає за візуальну складову взаємодії, включаючи естетичний вигляд кнопок, текстів, меню, індикаторів та інших елементів інтерфейсу. UI-дизайнер працює над тим, щоб кожен візуальний елемент гармонійно вписався в загальний стиль гри, відповідав її атмосфері та жанру. Наприклад, у казуальних іграх UI-дизайн може бути яскравим і простим, тоді як у хардкорних RPG важливо акцентувати увагу на деталізованих елементах, що підкреслюють складність ігрового світу.

Важливим завданням UX/UI-дизайну є забезпечення балансу між естетикою та функціональністю. Інтерфейс повинен бути візуально привабливим, але не переобтяженим декоративними елементами, які можуть ускладнити взаємодію. Наприклад, якщо в мобільній грі гравець не може легко знайти кнопку для початку рівня або меню налаштувань, це створює негативний досвід і знижує загальну якість продукту. Тому UX/UI-дизайнери працюють над тим, щоб усі елементи були розташовані логічно і відповідали очікуванням користувачів.

У мобільних іграх особливу увагу слід приділяти адаптивності інтерфейсу, адже пристрої різняться за розміром екрана, роздільною здатністю та пропорціями. UX/UI-дизайнер має враховувати ці параметри, щоб інтерфейс виглядав добре як на маленьких смартфонах, так і на великих планшетах. Це передбачає створення адаптивних макетів, які автоматично підлаштовуються під розмір екрана, а також тестування дизайну на різних пристроях для виявлення можливих проблем.

Ще одним важливим аспектом UX/UI-дизайну є інтерактивність і відгук на дії користувача. У мобільних іграх гравець очікує негайного реагування на свої дії, наприклад, при натисканні кнопки чи виконанні жесту. Інтерактивні елементи, такі як анімації натискань або спливаючі підказки, створюють відчуття живого зв'язку між гравцем і грою, що позитивно впливає на враження від продукту.

Колірна палітра, шрифти та стиль іконок відіграють важливу роль у сприйнятті гри. UX/UI-дизайнери працюють над створенням емоційного зв'язку між візуальними елементами та користувачами, обираючи кольори, які відповідають атмосфері гри, та шрифти, які легко читати навіть на невеликих екранах. Наприклад, у грі про магію можуть використовуватися м'які кольори, що нагадують місячне світло, а в грі-екшені перевага надається насиченим відтінкам, які підкреслюють динаміку.

Процес UX/UI-дизайну включає етапи дослідження, прототипування, тестування та вдосконалення. На початковому етапі дизайнери аналізують цільову

аудиторію гри, її звички та очікування. Наприклад, якщо гра орієнтована на молодь, інтерфейс може бути яскравішим і сучаснішим, тоді як для старшої аудиторії важливі простота та доступність. Далі створюються прототипи, які дозволяють візуалізувати ідеї та протестувати їх на реальних користувачах. Результати тестування допомагають визначити слабкі місця дизайну, які можна вдосконалити.

Таким чином, UX/UI-дизайн є критично важливим компонентом у розробці мобільних ігор. Він визначає, наскільки комфортно та приємно гравцям взаємодіяти з продуктом, забезпечує адаптивність і функціональність інтерфейсу, створює візуальну ідентичність гри та впливає на успішність її комерційної моделі. Інтеграція добре продуманого UX/UI-дизайну на всіх етапах розробки є запорукою створення якісного, конкурентоспроможного продукту, який знайде відгук у серцях користувачів.

1.6. Постановка задачі та вимоги до графічного контенту

Вимоги до графічного контенту та інтерфейсів мобільного додатку на прикладі Libtake визначаються комплексом факторів, які впливають на візуальну привабливість, функціональність і загальний користувацький досвід. У розробці мобільних додатків особливе значення має те, як графічний контент взаємодіє з інтерфейсом і наскільки органічно вони доповнюють один одного. У випадку Libtake, основний акцент робиться на забезпеченні естетики, адаптивності, ефективності використання ресурсів та інтуїтивності.

Графічний контент, який включає моделі, текстури, анімації та візуальні ефекти, має бути оптимізованим для мобільних пристроїв. Це означає, що ресурси повинні бути якомога менш вимогливими до пам'яті та процесора, зберігаючи при цьому високу якість зображення. Наприклад, текстури в Libtake стискаються без втрати ключових деталей, що дозволяє зменшити їх розмір, а моделі створюються з використанням оптимальної кількості полігонів. Таке рішення гарантує, що додаток

буде працювати плавно навіть на пристроях середнього та низького класу, не поступаючись у візуальній якості.

Кольорова палітра і стиль графіки мають відповідати концепції та цільовій аудиторії Libtake. Якщо додаток орієнтований на молодь, перевага надається яскравим, динамічним кольорам, які викликають асоціацію з енергією та сучасністю. У той же час, більш нейтральні та пастельні відтінки можуть бути обрані для того, щоб створити спокійний, дружній інтерфейс для ширшої аудиторії. Важливим є також дотримання єдиного стилю, що допомагає створити впізнаваність бренду та забезпечити гармонійність візуальної складової.

Особлива увага приділяється інтерфейсам, адже вони є головним засобом взаємодії користувача з додатком. Інтерфейс Libtake має бути зрозумілим і зручним для всіх категорій користувачів, незалежно від їхнього досвіду. Це досягається за рахунок інтуїтивно зрозумілих іконок, зручного розташування кнопок, а також адаптивного дизайну, який забезпечує комфортне використання програми на пристроях із різними розмірами екранів.

Під час розробки графічного контенту і інтерфейсів для Libtake проводяться численні тести. Вони дозволяють перевірити зручність використання, швидкість завантаження та адаптивність інтерфейсу. Зворотний зв'язок від тестувальників та користувачів враховується для внесення корективів у фінальний продукт.

Загалом, вимоги до графічного контенту та інтерфейсів у додатку Libtake спрямовані на забезпечення естетики, функціональності та позитивного користувацького досвіду. Грамотний підхід до цих аспектів дозволяє створити продукт, який є не лише технічно якісним, але й привабливим для користувачів, що сприяє його популярності та успіху на ринку мобільних додатків.



Рис. 1.2. UML-діаграма діяльності

Джерело: Розроблено автором

Висновок до розділу 1

У даному розділі розглянуто теоретичні основи створення 3D-контенту та інтерфейсів для мобільних ігор. Встановлено, що розвиток мобільної ігрової індустрії зумовлює потребу в якісній, але оптимізованій графіці. Особливу увагу приділено використанню сучасних інструментів, таких як Blender, Unity, Figma, що дозволяють ефективно створювати й адаптувати 3D-контент для мобільних платформ.

Також проаналізовано роль технічного художника, який забезпечує баланс між візуальною якістю та технічною ефективністю. Таким чином, для успішної розробки мобільних ігор необхідне поєднання художніх навичок, технічних знань і вміння працювати з відповідними програмами.

РОЗДІЛ 2. Розробка 3D-контенту та ігрового інтерфейсу, інтеграція в ігрове середовище

2.1. Проектування стилістики гри: персонажі, оточення, текстури

Проектування стилістики гри є ключовим аспектом створення візуального образу мобільної гри LibTake, що впливає на загальне сприйняття гравцем ігрового світу. Процес проектування охоплював розробку персонажів, оточення та текстур, які гармонійно поєднувалися між собою, відповідаючи загальній концепції гри. Усі елементи було реалізовано в стилістиці LowPoly, що забезпечувало баланс між естетичною привабливістю та технічною оптимізацією.

Оточення гри створювалося як єдиний візуальний простір, у якому прості форми, насичені кольори й чіткі контури поєднувалися для створення атмосферного ігрового світу. Кожен об'єкт у середовищі був розроблений із мінімальною кількістю полігонів, що забезпечувало ефективне використання ресурсів мобільних пристроїв. Моделі проходили оптимізацію, спрямовану на зниження кількості вертексів і усунення невидимих поверхонь. Візуальна виразність досягалася без втрати продуктивності, що особливо важливо для мобільного геймінгу.

Ефект Outline є важливим елементом візуального дизайну, що дозволяє чітко виділити об'єкти, з якими гравець може взаємодіяти, акцентувати увагу на ключових елементах та підвищити зрозумілість ігрового процесу. Для створення такого ефекту існує кілька підходів, серед яких найбільш поширеними є Screen Space Outline, Screen Space Edge Detection та Inverted Hull Outline. Кожен із цих методів має свої переваги та недоліки, які враховуються при виборі оптимального рішення для конкретного проекту.

Метод Screen Space Outline реалізується шляхом аналізу зображення у просторі екрана. Після рендерингу сцени формується текстура глибини або нормалей, яка використовується для виявлення контурів об'єктів. Основна ідея цього

підходу полягає у виявленні різких змін значень глибини або напрямку нормалей, що відповідають краям об'єктів. У цих місцях накладається контур, який виділяє модель. Цей метод є ефективним і не вимагає додаткових геометричних ресурсів, що робить його зручним для використання у складних сценах. Однак його продуктивність може знижуватися при високій роздільній здатності екрану, а якість контуру залежить від деталізації текстур нормалей і глибини.

Інший метод – Screen Space Edge Detection – також працює у просторі екрана, але аналізує вже відрендерене зображення для виявлення країв об'єктів за допомогою змін кольору між сусідніми пікселями. Цей підхід вимагає менше обчислень, ніж робота з текстурами нормалей або глибини, що спрощує його реалізацію. Він є особливо ефективним у сценах із високим контрастом кольорів між об'єктами та фоном. Проте у сценах із однорідною палітрою або дрібними деталями цей метод може створювати помилкові контури, що впливає на точність відображення ефекту.

Метод Inverted Hull Outline працює з геометрією об'єктів і є одним із найефективніших для створення чітких і контрольованих контурів. Він передбачає створення додаткового меша, який повторює форму основної моделі, але зміщується назовні відносно нормалей поверхні. Цей додатковий меш обробляється спеціальним шейдером, який дозволяє створити видимий контур. Такий підхід забезпечує якісний і стабільний ефект незалежно від складності сцени. Основною перевагою цього методу є можливість точно контролювати вигляд контуру, що робить його ідеальним для стилізованих ігор із простими моделями. Однак він вимагає додаткових обчислень для створення й рендерингу мешів, що може стати проблемою для об'єктів із великою кількістю полігонів.

У контексті розробки мобільної гри LibTake для реалізації ефекту Outline було обрано метод Inverted Hull Outline. Він забезпечуватиме оптимальне співвідношення між якістю візуалізації та ефективністю використання ресурсів. Завдяки створенню

додаткового меша навколо об'єкта та використанню шейдера для накладання контуру буде досягнуто високої чіткості ефекту без значного навантаження на систему. Це рішення стало ключовим для покращення ігрового процесу, дозволяючи гравцям легко орієнтуватися у світі гри та виділяти інтерактивні об'єкти.

Текстури, як важлива частина стилістики, також були створені з урахуванням LowPoly підходу. UV-розгортка має виконуватися таким чином, щоб уникати зайвих швів і забезпечувати простоту нанесення текстур. Основою для текстур стане мінімалістична палітра кольорів, що створювало стильний вигляд об'єктів. Відмова від дрібних деталей і високої роздільної здатності текстур дозволила зменшити розмір файлів, що було важливим для забезпечення плавної роботи гри на мобільних пристроях.

