

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)


Кафедра Медіасистем та технологій
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Проектування ігрового 3D-середовища українського дворику
(тема)


Виконав:
здобувач 4 року навчання,
групи ВППС-21-4


Юлія МАНСКОВА
(прізвище, ініціали)

Спеціальність 186 Видавництво та поліграфія
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма
Видавничо-поліграфічна справа
(повна назва освітньої програми)

Керівник 
доц. Ірина ТАБАКОВА
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри МСТ

Жанна ДЕЙНЕКО
(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
Кафедра _____ Медіасистем та технологій _____
Рівень вищої освіти _____ перший (бакалаврський) _____
Спеціальність _____ 186 Видавництво та поліграфія _____
Тип програми _____ Освітньо-професійна _____
Освітня програма _____ Видавничо-поліграфічна справа _____
(шифр і назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри МСТ _____
(підпис)
« 19 » травня 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

здобувачеві _____ *Мансковій Юлії Юріївні* _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ *Проектування ігрового 3D-середовища українського дворики* _____

Затверджена наказом по університету від _____ 19 травня 2025 р. № 385 Ст _____


2. Термін подання студентом роботи до екзаменаційної комісії _____ 4 червня 2025 р. _____

3. Вихідні дані до роботи
Тип видання: Ігрове тривимірне середовище; _____
Середовище поширення: Інтернет; _____
Характер ігрового простору: 3D. _____

4. Перелік питань, що потрібно опрацювати в роботі
Аналіз предметної області та завдання на кваліфікаційну роботу; Вибір програмного забезпечення; Розробка практичної частини; Економічна частина; Висновки. _____

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п. 5 включається до завдання за рішенням випускової кафедри)
Титульний слайд; Актуальність роботи; Мета та задачі роботи; Аналіз аналогів; Вибір інструментальних засобів та технологій; Проектування 3D-моделей; Розробка UV-розгортки; Запікання процедурних карт; Текстурування; Збирання сцени в ігровому рушії; Тестування; Економічна частина; Висновки; Публікації. _____

6. Консультанти розділів роботи (п. 6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п. 1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	доц. Табакова І.С.		30.05.2025
Економічна частина	ас. Легеза О.М.		27.05.2025

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної області і завдання на кваліфікаційну роботу	19.05.2025 – 20.05.2025	вик
2	Вибір програмного забезпечення	21.05.2025 – 22.05.2025	вик
3	Розробка практичної частини	23.05.2025 – 27.05.2025	вик
4	Економічна частина	24.05.2025 – 27.05.2025	вик
5	Оформлення пояснювальної записки	23.05.2025 – 27.05.2025	вик
6	Оформлення графічної частини	26.05.2025 – 28.05.2025	вик

Дата видачі завдання 19 травня 2025 р.

Здобувач



(підпис)

Керівник роботи



(підпис)

доц. Ірина ТАБАКОВА
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи містить: 57 с., 2 табл., 7 рис., 1 дод., 25 джерел.

ТРИВИМІРНЕ МОДЕЛЮВАННЯ, ІГРОВЕ СЕРЕДОВИЩЕ, UV-РОЗГОРТКА, ТЕКСТУРУВАННЯ, ІГРОВИЙ РУШІЙ.

Сучасна індустрія ігор стрімко розвивається, постійно з'являються нові технології і інструменти, завдяки яким студії створюють нові продукти, які дивують геймерів. Створення візуальної частини гри містить цілий ряд етапів, кожен з яких є важливим для того, щоб отримати не тільки красивий, але і відповідний правилам результат. Через те, що ігри можуть завантажуватись на різні пристрої і бути доступним якнайбільшій кількості гравців, тривимірні моделі повинні бути оптимізованими. Тому для того, щоб отримати середовище, яким зможуть насолодитись багато користувачів, потрібно розуміти і опрацювати все етапи його створення. В кваліфікаційній роботі розглянуте передове програмне забезпечення для моделювання, текстурингу і запікання, а також ігровий рушій. Окрема увага приділяється технологіям оптимізації сітки моделей, а також можливим дефектам і шляхам їх виправлення. Опрацьований пайплайн створення моделей і зборки ігрового середовища в ігровому рушії.

ABSTRACT

Explanatory note of the qualification work contains: 57 p., 2 tabl., 7 fig., 1 app., 25 sources.

THREE-DIMENSIONAL MODELING, GAME ENVIRONMENT,
UV-SCANNING, TEXTURING, GAME ENGINE

The modern game industry is developing dynamically, new technologies and tools are constantly emerging, thanks to which studios create new products that surprise gamers. The creation of the visual part of the game includes a number of stages, each of which is important in order to get not only a beautiful but also a result that complies with the rules. Since games can be downloaded to different devices and be accessible to as many players as possible, three-dimensional models must be optimized. Therefore, in order to get an environment that can be enjoyed by many users, you need to understand and work through all the stages of its creation. The qualifying work examines advanced modeling, texturing, and baking software, as well as the game engine. Special attention is paid to model mesh optimization technologies, as well as possible defects and ways to fix them. The pickup line for creating models and building the game environment in the game engine is worked out.

ЗМІСТ

	С.
СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	8
ВСТУП.....	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ	11
1.1 Розвиток і перспективи тривимірного моделювання	11
1.2 Загальні вимоги до ігрових 3D-моделей	13
1.3 Аналіз аналогів	14
1.4 Основні етапи створення 3D-середовища в game-індустрії.....	18
1.5 Основні методи моделювання	20
1.6 Оптимізація ігрових 3D-моделей	21
1.7 Текстурування 3D-моделей.....	23
1.8 Принципи PBR.....	24
1.9 Можливі дефекти і способи їх виправлення	28
2 ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	31
2.2 Програмне забезпечення для організації референсів	31
2.3 Програмне забезпечення для розробки 3D-моделі.....	32
2.4 Програмне забезпечення для підготовки графічних матеріалів, які будуть використовуватись у текстуруванні.....	33
2.5 Програмне забезпечення для текстурування	33
2.6 Програмне забезпечення для зборки ігрового середовища і тестування.....	34
3 РОЗРОБКА ПРАКТИЧНОЇ ЧАСТИНИ	35
3.1 Розробка концепту і організація референсів.....	35
3.2 Розробка низькополігональних 3D-моделей.....	36
3.3 Створення UV-розгортки	38
3.4 Розробка високополігональних 3D-моделей.....	40
3.5 Запікання процедурних карт.....	41

3.6 Розробка та додавання текстур на поверхню моделі.....	42
3.7 Збірка середовища в ігровому рушії і тестування	44
4 ЕКОНОМІЧНА ЧАСТИНА	47
ВИСНОВКИ	54
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	55
ДОДАТОК А Готове середовище	58

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

Полігон – плоска поверхня в тривимірному просторі, що побудована як мінімум трьома вершинами;

Low-poly – низькополігональна модель;

High-poly – високополігональна модель;

Hard edge – ребро 3D-моделі, яке чітко визначає кут або зміну напрямку поверхні, створюючи різкий, видимий перехід;

Soft edge – ребро 3D-моделі, яке визначає плавний, згладжений перехід між сусідніми гранями;

Пайплайн – послідовність етапів у процесі розробки;

Нормаль – вектор, перпендикулярний до поверхні в певній точці;

Пропс – 3D-модель, що використовується в ігровому середовищі;

Меш – сітка 3D-моделі, що складається з вершин, ребер та полігонів;

UV-острів – групі з'єднаних між собою UV-координат, які представляють собою безперервну частину 3D-моделі на UV-мапі.