Таким чином, процес проектування стилістики гри для LibTake поєднав у собі естетичну привабливість із технічною ефективністю. Усі елементи, від персонажів до текстур, будуть створені таким чином, щоб зберігати єдність стилю й відповідати вимогам оптимізації. Ця підхід забезпечує унікальну візуальну ідентичність гри та її комфортну роботу на мобільних пристроях.

2.2. Створення 3D-моделей з урахуванням мобільних платформ

Процес створення 3D-моделей для мобільної гри LibTake включав кілька ретельно продуманих етапів, кожен із яких був спрямований на досягнення високого рівня якості графіки та забезпечення максимальної оптимізації для мобільних пристроїв. Завдання полягало у створенні моделей, які відповідали б вимогам сучасних мобільних ігор і могли безпроблемно працювати навіть на пристроях із обмеженими ресурсами. Уся розробка базувалася на концепціях LowPoly стилю, який дозволяє зберігати чіткість і естетичність моделей, мінімізуючи їх деталізацію.

Першим і найважливішим етапом розробки було використання референсів, які визначали основний напрямок створення кожної моделі. Референси служили

джерелом натхнення та базою для формування пропорцій, масштабу й загальної стилістики об'єктів. Використання таких референсів дозволяло уникати помилок у формах і пропорціях, що могли б негативно вплинути на загальне враження від візуального ряду гри.



Рис 2.1. Референси наукової бібліотеки «Острозька академія»

Джерело: Google Maps

Наступним важливим етапом у виробництві LibTake стало створення 3D-моделей у Blender. Основний акцент було зроблено на оптимізацію продуктивності для мобільних пристроїв, а саме – мінімізацію кількості вертексів у моделях. Саме вертекси, а не трикутники, мають найвідчутніший вплив на швидкість рендерингу. Чим менше вершин – тим менше навантаження на графічний процесор, що критично для пристроїв з обмеженими технічними можливостями.

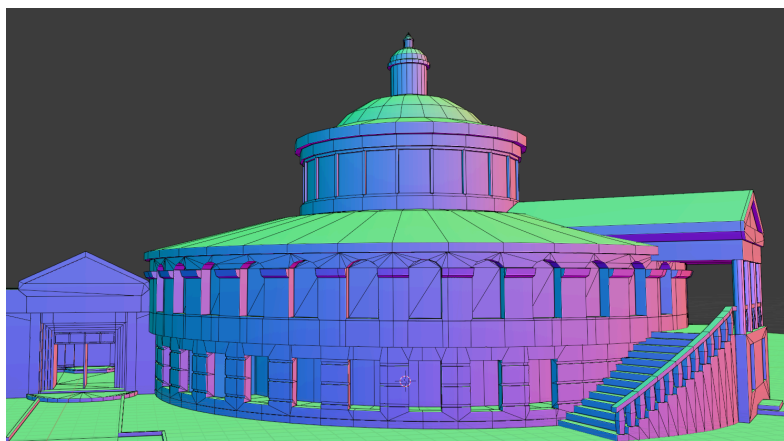


Рис 2.2. Прев'ю змодельованої будівлі бібліотеки

Джерело: Розроблено автором

Принципи моделювання:

1. Моделі мають містити мінімальну кількість вертексів, навіть якщо при цьому збільшується кількість трикутників.
2. Для всіх моделей використовується єдиний матеріал із назвою "UberMaterial", що ґрунтується на загальній палітрі гри.
3. Назви об'єктів у сцені повинні повністю збігатися з назвами відповідних мешів – це спрощує управління та експорт.

Щодо згладжування, Auto-Smooth застосовується лише у виняткових випадках. LowPoly-естетика гри передбачає виразні, чіткі грані, тому згладжування поверхонь не тільки не є необхідним, а й може зруйнувати бажаний візуальний ефект. До того ж, відмова від Auto-Smooth сприяє додатковому зниженню навантаження на GPU.

Для підтримання порядку у файлах і забезпечення простоти інтеграції моделей у Unity була впроваджена система уніфікації назв. Всі моделі отримували однакові назви з відповідними мешами, що полегшувало ідентифікацію об'єктів у проекті та знижувало ризик помилок під час їх імпорту чи експорту. Такий підхід також сприяв спрощенню управління матеріалами й текстурами, що було важливим для забезпечення структурованості великого ігрового проекту.

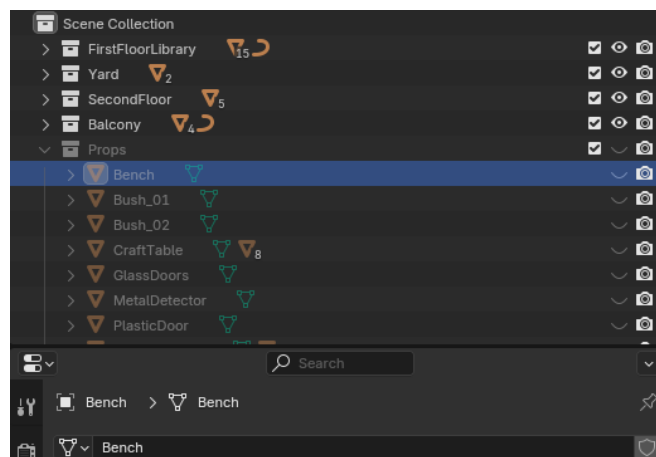


Рис 2.3. Структура проекту з моделями в Blender

Джерело: Розроблено автором

Під час розробки LibTake прийнято рішення про мінімізацію кількості текстур, щоб зменшити загальний обсяг гри та оптимізувати її для мобільних пристроїв. Частина матеріалів була отримана з відкритих джерел, зокрема з ресурсу ambientcg.com, після чого вони проходили адаптацію у редакторі растрової графіки GIMP. Наприклад, текстури бруківки та килима спочатку мали надто високу деталізацію та яскраву кольорову палітру, що контрастувало зі стриманим стилем LowPoly.

У GIMP ці текстури було ретельно оброблено – зменшено деталізацію, приглушено кольори й забезпечено безшовність. Це дозволило їм органічно вписатися в візуальну концепцію гри. Серед використаних інструментів:

1. Noise Reduction – для зменшення надлишкового шуму,
2. Shadows-Highlights – для корекції контрасту,
3. Desaturate – для пом'якшення надто яскравих відтінків,
4. А також художні фільтри, такі як Cartoon, Waterpixels, Posterize та Oilify, що допомогли стилізувати текстури під піксельно-ретро формат гри.

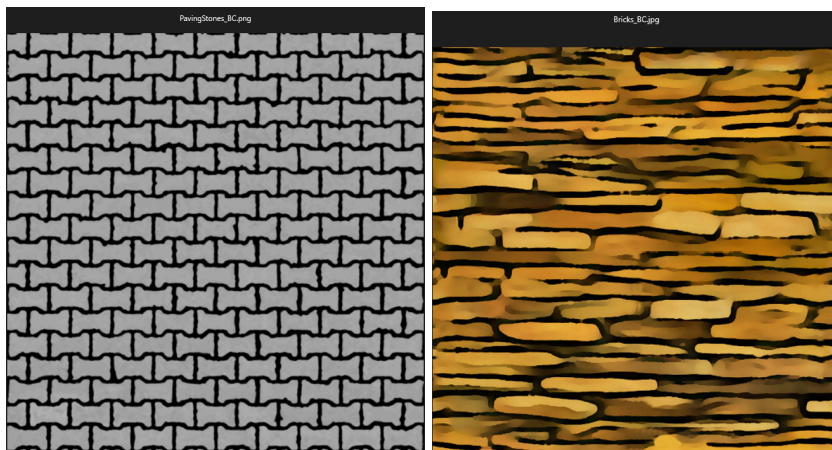


Рис 2.4. Приклади текстур зроблених методом модифікації існуючих

Джерело: Розроблено автором

На додаток до адаптації сторонніх текстур у LibTake було створено і власні унікальні графічні елементи. Однією з ключових стала спеціально розроблена кольорова палітра, яка використовується для всіх матеріалів у гри. Вона виконує роль

універсального кольорового шаблону, що дозволяє стилістично об'єднати всі об'єкти та значно скоротити кількість окремих текстур.

Палітра має надзвичайно компактний розмір – усього 32x32 пікселі, завдяки чому вона практично не навантажує систему. Це стратегічне рішення забезпечило ідеальний баланс між візуальною цілісністю гри та її оптимізацією для мобільних платформ. Використання єдиної кольорової бази дозволяє підтримувати стабільну продуктивність без втрати художньої якості.



Рис 2.5. Текстура палітри кольорів розміром 32x32 пікселі на якій відбувався UV-Mapping

Джерело: Розроблено автором

Текстура головного героя є ключовим елементом, спеціально створеним для гри LibTake. Вона була задумана з урахуванням обмежень мобільних пристроїв – розробники свідомо обрали низьку роздільну здатність, аби зменшити навантаження на систему. Водночас текстура зберегла виразний зовнішній вигляд та індивідуальність персонажа. Виконана у піксель-арт стилі, вона додає образу ностальгійної ретро-атмосфери, яка гармонійно поєднується з LowPoly-моделями в грі. Таке поєднання не лише підтримує загальну естетику проекту, а й створює візуальну цілісність без шкоди для якості.

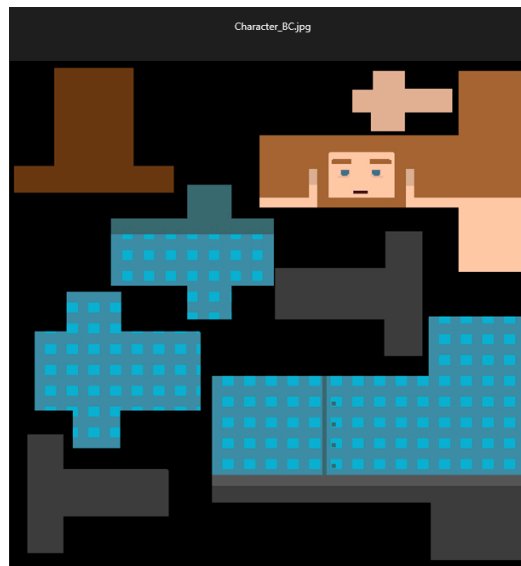


Рис 2.6. Текстура головного героя

Джерело: Розроблено автором

Одним із найбільш трудомістких, але водночас ключових етапів процесу стало створення UV-розгорток для моделей. UV-Mapping відіграє важливу роль у забезпеченні високої якості текстуровання, адже від його точності залежить, наскільки природно текстури виглядатимуть на поверхні моделей. У LowPoly стилі UV-розгортки розроблялися таким чином, щоб мінімізувати використання текстурних ресурсів і водночас зберегти їхню високу якість. Для моделей, які використовували палітру кольорів, UV-острови розташовувалися відповідно до пікселів текстури, що дозволяло зменшити розмір текстурних файлів без втрати їхньої деталізації. У тих випадках, коли застосовувалися безшовні текстури, UV-розгортка враховувала texel density, забезпечуючи рівномірну щільність текстури на всіх частинах моделі.

Особливої уваги вимагала підготовка UV-розгорток для карт освітлення (lightmaps), які відповідали за якісне відображення статичних тіней у грі. Хоча Unity має вбудовану функцію автоматичного створення таких UV, її робота не завжди задовольняла вимоги до якості, тому UV-розгортки створювалися вручну. Усі статичні об'єкти отримували додаткову UV, яка була оптимізована для роботи з

картами освітлення. Це дозволяло уникати перехрещення запечених пікселів і зменшувати кількість артефактів на освітлених поверхнях. Розмір текстур карт світла становив 64х64 пікселів, а відступи між UV-островами варіювалися від 2.5 до 4 пікселів, що забезпечувало їх коректне розташування й уникнення світлових перекриттів.

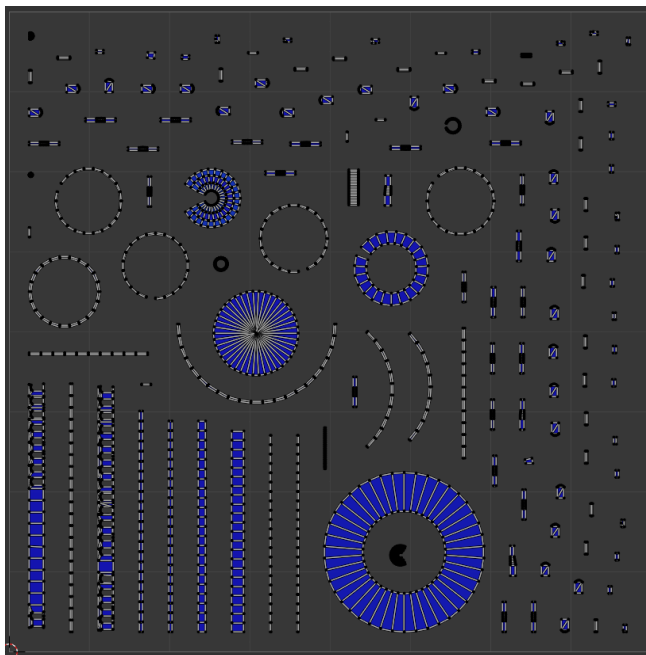


Рис 2.7. Приклад UV розгортки одного із об'єктів розгортки для карт світла

Джерело: Розроблено автором

Для забезпечення зручності геймплею було впроваджено ефект Outline, який підкреслював контури моделей і допомагав гравцю орієнтуватися у сцені. Серед кількох доступних методів реалізації Outline був обраний Inverted Hull Outline, оскільки він забезпечував баланс між якістю ефекту та оптимізацією. Цей метод передбачав створення додаткового меша, який повторював форму основної моделі, але був зміщений назовні відносно її нормалей. Контурний шейдер у Unity відповідав за рендеринг цього меша, забезпечуючи чітке й стабільне відображення контуру. Такий підхід дозволив уникнути значного навантаження на систему, що було критично важливо для мобільних платформ.

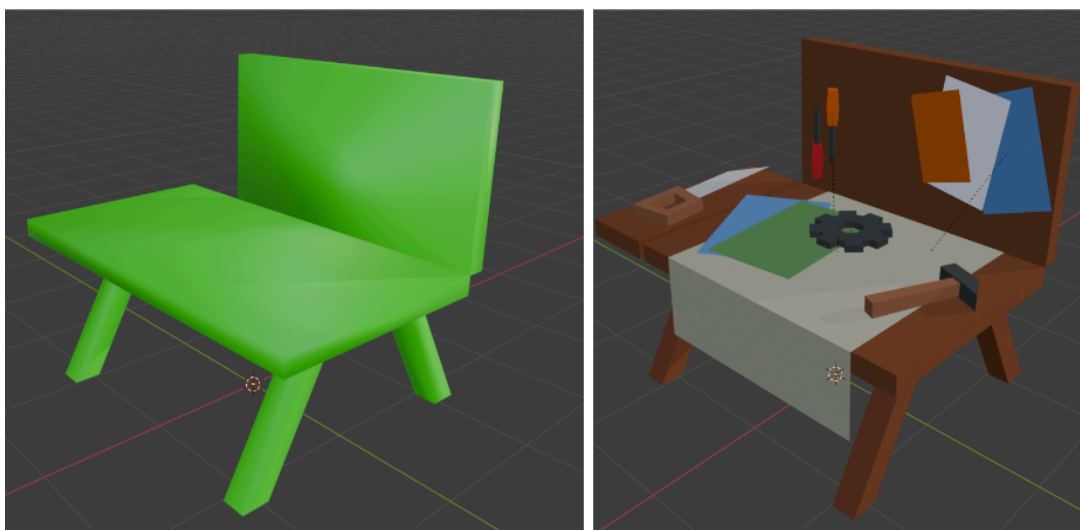


Рис 2.8. Приклад мешу Outline і самої моделі

Джерело: Розроблено автором

Результатом цього комплексного процесу стало створення 3D-моделей, які відповідали сучасним вимогам до мобільних ігор. Завдяки злагодженій організації роботи, увазі до деталей і використанню перевірених підходів вдалося досягти високої якості графіки без шкоди для продуктивності. Це забезпечило привабливий візуальний стиль гри та комфорт для гравців, навіть на пристроях із обмеженими ресурсами.

2.3. Розробка анімацій для взаємодії гравця із об'єктами гри

Процес розробки анімацій у мобільному ігровому додатку LibTake вимагав детального підходу з урахуванням оптимізації для мобільних пристроїв. Основна увага була зосереджена на забезпеченні плавності та природності рухів персонажів, одночасно зменшуючи навантаження на апаратну частину. Цей підхід дозволив досягти високої якості анімації, не перевищуючи технічних обмежень мобільних платформ.

Анімація гуманоїдних персонажів у грі була реалізована за допомогою кісткових анімацій, які є найбільш ефективним методом створення складних рухів у тривимірних моделях. Для цього використовувалася скелетна система, що включає

процес Rigging. Кожна модель персонажа оснащувалася внутрішнім скелетом, що складається з набору кісток, кожна з яких контролює певну частину моделі. Такий підхід дозволяє створювати реалістичні рухи, моделюючи взаємодію кісток, забезпечуючи природність дій персонажів.

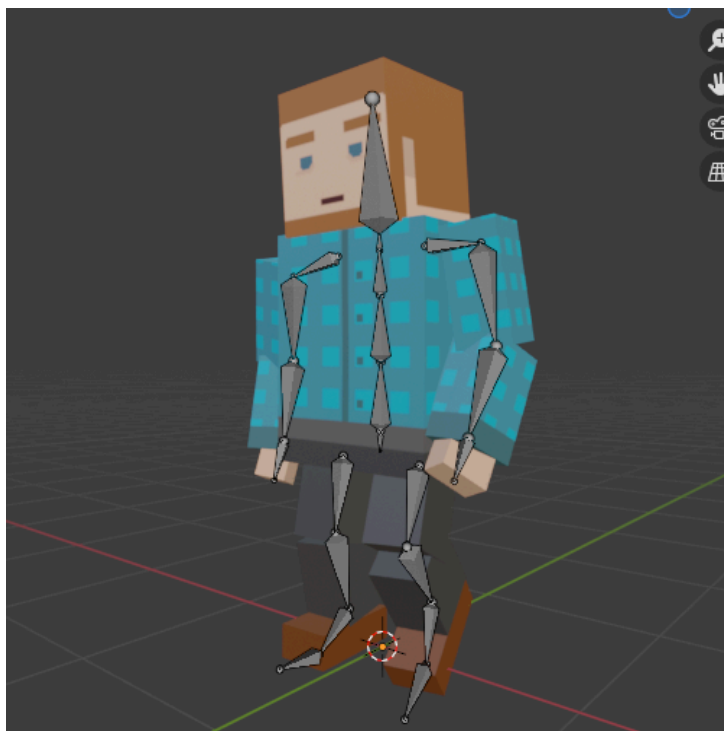


Рис 2.9. Скелет (Rig) головного героя гри

Джерело: Розроблено автором

Особливу увагу було приділено техніці ретаргетингу анімацій, яка дала змогу значно прискорити процес створення рухів. Ця методика дозволяє переносити анімації з однієї моделі персонажа на іншу, навіть за наявності відмінностей у їхній скелетній структурі. У грі LibTake ретаргетинг став ключовим інструментом для економії часу та ресурсів. Використовуючи готові набори анімацій із відкритих бібліотек або попередніх проектів, вдалося скоротити етап розробки, забезпечуючи при цьому високу якість.

Однак навіть із застосуванням ретаргетингу необхідно було виконувати значну доопрацювання анімацій. Це включало видалення зайвих кадрів, корекцію позицій

персонажа та забезпечення плавності переходів між рухами. Ретаргетинг часто створює недоліки, такі як різкі рухи або неприродність, що потребує ручного редагування. Завдяки цьому етапу вдалося досягти високої якості фінальних анімацій, що гармонійно поєднуються із загальним стилем гри.

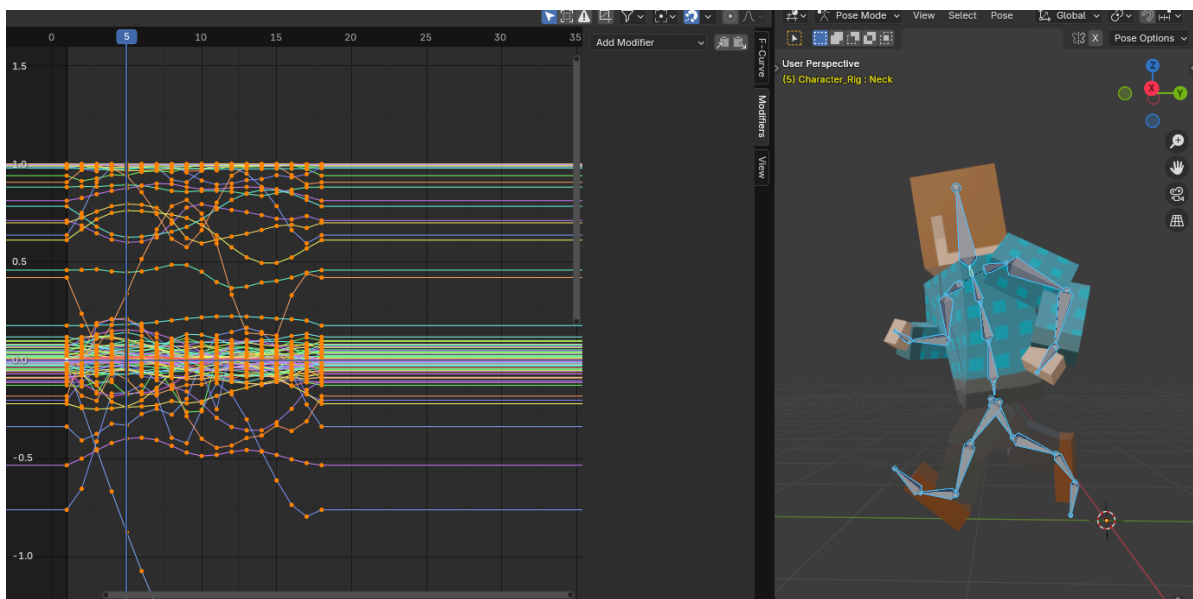


Рис 2.10. Вкладка анімацій в Blender у режимі редагування анімації бігу персонажа

Джерело: Розроблено автором

Схожий підхід застосовується під час роботи з технологією захоплення рухів (motion capture). Ця методика дозволяє записувати рухи людей і переносити їх у цифрові моделі, забезпечуючи реалістичність анімацій. Проте записані рухи часто потребують значної обробки для усунення неточностей, які можуть виникати через помилки обладнання або недоліки у процесі зчитування даних.