ВСТУП

Ігрова індустрія є динамічною і перспективною сферою, що постійно еволюціонує, залучаючи мільйони гравців по всьому світу. Кожен рік з'являються нові ігрові всесвіти, відбуваються технологічні прориви. Особливо помітною стає роль інді-розробників, які виділяються своєю амбітністю і готовністю до постійного навчання і вдосконалення своїх навичок. Також ігри стають важливим соціальним і культурним феноменом, дуже часто в їх основу закладаються сюжети з літератури чи кінематографа, а при створенні рівнів часто надихаються реальними локаціями.

Метою роботи є проєктування якісних і оптимізованих моделей, що будуть не лише візуально привабливими, але й функціональними для інтерактивного середовища. Кожна модель буде розроблена з особливою увагою до деталізації та реалістичності текстур, щоб максимально занурити гравця у віртуальний світ.

Для досягнення мети необхідно виконати такі завдання:

- провести аналіз предметної області і завдання на кваліфікаційну роботу;
- провести аналіз аналогів;
- вибрати і обґрунтувати використання ігрового рушія, а також програмного забезпечення для моделювання, запікання, текстурування;
- зібрати і структурувати референси;
- розробити низькополігональні моделі;
- зробити UV-розгортку;
- розробити високополігональні моделі;
- провести запікання і текстурування;
- розташувати моделі у ігровому рушії;
- зробити рендери.

Актуальність роботи полягає в тому, що ігрова індустрія стрімко розвивається і сучасний гравець потребує все цікавіших і більш візуально привабливих ігор. Завдяки новим інструментам і технікам 3D-художники мають можливість автоматизувати свою роботу і зробити її більш якісною і акуратною. Саме тому важливо опанувати ці засоби і з їх допомогою створювати такі ігрові тривимірні середовища, що забезпечать незабутній ігровий досвід.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Розвиток і перспективи тривимірного моделювання

В основі сучасного тривимірного моделювання лежить поява перших комерційно доступних систем автоматизованого проектування (САПР) у 1960-х роках. Системою числового програмування, що пізніше стала САПР, займався вчений Патрик Ханратті.

Першою програмою для роботи з комп'ютерною графікою вважається Sketchpad. Хоч вона і не пропонувала 3D-моделювання в сучасному його розумінні, але дозволяла маніпулювати графічними об'єктами на екрані. Також Sketchpad мала перші графічні інтерфейси користувача.

У 1980-х роках великі автомобільні і аерокосмічні компанії почали використовувати системи автоматизованого проектування для роботи з кресленнями і каркасами об'єктів. Одною з таких систем була CATIA (Computer Aided Three Dimensional Interactive Application), яку використовували такі компанії, як Boeing та General Motors. З того часу програма постійно оновлювалась, а її сучасна версія доступна для придбання з новітніми технологіями і інструментами. Популярна програма AutoCAD, яку часто використовують промислові дизайнери і інженери, з'явилася у 1982 році, мала значний успіх і заклала основи для одного з найвідоміших програмних продуктів у галузі САПР.

В 1995 році була створена система твердотільного та поверхневого моделювання Solid Works. Вона поєднувала в собі можливості прямого моделювання (можливість безпосередньо маніпулювати геометрією моделі) та параметричного моделювання (контроль над розмірами, відношеннями та історією змін).

В 2013 році вийшла програма Autodesk Fusion 360, яка мала вбудовані можливості для моделювання, співпраці, симуляції, рендерингу, анімації тощо. Застосування хмарних технологій зробило програму більш гнучкою для використання великими командами. Також в 2013 році був випущений перший мобільний додаток для 3D-моделювання.

В 2015 році з'явився сервіс Onshape з повністю веб-орієнтованою архітектурою. Для його використання не треба завантажувати ніякі файли, а також можна працювати в команді, а автоматичне збереження мінімізувало ризик втрати своїх напрацювань.

Таким чином, історія програм для 3D-моделювання триває вже декілька десятиліть. Але сучасний стан цих технологій лише покращився [12]. Тривимірне представлення об'єктів використовується в таких сферах як: медицина, навчання, культура, архітектура, кінематограф, ігри тощо. Зараз існують безкоштовні програми для моделювання, що робить цю сферу доступною для новачків, а для тих, кому не потрібно сильно заглиблюватись в цю сферу, а потрібно лише отримати прості ілюстрації для додавання на сайт чи в пост в соціальних мережах, можуть скористатись онлайн-сервісами, які мають прості інтерфеси і лише потрібні інструменти.

З розвитком технологій VR і AR тривимірні моделі почали користуватись ще більшим попитом. Ці технології використовуються у навчанні, даючи можливість учням побачити анатомію людини, історичні події чи планети. Все це позитивно впливає на запам'ятовування. З використанням доповненої реальності ігри стають все більш цікавими, так як в людини з'являється можливість повністю «зануритись» в атмосферу. За використання VR-шлему разом з спеціальними контролерами в сучасних програмах можна малювати, готувати чи грати на музичних інструментах.

1.2 Загальні вимоги до ігрових 3D-моделей

Графіка є невід'ємною частиною будь-якої гри. Навіть ще не придбавши гру, користувач може зацікавитись скріншотами ігрового середовища, персонажей чи їх зброї. Очікування геймерів стають все вищими з кожним роком, так як індустрія стрімко розвивається. Саме тому для 3D-моделей важливий їх зовнішній вигляд, але при цьому іграм потрібно завантажуватися швидко на різноманітних пристроях, тож спеціалісти застосовують різні техніки для того, щоб велика кількість об'єктів, що при цьому виглядають реалістично, не навантажувала комп'ютери.

Ключовою складовою оптимізації ігрових моделей є зменшення кількості полігонів їх сітки шляхом видалення точок чи ребер, які не впливають на побудову форми об'єкта. Під час моделювання 3D-художник може спеціально додавати більше геометрії для того, щоб відворити складну форму чи видавити полігони. А якщо в кінці її прибрати, це не вплине на вигляд моделі, проте це позитивно вплине на продуктивність гри. Але через малу деталізацію об'єкти не виглядатимуть реалістично. Саме через це були створені карти, що імітують рельєф. Тепер ця технологія використовується в усіх сучасних іграх, так як моделі виглядають деталізовано, але при цьому вони є низькополігональними. Зараз існують карти, що імітують висоту, прозорість, дзеркальність тощо.

Окрім того, що кількість полігонів повинна бути зменшена, сітка також повинна мати правильну будову. Будь-яка ігрова модель складається з трикутників чи прямокутників, інакше можна отримати «артефакти» при експорті до інших програм, анімації чи рендерингу [16]. Саме тому важливим етапом побудови ігрового 3D-об'єкта є перевірка правильності її сітки. Сучасні програми для моделювання автоматизують цей процес і дають можливість проаналізувати і виправити сітку у лічені секунди. Але якщо модель має складну будову, можливо знадобиться додаткова робота вручну.