Скелетна анімація використовувалася для створення ключових рухів персонажів, які охоплюють різноманітні дії, від ходьби та бігу до взаємодії з об'єктами. Зокрема, було реалізовано такі елементи, як переміщення персонажів у просторі, взяття предметів, а також прості статичні дії, які виконуються, коли персонаж перебуває у стані очікування. Кожна з цих анімацій була ретельно

розроблена, щоб забезпечити їхню відповідність стилістиці гри та її загальному дизайну.

Інтеграція створених анімацій у ігровий процес виконувалася за допомогою спеціальних анімаційних контролерів у Unity. Ці контролери забезпечують керування переходами між різними станами персонажа, такими як ходьба, біг або взаємодія з об'єктами. Завдяки цьому вдалося уникнути різких переходів між анімаціями та досягти плавності, що позитивно впливає на загальне враження від гри.

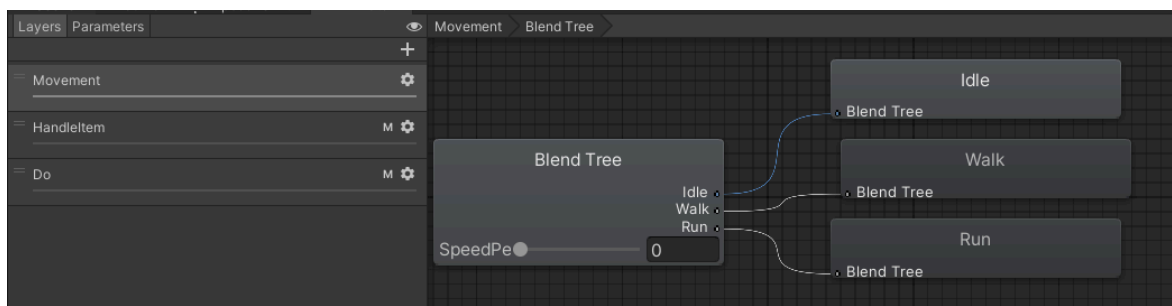


Рис 2.11. Приклад анімаційного контролера в середовищі Unity

Джерело: Розроблено автором

Для оптимізації продуктивності гри рухи об'єктів, які не потребують складної скелетної анімації, були реалізовані за допомогою скриптів. Наприклад, взаємодія з предметами інтер'єру або прості переміщення об'єктів виконувалися через базові трансформації, такі як зміна позиції або поворот, які не створюють значного навантаження на систему. Цей підхід дозволив суттєво зменшити використання ресурсів мобільних пристроїв, одночасно забезпечуючи бажаний рівень інтерактивності.

Таким чином, розробка анімацій у грі LibTake стала результатом поєднання сучасних методик та інструментів із акцентом на оптимізацію для мобільних платформ. Використання кісткових анімацій, ретаргетингу, захоплення рухів і скриптових рішень дозволило створити якісні, природні та плавні рухи персонажів і об'єктів, які відповідають високим стандартам сучасних ігор.

2.4. Створення UX/UI-дизайну з використанням сучасних інструментів

Figma було обрано як основний інструмент для розробки UI елементів завдяки її зручному інтерфейсу, можливостям спільної роботи та підтримці сучасних графічних стандартів. У цьому програмному середовищі були створені векторні зображення, що відображають елементи меню та різні етапи глобальної цілі.

Після завершення проектування користувацького інтерфейсу в середовищі Figma, усі елементи UI були експортовані у форматі PNG із прозорим фоном, що дозволило зберегти гнучкість під час подальшої обробки та інтеграції в гру. Наступним етапом стала додаткова оптимізація графіки у редакторі GIMP, метою якої було зниження навантаження на систему при збереженні високої візуальної якості інтерфейсу.

Ключові етапи обробки включали:

1. Оптимізація розміру файлів – шляхом налаштування параметрів експорту та форматів, вага кожного зображення була зменшена без помітної втрати якості. Це забезпечило швидке завантаження інтерфейсу навіть на малопотужних мобільних пристроях.
2. Згладжування контурів – для уникнення візуальних артефактів у процесі рендерингу були оброблені краї іконок і графічних елементів. Це дозволило досягти чистішого і професійнішого вигляду в грі.

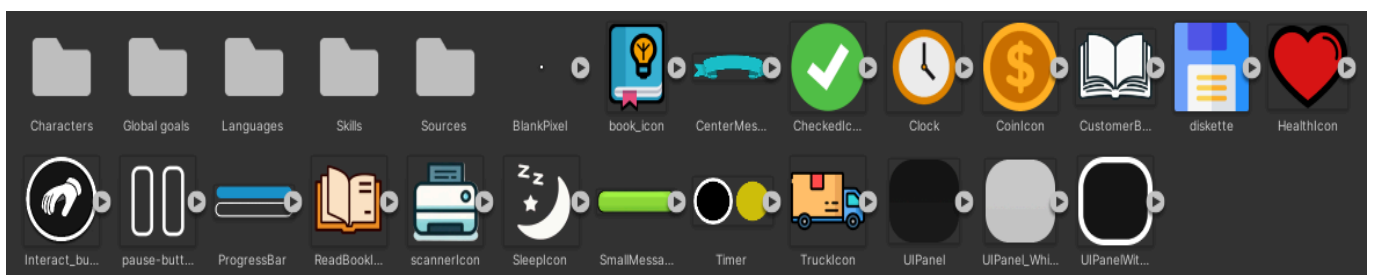


Рис 2.12. Створені та імпортовані основні UI елементи

Джерело: Розроблено автором

Після фінальної обробки всі графічні ресурси були імпортовані до рушія Unity. Під час імпорту застосовувалися спеціальні налаштування, спрямовані на оптимізацію продуктивності та збереження візуальної якості UI:

1. Тип текстури: Sprite (2D and UI) – стандартне налаштування для елементів інтерфейсу.
2. Формат: RGBA, що дозволяє підтримувати прозорість і коректно відображати графіку поверх будь-якого фону.
3. Масштабування: використано фільтр Bilinear для згладжування піксельних країв іконок.
4. Максимальний розмір: обмежений до 128x128 пікселів, що дозволило зменшити розміри текстурного атласу, який автоматично генерується Unity для UI-елементів.
5. Компресія: була вимкнена, щоб уникнути небажаних спотворень та втрати якості, які могли б вплинути на чіткість дрібних деталей.

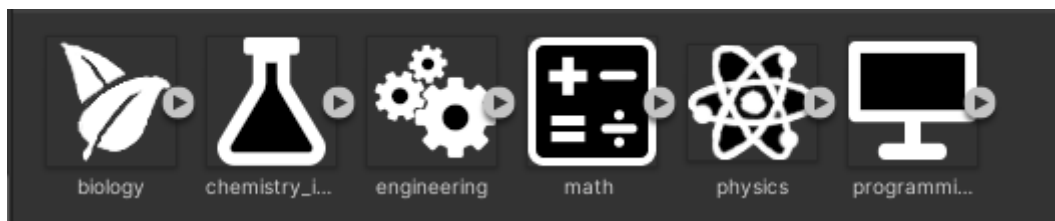


Рис 2.13. Створені та імпортовані UI елементи навичок персонажа
Джерело: Розроблено автором

2.5. Інтеграція контенту в Unity

Завершальний етап створення 3D-контенту та інтерфейсу для гри LibTake був зосереджений на його імпорті в середовище розробки Unity та подальшій інтеграції в ігрову сцену. Цей етап є одним із найважливіших у розробці, адже від якості інтеграції залежить не лише візуальний вигляд гри, але й її продуктивність на різних мобільних платформах. Основними завданнями на цьому етапі стали оптимізація моделей для мобільних пристроїв, правильне налаштування параметрів імпорту, створення префабів та гармонійне розташування контенту у грі.

Окрему увагу приділили текстурам. Щоб уникнути надмірного використання пам'яті, текстури були стиснені за допомогою формату компресії, підтримуваного мобільними пристроями, таких як ASTC або ETC2. Це забезпечило високу якість відображення текстур, мінімізуючи розмір файлів. Крім того, була реалізована технологія atlasing, яка об'єднує декілька текстур в один файл. Це дозволило зменшити кількість викликів до графічного процесора, що суттєво впливає на продуктивність гри.

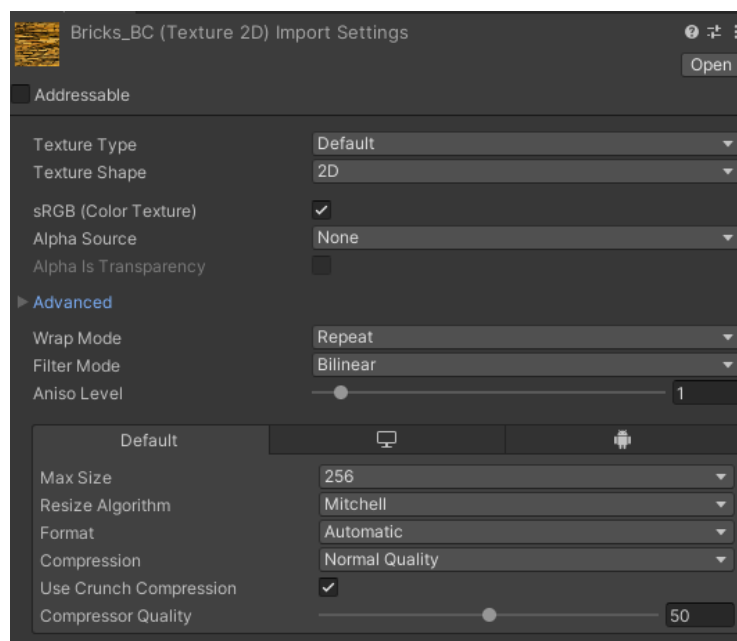


Рис 2.14. Приклад налаштування імпорту текстур

Джерело: Розроблено автором

Після завершення налаштувань імпорту моделі перетворювалися на префаби. У Unity префаби є універсальними шаблонами об'єктів, які можуть бути повторно використані у різних сценах гри. Створення префабів дозволило зберегти всі налаштування моделі, включаючи матеріали, текстури, а також параметри освітлення та фізики. Цей підхід не лише спростив розробку, але й зробив проект більш структурованим, що особливо важливо при роботі з великою кількістю об'єктів.