Ще одною простою, але важливою вимогою є однорідність вигляду створюваних моделей. Кожен з об'єктів відображає спільну історію і настрій сцени, тож всі вони мають доповнювати одне одного, а не виділятися серед інших. Для цього розписується концепція сцени, визначається іде і концепція розроблюваного середовища [1].

1.3 Аналіз аналогів

У сучасному світі індустрія 3D-ігор дуже швидко розвивається, виникають нові жанри та інструменти для їх створення. Є багато програм, що надають можливість розробити готовий продукт, а також безліч освітніх ресурсів, які допомагають охочим дізнатися всі теоретичні аспекти створення графіки, роботи з різними ігровими рушіями або тестуванням. Саме тому з'являються інді-розробники, які діляться своїми іграми на спеціальних платформах, і деякі з таких ігор стають популярними та обговорюваними серед геймерів. Одними з таких платформ є сайт itch.io, а також Steam, що дозволяють розробникам публікувати свої ігри та отримувати фідбек від інших користувачів.

Одним з таких прикладів є гра «No Pressure». Вона є студентським проектом, зробленим невеликою групою розробників, які створили вулиці міста Монреаль в Канаді, в якому персонаж може займатись паркурком, стрибаючи по будівлях (рис. 1.1). Стиль гри фотореалістичний, і можливо побачити, що команді вдалося передати архітектурні особливості міста. Але під час гри можливо помітити проблеми з оптимізацією моделей. Деякі текстури виглядають низькоякісними, що вказує на погану щільність UV-розгортки в тих моделях, які гравець може достатньо добре роздивитися. Під час створення ігрової графіки важливо коректно розподіляти кількість полігонів для різних об'єктів. Це означає, що моделі, що знаходяться на фоні, не доступні для сильного наближення камери, можуть бути менш деталізованими, але ті предмети, з якими персонаж взаємодіє, повинні бути

більш плавними, мати більше полігонів. Для реалістичності середовища в грі використовуються PBR-текстури, які пропонують додаткову інформацію та деталі. Серед них можуть бути: карти нормалей, металічності, шорсткості, висоти, прозорості тощо. Така технологія дуже часто трапляється в більшості сучасних тривимірних ігор і дозволяє відтворити ефект рельєфу не впливаючи на час завантаження контенту.



Рисунок 1.1 – Ігрове середовище «No Pressure»

Серією ігор, в яких середовище відіграє ключову роль, є «Euro Truck Simulator». Задачею гравця – мандрувати по різних містах на вантажній машині. Кожна з частин присвячена різним країнам, тому дослідження візуальної складової цієї гри є дійсно цікавим. Більшість локацій в грі схожі на реальні фотографії. 3D-художники, що брали участь в розробці, часто діляться результатами на мистецьких платформах, таких як Artstation, що дає можливість оцінити якість моделей. Спеціаліст, що займався розробкою розширення про балтійські держави поділився своїми результатами, на яких видно вулиці, які будуть проїжджати гравці (рис. 1.2). За цим фото добре видно, що будівлі, які оточені деревами, парканами, смітниками чи

лавочками, а також моделі, текстури яких відображають їх історію і вік, виглядають більш реалістично. Це пояснюється тим, що в реальному житті майже не можливо побачити ідеально чистий об'єкт на вулиці, адже навіть на нещодавно зведені споруди падає пил, дощ та бруд. Історію об'єктів можливо відобразити за допомогою різних інструментів в програмах для текстурування, одними з яких є генератори. Вони є засобами, що створюють реалістичні маски, завдяки яким якраз і можна отримати той самий ефект бруду чи стертої фарби на краях об'єкта. Окрім цього також існують фільтри, рівні, smart-матеріали та пензлі, якими 3D-художник може самостійно домалювати різні елементи.



Рисунок 1.2 – Ігрове середовище «Euro Truck Simulator 2»

У грі якісно виставлене світло, ми бачимо тіні від всіх предметів, це також додає реалістичності грі. Освітлення є одним з найважливіших факторів для досягнення фотореалістичності. Застосування технології динамічного глобального освітлення, трасування променів в реальному часі та освітлення за допомогою карт HDRI дуже часто використовуються в сучасних іграх для досягнення повного занурення в атмосферу. За

допомогою світла можна задати правильний настрій гри: м'яке, розсіяне освітлення створює атмосферу спокою та затишку, тоді як контрастне чи різке світло часто використовується в іграх жанру горор, в яких атмосфера повинна бути напруженою та страшною. Освітлення відіграє ключову роль не лише у створенні атмосфери, але і у керуванні увагою гравця. Завдяки світлу можна ефективно підкреслювати важливі об'єкти, наприклад, яскраво освітлювати ключові предмети або вказувати шлях до наступної локації. Це дозволяє ненав'язливо направляти гравця в потрібному напрямку, роблячи ігровий процес інтуїтивно зрозумілим.

Для оптимізації продуктивності та швидкості завантаження ігор часто використовується технологія запікання світла або *light baking*. Цей процес передбачає попереднє обчислення та збереження освітлення для статичних об'єктів (таких як стіни, елементи конструкцій, нерухомі меблі) у вигляді текстур. Хоча запечене освітлення не може динамічно реагувати на рух активних об'єктів (наприклад, гравця чи ворогів), воно суттєво знижує навантаження на систему, дозволяючи досягти високої візуальної якості без шкоди для продуктивності. Сучасна ігрова індустрія пропонує розробникам і художникам багатство технологій для створення фотореалістичних проєктів, дозволяючи їм обирати інструменти, що найкраще відповідають їхнім потребам. Для 3D-моделювання доступні як безкоштовні рішення на кшталт Blender, так і професійні інструменти, як Autodesk Maya [24]. Створення текстур часто здійснюється за допомогою Photoshop або Substance Painter.

Ключовим етапом розробки є розміщення всіх об'єктів в ігровому рушії. Unity та Unreal Engine надають для цього широкі можливості, включаючи додавання симуляцій, ігрових подій і складної взаємодії між моделями. Завдяки цим інструментам навіть незалежні розробники тепер мають доступ до інструментів для створення ігор, які не тільки візуально можна порівняти з якісними фільмами, але також ідеально оптимізовані для різних платформ.

1.4 Основні етапи створення 3D-середовища в game-індустрії

Перед початком роботи зазвичай формулюється ідея середовища. Це може бути письмовий опис історії цього місця, як ця сцена пов'язана з попередніми, яку має атмосферу тощо. Такий опис може допомогти у створенні скетчів і підборі референсів. Цей етап є обов'язковим при розробці будь-якої гри, так як спеціалістам потрібно уявляти вигляд сцени, об'єктів оточення, персонажів. При відтворенні реального місця буде достатньо фотографій, але при створенні фантастичних місць до роботи залучаються художники, які малюють концепт-арти [4]. Для цієї задачі зазвичай використовується програма Adobe Photoshop, яка пропонує широкий спектр інструментів для створення рисунків.

Далі починається етап моделювання, для якого кожний спеціаліст може обрати більш зручне для нього програмне забезпечення, так як зараз існує широкий спектр інструментів з різними підходами та функціоналом [8]. Найбільш популярними з них є Blender, 3Ds Max та Maya. На початку роботи 3D-художники одразу не створюють повноцінні деталізовані моделі – зазвичай спеціаліст починає з «блокіну», тобто розміщує прості фігури на місці майбутніх будівель і пропсів, підбираючи потрібну композицію і співвідношення розмірів. Після цього 3D-моделер переходить до створення низькополігональних моделей. В залежності від реалістичності гри, а також пристроїв, на яких планується можливість її відтворювати, об'єкти можуть бути більше чи менше деталізовані.

Для об'єктів будинків процес моделювання може відрізнитися. Так як зазвичай вони мають повторювані частини, то для того, щоб отримати готову модель, достатньо створити один такий фрагмент, зробити для нього текстури, а потім продублювати в ігровому рушії і зібрати, як конструктор. Такий спосіб не лише економить час 3D-художника, а й значно оптимізує використання пам'яті та підвищує продуктивність ігрового рушія, адже замість зберігання безлічі унікальних моделей, використовується лише один

базовий елемент. Крім того, цей підхід дозволяє збирати будинки різних розмірів, що дає можливість отримати цілі вулиці з використанням мінімальної кількості часу.

Для всіх моделей у грі робиться UV-розгортка, тобто тривимірний об'єкт розміщується на 2D-площині [9]. Цей етап потрібен для коректного відображення текстур на моделі. Після цього для об'єктів створюються їх високополігональні версії. При їх розробці можна не рахувати кількість утворених полігонів – ці моделі не будуть використовуватись у грі. Вони потрібні для того, щоб запекти всю інформацію про рельєф на низькополігональну модель, створюючи ілюзію деталізації. Основною такою картою є Normal map. Це кольорове зображення, в якому кожен канал відповідає компонентам X, Y і Z вектора нормалі поверхні [13]. На основі high-poly моделі також можуть створюватися такі карти, як: Ambient Occlusion, Thickness, Curvature, Position тощо.

Всі створені карти завантажуються в програму для текстурування, і за допомогою різних інструментів створюється фотореалістичний вигляд моделі. Програма Substance Painter пропонує широкий вибір пензлів, smart матеріалів, генераторів і масок для автоматизації роботи, тому в індустрії ігор вона є найбільш популярною для створення текстур. Вона також дозволяє робити експорт під різні рушії, такі як Unity чи Unreal Engine. Ця програма є популярною як серед тих художників, що працюють над фотореалістичними моделями, так і серед спеціалістів, що створюють стилізовані об'єкти, малюючи по поверхні пензлем.

Далі модель разом з матеріалом і текстурами імпортується в ігровий рушій. Саме на цьому етапі збирається вся сцена. За допомогою бібліотек, що пропонують сучасні програми для розробки ігор, створюється ландшафт. До середовища можна завантажити моделі трави, дерев чи каменів. Також на сцені розміщуються підготовлені будівлі і об'єкти середовища. Для більшої реалістичності в рушії додаються декалі, тобто текстури, які будуть накладені поверх підготовлених матеріалів для створення ще більш цікавого

зовнішнього вигляду об'єктів без навантаження на комп'ютер. Також одним з найбільш важливих елементів для досягнення фотореалістичності є світло. На сцені розміщуються елементи, які імітують сонячне світло, а також точкові і спрямовані джерела світла. Їх поєднання буде створювати тіні і підкреслювати рельєф та деталі всіх моделей на сцені.

Фінальним етапом проектування ігрового середовища є його комплексне тестування, спрямоване на виявлення дефектів, технічних проблем та оцінку відповідності запланованому ігровому досвіду. Цей етап включає візуальний огляд на наявність графічних артефактів, помилок геометрії чи некоректного відображення текстур, При відсутності таких проблем середовище додають до гри.

1.5 Основні методи моделювання

При створенні графіки для гри 3D-художник може обирати методи і способи створення моделей, в залежності від його навичок, програмного забезпечення і складності моделі. Існують два основних шляхи створення тривимірних об'єктів – коробкове моделювання і ліплення. Перша техніка означає створення бажаної моделі з примітивних фігур і поступове її ускладнення і вдосконалення [22]. Ліплення передбачає модифікацію сітки за допомогою пензлів [5]. Коробкове моделювання часто використовується для hard surface пропсів, а ліплення – для створення персонажів.

Також існує ряд менш популярних, але дієвих технік. При створенні будівель для пришвидшення і оптимізації роботи часто використовується спосіб модульного моделювання. Його ідея заключається в тому, щоб створити фрагменти будинку, а потім їх продублювати [14]. Для більшої гнучкості моделюються варіації цих елементів, які можна переставляти різними способами. Таким чином можна отримати ряд будівель, маючи лише набір базових фрагментів.

3D-моделі також можна створювати з NURBS-поверхонь. Основний контур 3D-форми визначається побудовою NURBS-кривих. Після розробки кривих їх можна використовувати для побудови поверхонь. Тобто в цьому способі поверхні моделі створюються шляхом "натягування" на каркас, який складається з кривих [21].

Зараз активно набирає популярність техніка фотосканування. Цей спосіб передбачає отримання фотографій об'єкта з різних ракурсів, після чого спеціальна програма їх аналізує і створює сітку моделі. Пропси, створені таким шляхом, доволі часто можна зустріти в сучасних іграх. Складність такого способу полягає в тому, що для отримання якісної моделі студії придбають спеціалізовані пристрої для зйомки. Також після генерації модель потребує доопрацювання і зменшення кількості полігонів.

1.6 Оптимізація ігрових 3D-моделей

Оптимізація моделей є процесом, який робить пайплайн створення ігрових середовищ настільки відмінним від етапів проектування об'єктів для кіно, мультфільмів чи візуалізації інтер'єру та продукції. Рендер ігрової сцени повинен відбуватись в реальному часі. Це впливає на час завантаження рівня, але дуже часто швидкість гравця дуже важлива [2]. Також гра має відриватись на пристроях з різними процесорами і відеокартами. Саме тому командам розробників ігор постійно приходиться працювати над оптимізацією моделей [10]. Існує ряд технік і інструментів, що дозволяють це робити легко і швидко.

При побудові моделі 3D-художник може спеціально додавати багато геометрії, щоб мати змогу робити нові деталі. Але якщо імпортувати таку модель у гру, вона може значно вплинути на її продуктивність. Через це потрібно видаляти всі непотрібні ребра. Зазвичай ними є ті, що не утворюють форму моделі. Це означає, що якщо їх прибрати, то вигляд моделі ніяк не зміниться. Але важливо пам'ятати, що після видалення ребер можуть

залишатись точки, які строюватимуть багатокутники, що є недопустимим у сітці ігрової моделі. Такі точки можна пришити до сусідніх.