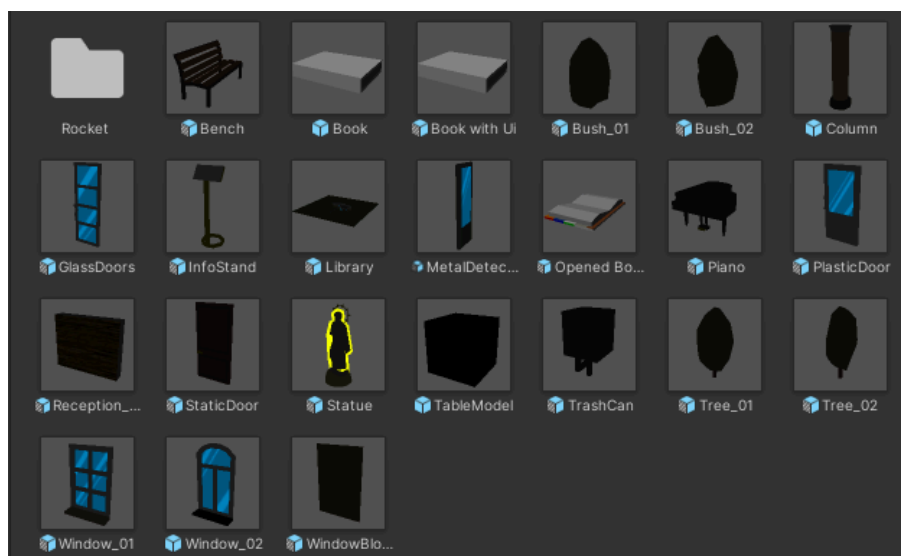


Рис 2.15. Підготовлені prefab-и моделей для розстановки на сцені

Джерело: Розроблено автором

Готові префаби інтегрувалися в ігрову сцену з урахуванням концепції дизайну. Розташування моделей виконувалося таким чином, щоб створити гармонійний ігровий простір, який відповідав загальній стилістиці гри LibTake. Наприклад, об'єкти, з якими гравець взаємодіє, розташовувалися в логічних і зручних для доступу місцях. Це забезпечило комфортну навігацію у грі та покращило загальне враження від ігрового процесу.

2.6. Освітлення та його оптимізація

Одним із ключових підходів до оптимізації продуктивності в грі LibTake стало впровадження технології запікання освітлення (light baking). Цей метод передбачає попередній розрахунок усіх світлових ефектів і тіней для статичних об'єктів із подальшим збереженням результатів у вигляді текстур освітлення (lightmaps). Завдяки цьому значно знижується навантаження на процесор під час гри, оскільки сцена не потребує обробки динамічного освітлення в реальному часі.

Особливу увагу в процесі запікання було приділено оптимізації роздільної здатності світлових карт. Для великих або другорядних об'єктів, які розташовані далеко від гравця або не мають важливого візуального значення, використовувалися

текстури зниженої якості. Це дозволило зекономити пам'ять і зменшити обсяг обчислень. Натомість для ключових елементів сцени – таких як інтер'єрні об'єкти, архітектура або декоративні елементи – застосовувалися карти вищої деталізації, що забезпечило точне та якісне освітлення.

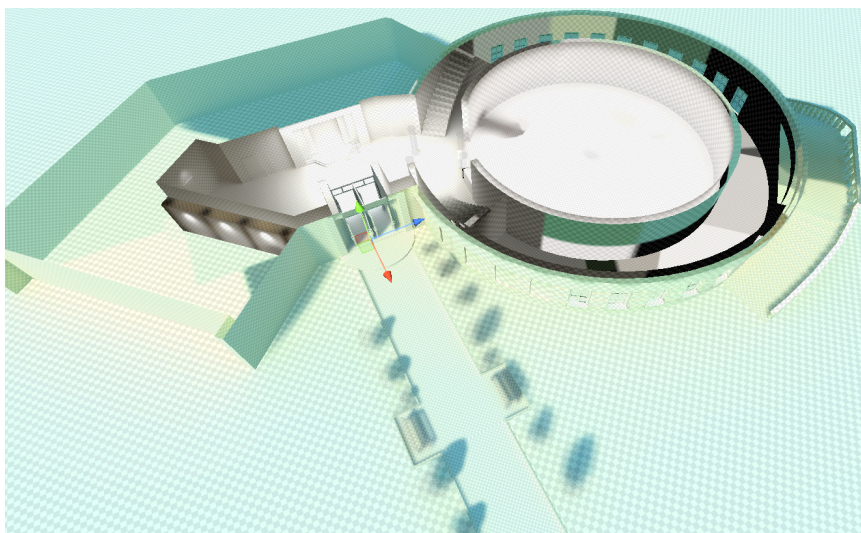


Рис 2.16. Результати запікання карт світла

Джерело: Розроблено автором

Ще одним важливим інструментом для досягнення хорошого освітлення в LibTake стали Light Probes – технологія, яка дозволяє переносити ефекти запеченого світла на динамічні об'єкти, що не можуть бути запечені напряму (наприклад, персонажі або рухомі предмети). Light Probes фіксують інформацію про колірну температуру, інтенсивність освітлення та загальний характер світла у своєму оточенні після запікання. Під час гри ці дані використовуються для взаємодії з динамічними об'єктами, створюючи ефект відблисків, м'яких тіней і атмосферного освітлення, що відповідає статичному середовищу.

Розміщення Light Probes на сцені здійснювалося вручну – переважно у критичних зонах: поруч із джерелами світла, в затемнених ділянках і поблизу точок взаємодії динамічних і статичних елементів. Такий підхід дозволив досягти плавних переходів між освітленими зонами і створити відчуття цілісності освітлення сцени.

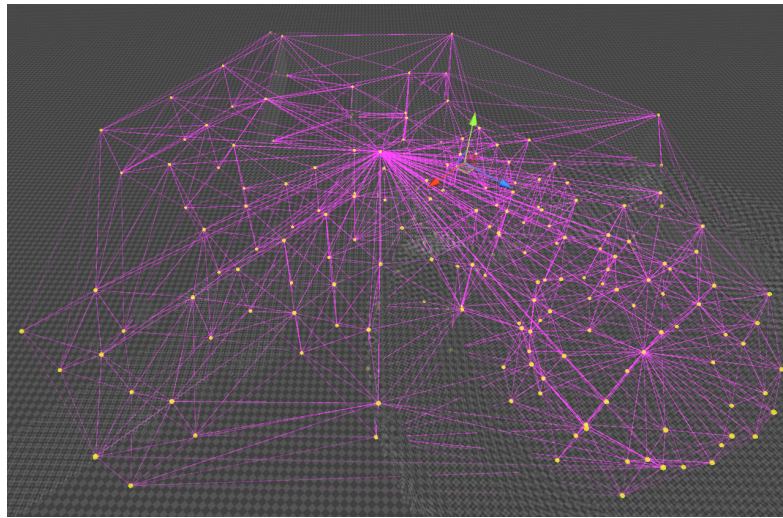


Рис 2.17. Розміщенні Light Probes на сцені

Джерело: Розроблено автором

Завдяки раціональному використанню запеченого освітлення, адаптивному підходу до роздільної здатності світлових карт та ретельно налаштованим Light Probes, вдалося забезпечити як візуальну якість, так і високу продуктивність LibTake на широкому спектрі мобільних пристроїв. Цей комплексний підхід до освітлення став одним із ключових факторів, що забезпечив плавність гри та занурення у візуальне середовище без зайвих витрат ресурсів.



Рис 2.18. Фінальний вигляд сцени в Unity

Джерело: Розроблено автором

Висновок до розділу 2

У розділі описано процес створення 3D-контенту та UI для мобільної гри LibTake. Застосування стилю LowPoly дало змогу досягти візуальної привабливості при низькому навантаженні на систему. Реалізовано ефект Outline методом Inverted Hull, що покращив сприйняття моделей.

Опрацьовано текстури, UV-Mapping та анімацію з ретаргетингом. Налаштоване освітлення в Unity забезпечило потрібний настрій сцени без перевантаження. Інтерфейс розроблений у єдиному стилі, з урахуванням зручності для користувача на мобільних пристроях.

РОЗДІЛ 3. Оновлення ігрового додатку, оцінка результатів та перспективи розвитку проекту

4.1. Створення оновлення з новою глобальною ціллю для ігрового додатку

Впровадження нового контенту є ключовим фактором підтримки інтересу користувачів до мобільних ігрових додатків. На основі аналізу взаємодії користувачів з додатком та отриманого зворотного зв'язку було виявлено потребу в розширенні ігрового досвіду через додавання нової глобальної цілі. Вона повинна гармонійно інтегруватися в існуючу механіку гри, водночас пропонуючи користувачам свіжий контент та нові елементи геймплею.

Глобальна ціль "Лікування зомбі" була розроблена як тематичне доповнення, що розширює сюжетну лінію гри та додає нові візуальні елементи. Особлива увага в процесі розробки приділялася технічній оптимізації всіх елементів для забезпечення стабільної продуктивності на цільових мобільних платформах. Розробка кожного компонента супроводжувалась ретельним тестуванням та оцінкою впливу на загальну продуктивність додатку.

Для нової глобальної цілі гри необхідно було створити анімований шейдер для моделі комп'ютера який проводить обчислення. Процес створення анімованого шейдера за допомогою ShaderGraph є складним і методично організованим підходом до формування візуальних ефектів, заснованим на використанні взаємопов'язаних вузлів (нодів). Кожен із цих нодів виконує певну функцію в загальній архітектурі шейдера, а їхня взаємодія дозволяє досягти бажаного результату. У рамках цього дослідження було розроблено шейдер для анімації цифрових символів, що безперервно трансформуються на текстурній поверхні, створюючи враження потоку даних.

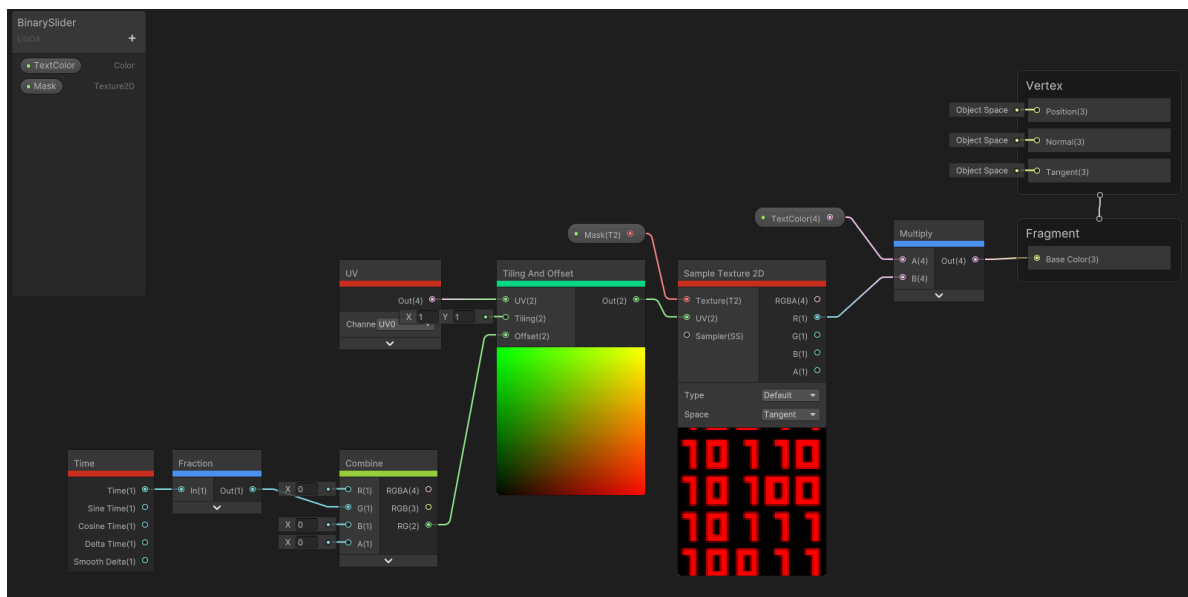


Рис 3.1. Графічна візуалізація структури нодів, використаних для створення анімаційного шейдера

Джерело: Розроблено автором

Однією з ключових особливостей вибору такого підходу є обрання Unlit шейдера, що стало обґрунтованим рішенням з огляду на особливості проекту. Unlit шейдери функціонують незалежно від освітлення сцени, що дозволяє уникнути складних розрахунків щодо світлотіні, зосереджуючись на точному відтворенні кольорових характеристик та анімаційних ефектів, не навантажуючи систему додатковими обчислювальними витратами.