Попри сильне зменшення кількості полігонів у ігрових моделях вони все одно повинні бути деталізованими і виглядати реалістично. Для цього використовується технологія запікання. Для цього створюються ще одна версія моделі, але вона в свою чергу може мати будь-яку кількість полігонів – цей об'єкт не буде використовуватись у самій грі. High-poly модель потрібна для того, щоб перенести всю інформацію про рельєф з неї на low-poly модель. Ця інформація буде відображатись завдяки картам нормалей.

Для коректного відображення текстур для low-poly моделі робиться UV-розгортка, яку теж можна оптимізувати. Під час розкладання тривимірної моделі потрібно підтримувати єдине значення щільності текселя для всіх UV-островів. Це термін, який використовується у комп'ютерній графіці для опису кількості деталей текстури на одиницю площі 3D поверхні [17]. Щільність текселю визначає, чи буде текстура виглядати однаково на всіх частинах об'єкту. 3D-моделери намагаються компоувати карту UV таким чином, щоб це значення було як можна більшим, так як в такому випадку текстура виглядатиме якіснішою. Але також існують загальноприйняті значення щільності текселя, які студії обирають в залежності від того, наскільки близько гравець зможе підійти до об'єктів. Таким чином для ігор, в яких ми бачимо світ від першої особи, найчастіше обирається значення приблизно 1200 р/м, а якщо від третьої – 600 р/м. Часто навіть приходиться перероблювати цей етап декілька разів, щоб знайти таку композицію, при якій площа UV карти буде найбільш щільно заповнена. Для того, щоб додатково звільнити місце, моделери також можуть застосовувати техніку оверлапу. Це означає, що UV-острови однакових елементів накладаються один на одного. Але при цьому варто заздалегідь розуміти, як виглядатиме текстура, бо при такому розміщенні UV-островів зображення буде повторюватись. Саме тому таку техніку доцільніше використовувати тільки для маленьких чи непомітних елементів.

1.7 Текстурування 3D-моделей

Створення фотореалістичної текстури потребує уважної кропіткої роботи з використанням різних інструментів. Програма Substance Painter пропонує 3D-моделерам широкий вибір засобів для автоматизованої роботи і створенні фотореалістичних моделей.

Одними із популярних засобів для створення реалістичної моделі є генератори. Вони створюють маску, яка може імітувати реалістичний пил, бруд, затерті ребра об'єкта тощо. Для натурального результату генератори беруть інформацію з процедурних карт, таким чином вони можуть аналізувати, де повинно бути більше пошкоджень чи бруду, а де менше. Іншим засобом для створення текстури є фільтри. Ці ефекти дуже схожі на ті, що пропонує програма Adobe Photoshop. Завдяки фільтрам ми можемо розмити шар, зробити його більш контрастним чи змінити його колір. Щоб результат роботи виглядав більш натурально, його вигляд треба зробити менш акуратним. Для цього в програмі є бібліотека пензлів, якими можна зтерти непотрібні фрагменти чи навпаки домалювати нові.

Для полегшення роботи в програмі пропонуються готові smart-матеріали. Це заздалегідь сформовані групи шарів, які містять основний матеріал, а також пошкодження, створені за допомогою генераторів і ефектів. Художник також може створити свій smart-матеріал і зберегти у своїй бібліотеці для подальшого використання у наступних проектах.

Окрім цього в програму можна завантажувати інші підготовлені матеріали. Наприклад, можна створити власний альфа-пензель, створивши чорно-біле зображення. Також можна завантажувати зображення, hdri-карти, маски, матеріали тощо.

Anchor Point – це технологія в програмі Substance Painter, яке виводить процес створення текстур на новий рівень. Ідея цього засобу заключається в тому, щоб зберегти створену маску, а потім використовувати її декілька разів. Таким чином 3D-художник може прискорити процес роботи, але при

цьому результат буде акуратним. Головне враховувати принцип роботи цього засоба – Anchor Point застосовується тільки для шарів, що знаходяться вище.

Ще одною корисною особливістю програми є підтримка UDIMs. Вони створюються для великих моделей з високою деталізацією, яким потрібно мати якісну текстуру. Програма Substance Painter дає можливість імпортувати такі моделі і працювати над їх текстурами.

Якщо потрібно показати проміжний етап роботи і отримати зображення моделі з текстурою, в програмі є вбудований рендер. Він дозволяє налаштувати камеру, фон, застосувати HDRI карти для кращого освітлення, додати фільтри тощо. Також Iray Renderer дозволяє експортувати зображення одразу до свого портфоліо на платформі ArtStation. Але важливо враховувати, що результат, отриманий в програмі Substance Painter та ігровому рушії Unreal Engine може значно відрізнятись, тож не варто спиратись на вигляд моделі в програмі для текстурвання, а краще перевірити одразу в рушії.

Для отримання рендерів програма також пропонує післяобробку. Для об'єктів можна провести кольорову корекцію, додати віньєтку, змінити тонове відображення, а також застосувати колірні профілі. Важливо розуміти, що ці ефекти не змінюють текстури, а лише впливають на візуальне відображення моделі в програмі.

1.8 Принципи PBR

Фізично коректний рендеринг (PBR), також відомий як фізично коректний шейдинг (PBS), – це метод затінення та рендерингу, який дає більш точне уявлення про те, як світло взаємодіє з властивостями матеріалу [24]. За допомогою цієї технології можна створювати фотореалістичні моделі, але для того, щоб об'єкти мали натуральний вигляд, існують певні правила налаштування PBR-матеріалів.

Існують два підходи для роботи з PBR: *metallic* і *specular workflow*. Вони є дуже схожими, але використовують різні фізичні методи обчислення текстур. Робочий процес *Metallic* пропонує кілька переваг, які роблять його популярним вибором для створення реалістичних матеріалів. Однією з таких є його простота та ефективність. Використовуючи робочий процес *Metallic*, можна досягти точних та візуально привабливих результатів з меншою кількістю текстурних карт порівняно з робочим процесом *Specular*. Застосування однієї карти *Metallic* дає змогу чітко розділити металеві та неметалеві ділянки, що полегшує створення матеріалів з різним ступенем відбиття. Крім того, робочий процес *Metallic* забезпечує кращий контроль над виглядом металів, дозволяючи створювати високо відбивні поверхні з діапазоном показників шорсткості. Однією з переваг робочого процесу *Specular* є його гнучкість, оскільки його можна застосовувати для широкого спектру матеріалів, враховуючи як металеві, так і неметалеві речовини. Він забезпечує детальний контроль над зовнішнім виглядом матеріалів, що дозволяє створювати високодеталізовані та нюансовані поверхні.

В пайплайні створення ігрової моделі робляться два набори мап: ті, що отримуються при запіканні і ті, що отримуються при текстуруванні. До перших відносяться: *Normal*, *World Space normal*, *ID*, *Ambient Occlusion*, *Curvature*, *Position*, *Thickness*. Кожна з них має своє практичне застосування:

- *Normal map* є кольоровою картою рельєфу, в якій кожен колірний канал відповідає компонентам X , Y і Z вектора нормалі поверхні на кожному текселі [7]. Вектори нормалей зберігаються відносно локальної системи координат кожного пікселя на поверхні моделі. Є ще один різновид цієї карти – *World Space normal map*, в ній вектори нормалей зберігаються відносно глобальної системи координат. *World Space Normal Map* має дуже нішеві застосування, переважно для отримання дуже специфічних ефектів або в певних рендерерах;

- *ID map* є допоміжною картою, за допомогою якої можна розділити частини моделі на групи, пофарбувавши їх в різні кольори. Вона

пришвидшує процес текстурування, так як спрощує процес нанесення матеріалів і текстур на підготовлені зони;

– Ambient Occlusion є чорно-білою текстурою, яка обчислює ослаблення та перекриття світла між деталями поверхні та іншими об'єктами в сцені, а потім додає реалістичне затінення до моделей;

– Curvature map є чорно-білою текстурою, яка відображає випуклість і вгнутість поверхні об'єкта. Дуже часто ця карта використовується для застосування генераторів, створюючи ефект стертих кутів і скупчення бруду;

– Position map зберігає інформацію про просторову позицію кожної точки на поверхні моделі у тривимірному просторі. Кожен канал цієї карти відповідає одній з осей у тривимірному просторі. Вона може застосовуватися у текстуруванні для відтворення таких ефектів, що залежать від абсолютного розташування об'єкта в просторі. Наприклад, за допомогою цієї карти можна відтворити ефект пилу, які знаходять лише зверху об'єкта;

– Thickness є специфічною чорно-білою картою, яка зберігає інформацію про товщину моделі. Зазвичай вона використовується для створення ефекта підповерхневого розсіювання.

Під час текстурування перелічені карти будуть використовуватись для відтворення ефектів чи прискорення роботи, але вони не будуть використовуватись у грі. Вийнятками є лише карти Normal map і Ambient Occlusion. Вони можуть бути зміненими під час текстурування, але після цього вони будуть імпортовані в ігровий рушій. Окрім них буде створений ряд нових карт:

– Base color map визначає кольоровість об'єкта. Для підходів Metallic і Specular застосування цієї карти може відрізнитись. Карта BaseColor містить як значення діелектричного матеріалу з карти Diffuse, так і значення металевого матеріалу з карти Specular. Карта Metallic визначає ці області. Зі збільшенням значення metallic значення BaseColor розділяється між внеском Diffuse та Specular;

– Roughness map визначає, наскільки поверхня шорстка або гладка. Вона визначає, як світло розподіляється по поверхні моделі. Гладка поверхня дає більш чіткі відблиски, а та, що має мікрорельєф, буде розсіювати світло в більшій кількості напрямків, що призведе до більш розмитих відблисків. Інвертованою картою для Roughness map є Glossiness map, яка використовується в Specular workflow;

– Metalness map визначає металічність об'єкта. Ця карта є показником того, наскільки сильно поверхня відбиває навколишнє середовище;

– Height map використовується для відтворення випуклих чи вогнутих деталей. На відміну від карт нормалей, ми можемо керувати значенням висоти, змінюючи значення від 0 до 1;

– Opacity map використовується для створення вікон, скляних стаканів, окулярів тощо, так як вона визначає прозорість. Ця карта є чорнобілою, в якій білий колір означає повну непрозорість, а чорний – непрозорість;

– Emissive Map є картою, що використовується для створення ефекта випромінення світла. Ця карта може визначати колір і інтенсивність свічення.

Таким чином, існує безліч карт, за допомогою яких можна створити фото реалістичний об'єкт. Але важливо враховувати, що для отримання натурального результату важливо дотримуватись рекомендованих значень для деяких з цих map.

При виборі кольору об'єкта важливо одразу визначити, чи він є металом чи неметалом. Для різновидів металів є заздалегідь розраховані правильні значення кольору. Наприклад, для відтворення матеріалу золото потрібно використовувати значення sRGB 255, 226, 155, а для алюмінія – 245, 246, 246 [3]. Також при текстуруванні металів враховується їх схильність до корозії. Іржа змінює відбивну здібність металу, тож ці ділянки обробляються як діелектричний матеріал. Діелектрики погано проводять електрику і відбивають набагато менше світла. Для більшості поширених неметалевих матеріалів значення відбиття знаходиться в діапазоні від 2% до 5%. Це дуже мале відбиття порівняно з металами. Для того, щоб об'єкти не

виглядали так, ніби вони тільки що були виготовлені на виробництві, а імітували взаємодію з навколишнім світом, до них застосовуються Roughness maps. Але для значення шорсткості теж має визначені межі. Для того, щоб об'єкт не виглядав дуже глянцево чи матово, потрібно вказувати значення від 0,2 до 0,8.

При застосуванні PBR-матеріалів варто пам'ятати, що вони будуть по-різному виглядати в програмі для текстурування і ігровому рушії. Тож не варто спиратись лише на результат, отриманий під час роботи в Substance Painter, потрібно також перевірити, який вигляд матиме модель в Unreal Engine. Причинами відмінностей можуть бути різні налаштування освітлення, колірний простір та алгоритми рендерингу.

1.9 Можливі дефекти і способи їх виправлення

Під час створення тривимірної моделі цілком можлива поява різноманітних дефектів чи артефактів. Вони можуть швидко виправлятися за допомогою інструментів, які пропонують сучасні програми для моделювання, але для цього важливо розуміти сутність цих проблем, а також можливі причини появи.

Одним з найбільш популярних дефектів, який може з'явитись через просту неуважність 3D-художника, є «нульовий фейс». Він може з'явитись у разі використання інструмента Extrude (Видавнення). Після його вибору створюється нова геометрія, яка повинна бути переміщена, але, якщо цього не зробити, вона залишиться поверх старої. Так само на одному місці можуть бути розташовані декілька точок, в такому випадку вони будуть утворювати багатокутники, яких не може бути в ігровій моделі. Наявність цих дефектів можна перевіряти за допомогою спеціальних інструментів в програмах для моделювання. Наприклад, в Autodesk Maya є інструмент Clean up, в якому можна обрати дефекти, наявність яких ми хочемо перевірити, а програма

підсвітить ті точки чи полігони, які є неправильними. Також цей інструмент може самотійно виправляти сітку.

Під час UV-розгортки можуть виникати дефекти, які суттєво впливатимуть на відображення текстур. При розкладанні вручну важливо уважно слідкувати за тим, щоб UV-острів не розтягувався чи змінював розмір. Для того, щоб всі UV-острова були правильними, варто використовувати інструменти Straighten, Unfold чи Optimize. Також важливо враховувати відступи між островами і від країв, так як при запіканні інформація виходитиме за їх межі. Якщо не врахувати ці відстані, то на моделі з'являться дефекти. Оптимальним значенням відступів між островами для карти розміром 4096x4096 є 32 пікселя, а від краю – 16 пікселів. Ще одним правилом, невиконання якого може призвести до появи дефектів, є наявність порізів на hard edge. Якщо не порізати UV-острів з нав'язаними в ньому жорсткими ребрами, то на карті вони відобразяться темними товстими лініями.