Аналіз архітектури шейдера виявляє кілька ключових компонентів, які відповідають за формування кінцевого візуального ефекту. Першим етапом є використання ноду Time, який генерує числові значення, що змінюються залежно від часу рендерингу сцени. Цей нод є основою для створення динамічних анімаційних ефектів, оскільки забезпечує безперервний потік часових даних.

Дані, отримані від ноду Time, передаються до ноду Fraction, який нормалізацією значень. Нормалізація перетворює будь-яке число у відрізок від 0 до 1, усуваючи цілу частину числа. Це критично важливо для забезпечення циклічності анімації,

гарантуючи, що зміни текстурних координат залишаються в межах стандартного діапазону значень і виключають можливість накопичення великих значень, які можуть призвести до неконтрольованої поведінки шейдера.

Далі нормалізоване значення передається в нод Combine, де воно інтегрується в двовимірний вектор координат, причому значення від Fraction передається в зелений канал (G), а червоний канал (R) залишається незмінним. Така структура дозволяє зберегти стабільну горизонтальну координату текстури, при цьому вертикальна координата змінюється динамічно.

Нод Combine є гнучким інструментом для створення векторних значень різних розмірностей, що дозволяє ефективно працювати з даними в межах однієї шейдерної мережі.

Після цього сформований вектор передається до ноду TilingAndOffset, який виконує масштабування (Tiling) та зміщення (Offset) текстури. Масштабування дозволяє змінювати розмір текстурного елемента, а зміщення — переміщувати текстуру у просторі UV-координат, що є основою для створення анімації. Це забезпечує рух текстури по вертикалі, що надає ілюзію зміни цифрових символів.

Особливість ноду TilingAndOffset полягає в тому, що ці операції виконуються в просторі UV-координат, не впливаючи на фізичну геометрію об'єкта, що знижує навантаження на систему та підвищує ефективність обчислень.

Після трансформації координат текстури, вони передаються до ноду SampleTexture2D, який забезпечує доступ до текстурних даних з урахуванням нових координат. Цей нод здійснює інтерполяцію між сусідніми пікселями текстури для забезпечення плавного переходу при зміщенні координат. SampleTexture2D також дозволяє контролювати параметри текстури, такі як рівень деталізації (LOD), фільтрацію та обгортання (wrap mode).

Для коректного відображення результату застосовується колірна модуляція за допомогою ноду Multiply, який виконує множення кольорових даних, отриманих з текстури, з кольором, заданим в ноді Color. Це дозволяє точно налаштувати колір анімованих символів відповідно до загальної кольорової схеми сцени.



Рис 3.2. Інтегрована модель комп'ютера з анімованим матеріалом

Джерело: Розроблено автором

Однією з важливих характеристик розробленого шейдера є його належність до типу Unlit, що має суттєве практичне значення. Оскільки цей тип шейдера не враховує вплив освітлення на матеріал, візуальні властивості залишаються стабільними незалежно від змін умов освітлення сцени чи положення камери. Це ідеально підходить для об'єктів, які повинні випромінювати світло, таких як екрани пристроїв або голографічні проекції.

Використання Unlit шейдера також забезпечує стабільну продуктивність, оскільки відсутність необхідності в обчисленні світлотіні значно знижує навантаження на графічний процесор, що особливо важливо для мобільних платформ та систем віртуальної реальності.

Після створення візуальних ефектів для комп'ютерної моделі, наступним завданням стала розробка моделей персонажів для нової глобальної цілі. Необхідно було створення моделі зомбі було адаптовано вже існуючу модель відвідувача, що дозволило зберегти стилістичну єдність і зменшити час розробки. Модель має

оптимізовану топологію та топологію окремим об'єктом для Outline ефекту, що є прийнятним для персонажа в мобільній грі. Основні зміни включали:

1. Модифікацію текстури шкіри з заміною на блідо-зелений відтінок, що асоціюється з зараженням вірусом
2. Зміну назви анімаційного скелету та його дочірніх об'єктів для коректної інтеграції в ігровий рушій
3. Оптимізацію вагових карт для зменшення артефактів анімації при збереженні низько полігональної сітки

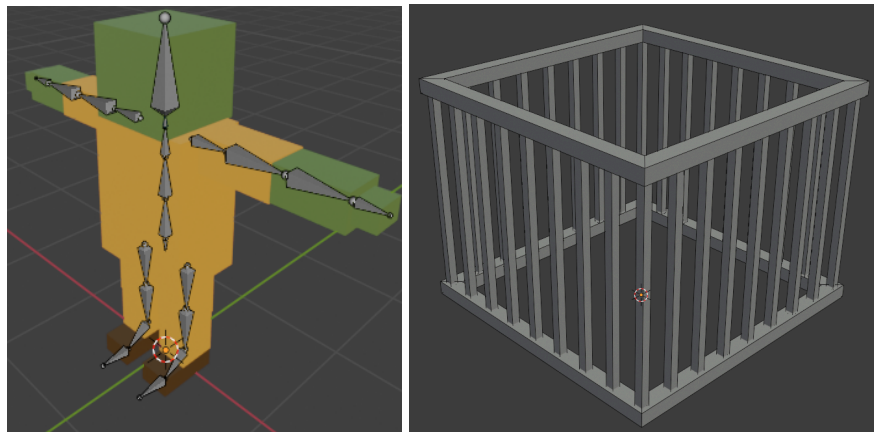


Рис 3.3. Моделі зомбі та клітки для нього

Джерело: Розроблено автором

У рамках глобальної цілі, яка полягає у створенні вакцини від зомбі-вірусу, було створено та імпортовано набір UI елементів, що покликані доповнити ігровий досвід. Серед створених UI елементів:

1. Іконка мікроскопу: зображення, яке ілюструє перший етап, пов'язаний із ідентифікацією вірусу. Іконка має мінімалістичний стиль із плавними лініями та високою читабельністю.
2. Іконка ДНК: графічний елемент, що представляє аналіз структури вірусу. Дизайн виконано у вигляді подвійної спіралі, яка асоціюється з молекулою ДНК.

3. Іконка колби: цей елемент символізує синтез антигенів. Колба зображена з нанесеним знаком черепа та рідиною, яка випускає бульбашки. Сама рідина виконана у вигляді градієнту від світло-зеленого до темного відтінку.
4. Іконка формули: має строгий вигляд із зображенням хімічної формули
5. Іконка комп'ютера: відображає процес комп'ютерної симуляції, що виконується в грі. Іконка включає зображення екрану із символами нулів і одиниць, що імітує цифровий код.
6. Іконка шприца: є завершальним елементом, що використовується на етапі ін'єкції вакцини. Зображення виконано з акцентом на медичну символіку.

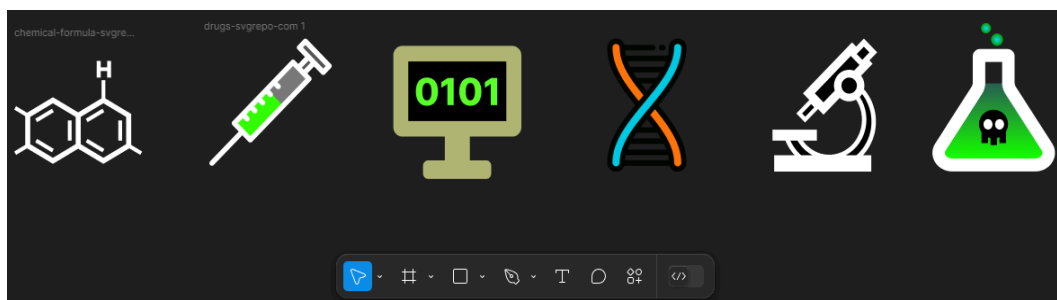


Рис 3.4. Створені UI елементи для глобальної цілі “Лікування зомбі”

Джерело: Розроблено автором

Імпортовані елементи були розміщені у відповідних частинах інтерфейсу гри, таких як меню, екрани підказок та інформаційні панелі. Їх інтеграція була виконана з використанням Canvas у Unity, що дозволило коректно масштабувати UI залежно від роздільної здатності пристрою.

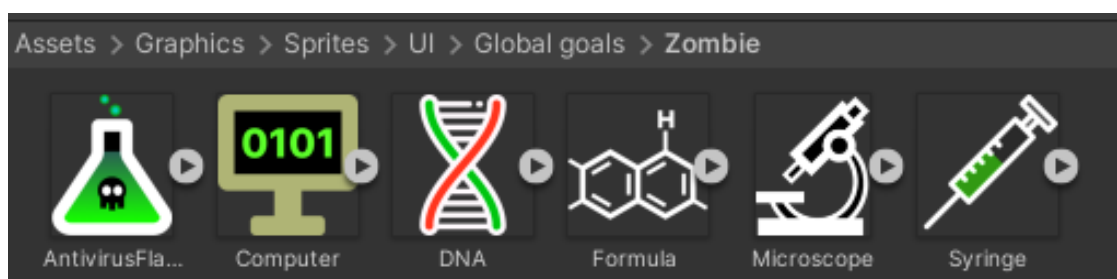


Рис 3.5. Оброблені в GIMP та імпортовані UI елементи для нової глобальної цілі гри

Джерело: Розроблено автором

4.2. Доповнення контентом на основі зворотного зв'язку

У процесі тестування ігрового додатку було систематично зібрано та проаналізовано зворотний зв'язок від користувачів, що дозволило виявити ключові напрямки для вдосконалення проекту. Найбільш значущі рекомендації користувачів стосувалися необхідності включення нової глобальної цілі для підвищення різноманітності геймплею, розширення можливостей для взаємодії з контентом, а також модернізації процесу вибору персонажа для забезпечення більшої персоналізації ігрового досвіду. На основі отриманих відгуків було розроблено та імплементовано комплекс функціональних та графічних доповнень.

Для реалізації нової глобальної цілі "Рекурсія" було застосовано інноваційний підхід із використанням ShaderGraph – візуального редактора шейдерів у середовищі Unity. Завдяки цьому інструменту було створено спеціалізований шейдер, який забезпечує циклічне відображення різних станів текстури, імітуючи послідовну зміну етапів розробки ігрового проекту.

Створений шейдер функціонує за принципом фліпбука (flipbook), де цілісне зображення складається з окремих фреймів, організованих у вигляді сітки. Перевагою даного підходу є можливість відтворення складних анімаційних послідовностей без необхідності використання об'ємних ресурсів системи. Шейдер забезпечує плавний перехід між різними станами текстури, створюючи цілісну візуальну анімацію процесу розробки гри.



Рис 3.6. Створений шейдер фліпбука для симуляції етапів розробки в глобальній цілі
Джерело: Розроблено автором

Імплементація даного шейдера дозволила суттєво оптимізувати використання ресурсів, оскільки замість зберігання численних окремих текстур або використання ресурсномістких скриптів для анімації, всі візуальні стани обробляються через єдиний матеріал. Такий підхід особливо важливий для мобільних пристроїв з обмеженими обчислювальними можливостями.

В рамках імплементації глобальної цілі "Рекурсія" було розроблено спеціалізовану 3D-модель комп'ютера, яка виступає центральним інтерактивним об'єктом. При створенні моделі особлива увага приділялась оптимізації геометрії для забезпечення високої продуктивності на цільових мобільних платформах.

Модель розроблена з використанням мінімалістичної топології та складається з трьох ключових компонентів: клавіатура, системний блок, та монітор – центральний компонент, на який накладено анімований матеріал із раніше створеним шейдером. Модель містить лише необхідні полігони для збереження

характерного вигляду комп'ютера, при цьому загальна кількість полігонів була мінімізована для забезпечення стабільного відображення на мобільних пристроях.



Рис 3.7. Створена модель комп'ютера для глобальної цілі “Рекурсія”

Джерело: Розроблено автором

Для забезпечення інтуїтивно зрозумілого користувацького досвіду було розроблено комплекс UI елементів, що візуалізують прогрес у досягненні глобальної цілі "Рекурсія". Кожен етап на шляху до досягнення цілі представлений унікальним графічним елементом, що відображає відповідну стадію розробки гри.

Розроблені UI елементи включають візуальні репрезентації наступних етапів: створення графіки, розробка механік, створення музики – графічний елемент, пов'язаний із аудіо-компонентами гри, прототипування, тестування, фінальний комп'ютер.

При створенні даних елементів особлива увага приділялась стилістичній єдності з основним візуальним оформленням гри та чіткій ідентифікації кожного етапу. Для всіх графічних елементів було проведено оптимізацію через налаштування параметрів імпорту в Unity, що дозволило значно зменшити розмір кінцевого додатку.

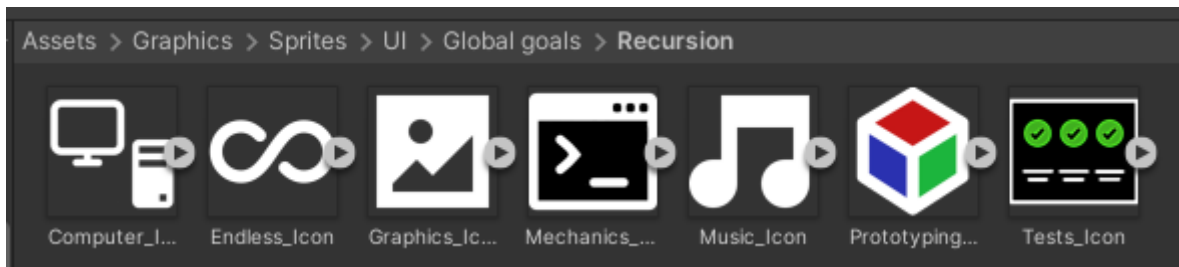


Рис 3.8. Створені UI елементи для глобальної цілі “Рекурсія”

Джерело: Розроблено автором

На основі зворотного зв'язку користувачів, які висловили бажання мати більше варіантів для персоналізації ігрового процесу, було розроблено розширену систему вибору персонажа. Ця система включає чотири варіанти аватарів, що надає користувачам більше можливостей для самоідентифікації в ігровому світі.

Для кожної моделі були створені відповідні UI елементи, які використовуються в меню вибору персонажа. Ці елементи розроблені з урахуванням принципів UX/UI дизайну для забезпечення інтуїтивно зрозумілого процесу вибору. Візуальні репрезентації виконані в єдиному стилі для збереження цілісності інтерфейсу гри.

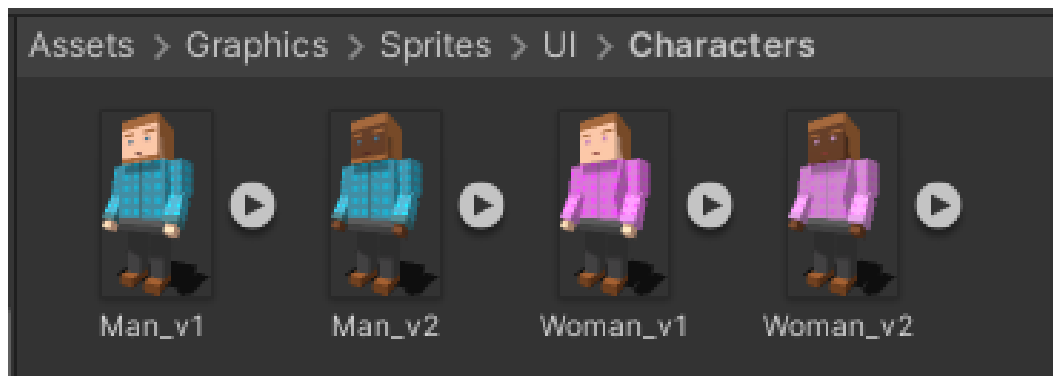


Рис 3.9. Створені UI елементи для вибору персонажу

Джерело: Розроблено автором

4.3. Оптимізація продуктивності

Перший етап оптимізації продуктивності проєкту був зосереджений на попередній обробці 3D-моделей безпосередньо у спеціалізованому редакторі перед їх імпортом до ігрового рушія Unity. Цей етап мав виключне значення для

забезпечення оптимальної продуктивності кінцевого продукту на мобільних пристроях з обмеженими ресурсами. Процес включав комплекс взаємопов'язаних технік оптимізації геометрії та підготовки моделей для ефективного використання в ігровому середовищі.

Одним з найважливіших аспектів оптимізації стало видалення невидимих полігонів – елементів геометрії, які не видно користувачеві під час гри. Такі полігони часто виникають у процесі моделювання і можуть бути розташовані всередині об'єктів, під поверхнями інших елементів, або в місцях, недоступних для огляду камерою. На рисунку 3.5 представлено приклад такої оптимізації, де в червоній області позначено невидимі полігони, що підлягали видаленню.

Даний підхід дозволив значно зменшити загальну кількість полігонів у сцені, що безпосередньо вплинуло на швидкість рендерингу та використання пам'яті. За результатами тестування, видалення невидимих полігонів зменшило загальну кількість геометрії приблизно на 15-30% залежно від складності моделі, що суттєво покращило продуктивність без жодних візуальних втрат.

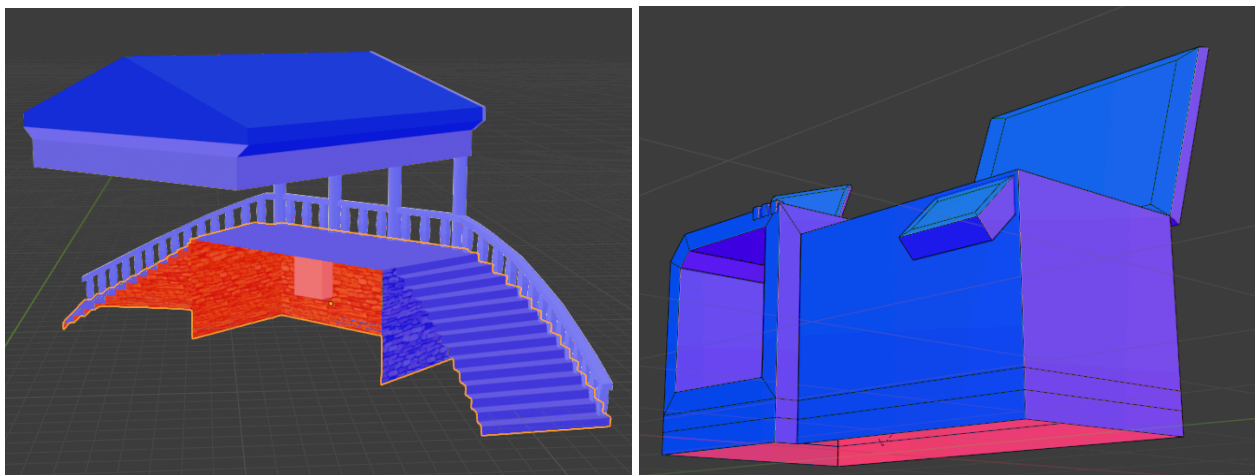


Рис 3.10. Приклад оптимізованих моделей шляхом видалення невидимих полігонів
(в червоній області)

Джерело: Розроблено автором

Наступний етап передбачав оптимізацію безпосередньо в Unity за допомогою спеціальних налаштувань імпорту, що дозволяло суттєво покращити продуктивність гри. Завдяки комплексному підходу вдалося досягти оптимального співвідношення між якістю 3D моделей та швидкістю гри на мобільних пристроях.

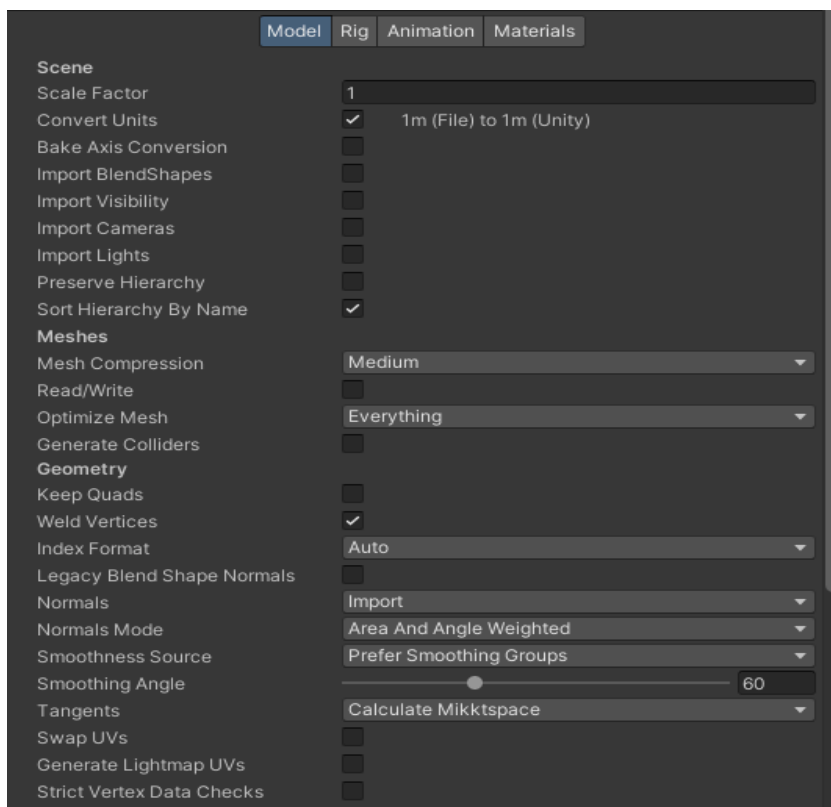


Рис 3.11. Приклад налаштування моделі

Джерело: Розроблено автором

Для імпорту статичних моделей, таких як будівлі, ландшафтні елементи та декорації, було розроблено оптимізований профіль налаштувань, що дозволяє значно зменшити обсяг даних без суттєвих втрат у якості. Оскільки ці об'єкти не потребують анімації або динамічних змін форми, імпорт низки параметрів було відключено.

Технологія BlendShapes, яка використовується для анімації мимічних виразів або інших деформацій, виявилася непотрібною для статичних моделей, тому її вимкнення дозволило зменшити розмір моделей у пам'яті приблизно на 15-20%. Це

також допомогло скоротити час завантаження ресурсів, що позитивно позначилося на загальній продуктивності. Крім того, відсутність обробки BlendShapes зменшила навантаження на центральний процесор, що особливо важливо для мобільних платформ, де кожен додатковий ресурс може впливати на продуктивність.