При запіканні процедурних карт також можуть виникнути дефекти. Один з найбільш поширених – поява чорних плям через неприбрані оверлапи в UV-розгортці. Оверлапи створюються для економії місця на UV-сеті і збільшення значення щільності текселя, але перед тим, як імпортувати модель в Marmoset Toolbag чи аналогічні програми, їх потрібно прибрати за межі сету. Також при створенні оперлапів важливо враховувати, чи буде на одну зі сторін об'єкта падати тінь від іншого об'єкта, так як при запікання вона відобразатиметься на обох його сторонах.

Під час текстурування може виникнути проблема, пов'язана з відображенням карти нормалей. Існують 2 види API (набори функцій), які використовуються у програмах для діалогу з графічним процесором. Тобто це 2 стандартних формата карт нормалей. Обидва мають однакові можливості, тільки їхній зелений канал (карта RGB) інвертований відносно одного. Якщо проєкт в програмі Substance Painter налаштований на Open GL, а карта, що завантажується – на Direct X, текстура може виглядати

неправильно. Для виправлення цієї проблеми можна інвертувати зелений канал прямо в програмі для текстуровання, або заздалегідь підготувати її за допомогою Adobe Photoshop.

При роботі з ігровим рушієм також можливі різноманітні дефекти. Дуже часто при налаштуванні матеріалу об'єкт може виглядати дуже глянцеvim. Зазвичай це пов'язано з неправильними налаштуванням самого матеріалу. При завантаженні набору текстур серед них обов'язково є кольорові, які зберігаються у sRGB, а є карти, які є масками або числовими значеннями від 0 до 1. Такими картами можуть бути Roughness, Metallic, Ambient Occlusion тощо. Освітлення також може викликати дефекти в Unreal Engine. Наприклад, світло може «протікати» через тонкі стіни або при динамічному освітленні можуть з'являтися «артефакти» у вигляді сходинок при неправильному налаштуванні.

2 ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Програмне забезпечення для опису концепції середовища

Для того, щоб зрозуміло описати ідею розроблюваної сцени, і отримати структуровану інформацію про моделі, композицію і загалом зрозуміти пайплайн створення ігрового середовища, вибрана програма, в якій дуже легко створювати абзаци тексту, списки, таблиці і додавати коментарі. Цією програмою є Microsoft Word. Цей редактор тексту має всі необхідні інструменти для того, щоб сформувати структурований і простий файл, який можна буде відправити іншим розробникам в художникам.

2.2 Програмне забезпечення для організації референсів

Для організації всіх референсів, скетчів і ідей зі створення моделей використовується програма PureRef. Вона має спеціальні функції, що зроблені саме для спеціалістів, що працюють з 3D-графікою. Програма має простий, але зрозумілий інтерфейс, вона дозволяє завантажувати безліч фотографій з різних джерел, групувати їх, додавати підписи. Найбільш корисною є можливість закріпити вікно поверх іншої програми, що дозволить бачити референси під час створення моделі. Це оптимізує роботу 3D-художника, так як важливо бачити деталі об'єкта з різних сторін для створення точної моделі. Ця програма є дуже популярною серед 3D-художників, так як дуже часто під час роботи потрібно бачити об'єкт з різних боків, а якщо відкривати їх по черзі у вкладках комп'ютера, то можна швидко заплутатись. Ця програма використовується для того, щоб підготувати вигляд моделі з різних боків, згрупувати приближені зображення окремих деталей, а також додати референси, на основі яких можна буде створити текстури.

2.3 Програмне забезпечення для розробки 3D-моделі

Autodesk Maya – це стандартизована програма в індустрії ігор і кіно. Вона містить широкий функціонал для полігонального і сплайнового моделювання, скульптингу, ретопології, UV-розгортки, текстурювання і бейкінгу [20]. Також програма активно використовується для комп'ютерної анімації і візуальних ефектів.

Autodesk Maya використовувалася для роботи над всесвітньо відомими проєктами. Наприклад, для популярної серії ігор «The Last of Us» в програмі були створені моделі персонажів, їх анімація, а також об'єкти оточення. Також велику роль ця професійна програма для тривимірного моделювання відіграла у створенні персонажів і неймовірно деталізованих історичних елементів середовища для серії ігор «The Assassin's Creed». Програма Autodesk Maya підтримує технології захоплення руху, що дозволяє студіям задіювати реальних акторів для того, щоб досягнути реалістичної анімації героїв у грі чи кіно.

Для полегшення і прискорення роботи професійні студії часто використовують плагіни, які працюють разом з Maya. Одними з популярних є плагін Golaem Crowd – симуляція натовпів для створення атмосфери «живого» міста чи Shave and a Haircut – для створення реалістичних зачісок. Оптимізує процес створення ігор інтеграція між Maya і популярними ігровими рушіями, як Unity і Unreal Engine. Спеціалісти можуть працювати в двох програмах одночасно, так як всі моделі і сцени з Maya імпортуються в рушії і організуються в окремі теки, де можна легко знайти всі моделі, матеріали, текстури і анімації.

Таким чином, Autodesk Maya є потужним програмним забезпеченням для створення тривимірних ігор. За допомогою програми можна не тільки створювати якісні моделі персонажів і оточення, а і легко інтегрувати їх в ігрові рушії для подальшої роботи над ними.

2.4 Програмне забезпечення для підготовки графічних матеріалів, які будуть використовуватись у текстурованні

Перед тим, як перейти до етапу текстуровання, потрібно підготувати ряд файлів, які будуть застосовуватись під час текстуровання. Це можуть бути зображення, які використовуватимуться як колір об'єкта, але також дуже корисним буде підготовка чорно-білих зображень для створення масок чи альфа-пензлів. Потрібний функціонал для створення таких матеріалів пропонує програма Photoshop, яка підтримує інтеграцію з іншими програмами компанії Adobe, в тому числі і інструментом для текстрування Substance Painter.

2.5 Програмне забезпечення для текстуровання

Substance Painter – це професійна програма для текстурингу. Її можна використовувати для роботи як з фотореалістичними, так і стилізованими моделями. Програма має декілька ключових особливостей, які виділяють її серед інших. По-перше, вона має систему шарів. Розробники програми додали цю складову, так як бачили, як вона спрощує роботу 2D-художникам під час роботи в Adobe Photoshop. Моделери можуть додавати окремі шари, маскувати їх, об'єднувати в групи і на їх основі створювати шаблони. Таким чином можна створювати складні деталізовані текстури і змінювати окремі елементи на будь-якому етапі роботи. Другою особливістю є технології PBR-текстуровання. Вони активно використовуються для створення фотореалістичних і при цьому оптимізованих ігрових сцен. Ще однією корисною перевагою програми Substance Painter є велика бібліотека текстур, пензлів, декалей, фільтрів тощо. Тож програма дозволяє як працювати зі своїми ресурсами, попередньо імпортуючи їх, так і з матеріалами, які по замовчанню містяться в бібліотеці.