Імпорт параметрів Visibility також був вимкнений. Видимість, задана в 3D-редакторі, часто є зайвою у фінальній ігровій сцені, тому управління нею було передано безпосередньо Unity.

Камери, створені в процесі моделювання, також не імпортуються, оскільки вони зазвичай не відповідають специфіці геймплею. Для проекту була розроблена власна система камер, оптимізована під ігрові механіки, що дозволило уникнути дублювання функцій та зменшити загальну кількість об'єктів у сцені. Це сприяло уніфікації налаштувань рендерингу та полегшенню подальшого налаштування графіки.

Так само не імпортувалися світлові джерела, оскільки освітлення сцени проектувалося безпосередньо в Unity. Це дозволило забезпечити узгодженість візуального стилю та використовувати оптимізовані методи освітлення, такі як Baked Lighting, що є критично важливим для мобільних платформ. Відмова від імпорту світла також зменшила навантаження на графічний процесор, що дало змогу підвищити продуктивність рендерингу сцени та покращити стабільність гри на слабших пристроях.

Анімаційні дані для статичних об'єктів виявилися абсолютно непотрібними, тому їх імпорт був вимкнений. Це рішення дозволило зменшити загальний розмір файлів проекту, прискорити завантаження ресурсів та звільнити оперативну пам'ять для більш критичних елементів гри. Відсутність непотрібних анімаційних даних також спростила оптимізацію сцени, оскільки ігровий рушій не витрачав ресурси на обробку зайвих компонентів.

Щодо параметрів, які залишилися активними, було обрано середній рівень стиснення сітки (Mesh Compression Medium). Він забезпечує оптимальний баланс між зменшенням розміру моделі та збереженням візуальної якості. Стиснення впливає на точність координат вершин, нормалей та текстурних координат, що дозволило зменшити розмір моделі в пам'яті приблизно на 30-40%. Тести показали, що перехід на високий рівень стиснення давав лише незначний виграш у розмірі файлу (5-7%), проте призводив до появи артефактів, особливо помітних на геометрії з плавними вигинами. Водночас низький рівень стиснення не забезпечував достатньої економії ресурсів, що робило його непридатним для цільових пристроїв середнього класу.

Було також активовано повну оптимізацію сітки (Optimize Mesh Everything). Ця опція дозволила Unity перебудувати структуру моделі для ефективного рендерингу. В процесі оптимізації відбувалося злиття суміжних вершин з однаковими атрибутами, оптимізація порядку індексів для покращеного використання кешу вершин, перерахунок нормалей для якісного освітлення та покращення розташування трикутників, що зменшило кількість викликів рендерингу. Практичні випробування показали, що завдяки цим заходам продуктивність рендерингу зросла на 15-25% у складних сценах, що містять велику кількість однотипних об'єктів. Це особливо важливо для мобільних платформ, де пам'ять та обчислювальні ресурси обмежені, а ефективне використання ресурсів впливає на стабільність та швидкість роботи гри.

Для статичних об'єктів було встановлено режим рігу як None. Цей параметр сигналізує рушію, що моделі не потребують додаткової обробки скелетної анімації, що значно скорочує обчислювальні ресурси.

Висновок до розділу 3

У третьому розділі було проведено комплексне оновлення ігрового додатку шляхом впровадження нової глобальної цілі "Лікування зомбі", що дозволило

розширити ігровий контент та підвищити рівень залучення користувачів. Створення анімованого шейдера за допомогою ShaderGraph забезпечило візуальну привабливість без значного навантаження на систему, що є критично важливим для мобільних платформ. Вибір Unlit шейдера дозволив оптимізувати обчислювальні процеси та гарантувати стабільність відображення матеріалів незалежно від змін освітлення сцени.

На основі користувацького зворотного зв'язку було реалізовано нову глобальну ціль "Рекурсія", яка доповнила ігровий процес інноваційними механіками та забезпечила додаткову різноманітність геймплею. Розробка спеціалізованого шейдера фліпбука сприяла ефективній анімації без використання ресурсомістких рішень.

Особливу увагу приділено розширенню можливостей персоналізації за допомогою нової системи вибору персонажів, що позитивно вплинуло на користувацький досвід. Додатково було створено набір нових UI елементів, стилістично узгоджених із загальним оформленням гри, що підвищило естетичну привабливість інтерфейсу.

Проведена оптимізація 3D-моделей, зокрема видалення невидимих полігонів, дозволила суттєво зменшити обсяг обчислень та забезпечити стабільну роботу додатку навіть на пристроях із обмеженими технічними характеристиками.

Таким чином, оновлення, реалізовані у межах цього етапу, не лише збагатили ігровий контент, але й суттєво підвищили продуктивність додатку, що створює надійне підґрунтя для подальшого розвитку проекту.

ВИСНОВКИ

У кваліфікаційній роботі успішно вирішено актуальне науково-практичне завдання розробки 3D-контенту та інтерактивного інтерфейсу для мобільного ігрового додатку Libtake з використанням сучасних інструментів моделювання, текстурювання та інтеграції. Проведене дослідження дозволило досягти поставленої мети та отримати низку важливих результатів, що мають як теоретичну, так і практичну цінність.

Аналіз сучасного стану мобільної ігрової індустрії підтвердив стабільне зростання ринку та підвищення вимог користувачів до якості графічного контенту і зручності інтерфейсів. Встановлено, що ключовим чинником успішності мобільних 3D-ігор є забезпечення балансу між візуальною привабливістю та продуктивністю, що актуалізує завдання оптимізації ігрового контенту для обмежених ресурсів мобільних пристроїв.

У процесі роботи досліджено й проаналізовано сучасні інструменти створення 3D-графіки та інтерактивних інтерфейсів. Обґрунтовано вибір технологічного стеку, що включає Blender для моделювання та анімації, GIMP для текстурювання, Figma для прототипування інтерфейсу та Unity для інтеграції контенту. Такий вибір забезпечив необхідний баланс між функціональністю, гнучкістю та економічною ефективністю розробки.

Розроблена методологія створення 3D-контенту для мобільних платформ охоплює етапи аналізу технічних обмежень пристроїв, розробку концепт-арту, створення оптимізованих моделей із низькою кількістю полігонів, текстурювання, налаштування матеріалів і шейдерів, програмування інтерактивних елементів та оптимізацію продуктивності.

У межах роботи створено комплексний набір 3D-контенту адаптованих під обмеження мобільних платформ. Усі моделі оптимізовані відповідно до

встановлених обмежень кількості полігонів та розмірів текстур, що забезпечило стабільну продуктивність додатку.

Спроековано та реалізовано інтуїтивно зрозумілий інтерактивний UX/UI-інтерфейс, оптимізований для використання на тач-екранах мобільних пристроїв. Проведене юзабіліті-тестування засвідчило високу ефективність розробленого інтерфейсу: більшість користувачів змогли виконати основні дії без потреби в інструкціях, що свідчить про якість реалізації дизайну.

Значну увагу було приділено технічній оптимізації продуктивності ігрового додатку. Застосування текстурних атласів дозволило суттєво зменшити кількість викликів рендерингу (draw calls), оптимізоване освітлення на основі запеченого світла знизило навантаження на процесор мобільних пристроїв, а використання анімованих шейдерів, створених за допомогою ShaderGraph, забезпечило складні візуальні ефекти при мінімальному споживанні ресурсів.

Розроблено та впроваджено нове оновлення ігрового додатку з глобальною ціллю "Лікування зомбі", яке включало створення оригінальних 3D-моделей, спеціалізованих UI-елементів та інтеграцію нових ігрових механік. Впровадження цього оновлення сприяло покращенню утримання користувачів та підтвердило ефективність запропонованих рішень.

Практичне значення результатів дослідження підтверджується успішною інтеграцією розробленого контенту в ігровий додаток Libtake та позитивними відгуками користувачів. Розроблені методики та рішення можуть бути використані при створенні інших мобільних ігор, а також мають навчальну цінність для підготовки фахівців у сфері розробки комп'ютерних ігор.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Unity Documentation. Optimizing Graphics Performance. Режим доступу: <https://docs.unity3d.com/2022.3/Documentation/Manual/OptimizingGraphicsPerformance.html> (дата відвідування 01.02.2025)
2. Unity Core RP Library Package manual. Режим доступу: <https://docs.unity3d.com/Packages/com.unity.render-pipelines.core@12.1/manual/> (дата відвідування 01.02.2025)
3. Unity URP Package manual. Режим доступу: <https://docs.unity3d.com/Packages/com.unity.render-pipelines.universal@11.0/manual/> (дата відвідування 01.02.2025)
4. Unity Blog. Optimize your mobile game performance. Режим доступу: <https://unity.com/blog/games/optimize-your-mobile-game-performance-expert-tips-on-graphics-and-assets> (дата відвідування 01.02.2025)
5. Unity Blog. Get our 2022 LTS best practice guides for optimizing games. Режим доступу: <https://unity.com/blog/engine-platform/updated-2022-lts-best-practice-guides> (дата відвідування 02.02.2025)
6. Unity Blog. Introduction to the Universal Render Pipeline for advanced Unity creators. Режим доступу: <https://unity.com/resources/introduction-universal-render-pipeline-for-advanced-unity-creators-2022lts> (дата відвідування 02.02.2025)
7. Unity Blog. Unity Shader Variants Optimization & Troubleshooting Tips. Режим доступу: <https://unity.com/blog/engine-platform/shader-variants-optimization-troubleshooting-tips> (дата відвідування 09.02.2025)
8. Blender Documentation. Blender Foundation. Режим доступу: <https://docs.blender.org/> (дата відвідування 15.02.2025)
9. GIMP Documentation. GIMP Team. Режим доступу: <https://www.gimp.org/docs/> (дата відвідування 18.02.2025)
10. AmbientCG. Free Textures for 3D Projects. Режим доступу: <https://ambientcg.com/> (дата відвідування 20.02.2025)

11. Unreal Engine Documentation. Performance Guidelines for Mobile Devices.

Режим доступу:

<https://dev.epicgames.com/documentation/en-us/unreal-engine/performance-guidelines-for-mobile-devices-in-unreal-engine> (дата відвідування 08.03.2025)

12. Cgtrader Blog. A Comprehensive Guide to Low Poly 3D Models and its

Application. Режим доступу:

<https://www.cgtrader.com/blog/a-comprehensive-guide-to-low-poly-3d-models-and-its-application> (дата відвідування 14.03.2025)

13. Polycount Wiki. Catalog of information aimed at game development artists.

Режим доступу: <http://wiki.polycount.com/wiki/Polycount> (дата відвідування 15.03.2025)

14. NVIDIA CG Documentation. NVIDIA CG language math reference. Режим

доступу: <https://developer.download.nvidia.com/cg/> (дата відвідування 15.03.2025)

15. Game UI Database. The Ultimate reference tool for game interface designers.

Режим доступу: <https://www.gameuidatabase.com/> (дата відвідування 15.03.2025)

16. Godot Shaders. Community-driven shader library for the Godot game engine.

Режим доступу: <https://godotshaders.com/> (дата відвідування 20.03.2025)

17. Essential GameDev Resources by ShadowShardTools. Curated resources for

game developers. Режим доступу:

<https://shadowshardtools.github.io/Essential-Gamedev-Resources/> (дата відвідування 10.04.2025)