2.6 Програмне забезпечення для зборки ігрового середовища і тестування

Unreal Engine – це ігровий рушій, який використовується для створення більшості сучасних ігор. Серед них такі гіганти, як: «Fortnite», «Street Fighter V» і «Batman Arkham Knight». Рушій пропонує інструменти для створення високоякісної графіки, підтримує роботу з віртуальною реальністю, а також з різними платформами: ПК, смартфони, ігрові консолі. Однією з ключових переваг Unreal Engine, яка відрізняє його від багатьох інших рушіїв, є можливість створювати ігри та інтерактивні досвіди без глибоких знань кодування. Це реалізується завдяки Blueprints – інструменту візуального програмування [6]. Blueprints дозволяють розробникам, навіть тим, хто не має досвіду в C++, створювати складну логіку гри, взаємодії та функціонал, просто з'єднуючи візуальні вузли. Таким чином, базові техніки гри, а також прототипи можуть бути створені навіть художником або дизайнером, значно прискорюючи процес розробки та спрощуючи пайплайн створення ігор. Ця функція робить Unreal Engine привабливим як для великих студій, так і для невеликих інді-команд.

3 РОЗРОБКА ПРАКТИЧНОЇ ЧАСТИНИ

3.1 Розробка концепту і організація референсів

Розробка ігрового середовища є процесом, в якому потрібно об'єднати візуальний вигляд сцени і його привабливість, створення логічної композиції, виставлення світла для коректного рендерингу тощо. Саме тому дуже важливим етапом в розробці є розробка концепту, який дасть бачення того, як буде виглядати середовище гри, які будуть співвідношення об'єктів і які кольори і матеріали використовуватимуться. Цей етап є основою для подальшої роботи.

Спочатку важливо письмово сформулювати концепцію середовища. Це може слугувати основою для визначення композиції, розміщення всіх об'єктів. Варто визначити, які моделі гравець зможе розгледіти краще, бо буде з ними комунікувати, а які об'єкти будуть лише декором. Це впливатиме на прийняття рішень при створенні текстур, так як елементи фону можна зробити більш одноманітними, а важливі моделі можуть мати дрібні деталі чи пасхалки у вигляді надписів і наліпок.

Є два способи, як можна представити початковий вигляд ігрового середовища – скетчі і референси. Рисунок сцени буде корисним в тій ситуації, коли середовище, яке буде створюватись, не існує. Але зазвичай середовища складаються з будівель і пропсів, які засновані на реальних об'єктах, тож для цього потрібні фото. Референси є зображеннями, які показують модель з різних сторін, відображають деталі для легшого перенесення в 3D-формат. Для розроблюваного середовища використовувались не тільки фотографії, а і об'ємні зображення вулиць з карт Google.

Для того, щоб зібрати всі рисунки і референси використовувалась програма PureRef. Вона є середовищем для організації і групування

графічних файлів і виведення їх на екран поверх інших вікон. Таким чином в програмі були сформовані такі групи: загальний план для визначення розташування об'єктів і загальної композиції, зображення об'єктів з різних сторін і їх деталі, а також зображення різних матеріалів, що використовуватимуться при текстуруванні.

3.2 Розробка низькополігональних 3D-моделей

Для середовища були підготовлені декілька груп пропсів і фрагментів будівель. В ході моделювання були використані і опрацьовані різноманітні техніки і інструменти тривимірного моделювання, які пропонує програма Autodesk Maya (рис. 3.1).

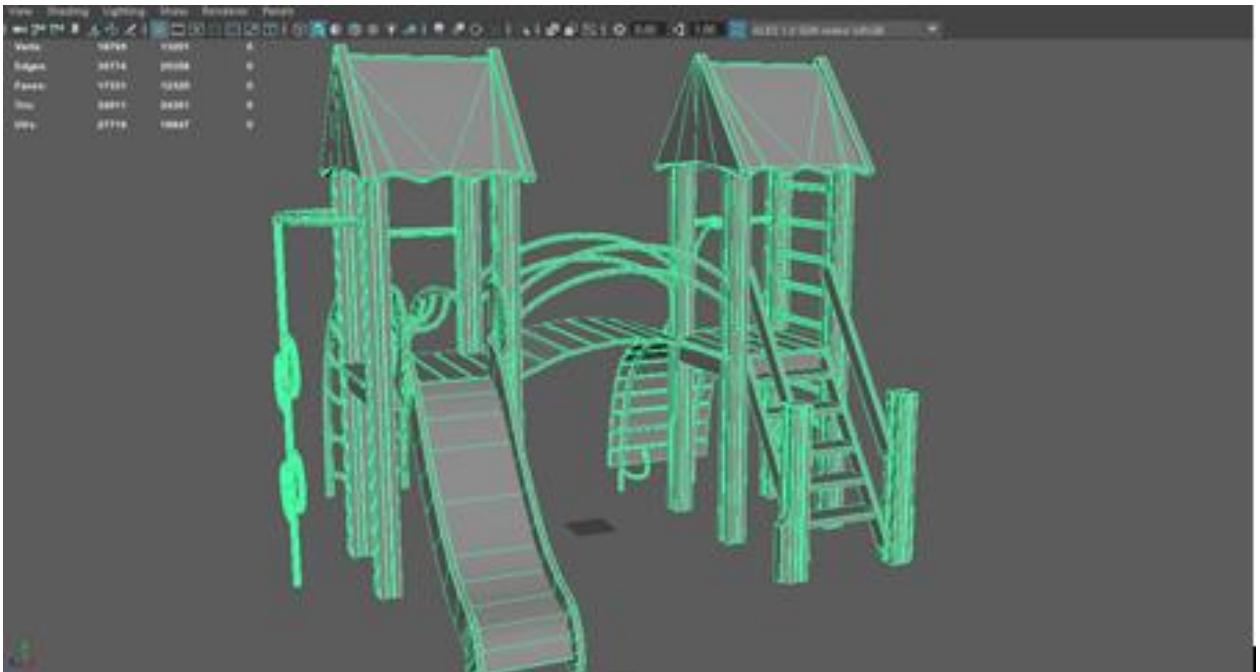


Рисунок 3.1 – Створена низькополігональна 3D-модель

Спочатку була створена модель ігрового майданчика. Для визначення габаритів об'єкта попередньо були знайдені референси з зазначеними розмірами. Першим етапом моделювання був блокінг, тобто створення форми майданчика за допомогою простих геометричних фігур. Після цього

почалась побудова низькополігональної моделі. Для цього була використана технологія полігонального моделювання, але для створення вигнутих пирил була застосована техніка Sweep mesh, тобто створення сітки навколо кривої. На дитячому майданчику також розміщений канат, який теж був створений шляхом протягування профіля за кривою за допомогою інструментів Sweep mesh, а для того, щоб він натурально лежав на кривій поверхні, для цього використовувалась технологія симуляції тканини. Канат був зазначений як тканина, а дерев'яні сходи під ним – як Passive Collider. Для того, щоб канат не впав під час симуляції, його крайні точки були прив'язані до мешу майданчика, таким чином був отриманий природній вигляд каната. Для перетворення тканини на звичайний меш для нього була видалена історія. Це зроблено для того, щоб при повторному ввімкненні симуляції об'єкт не рухався. Для створення форми гірки був використаний інструмент Loft, який будує сітку між декількома кривими.

Далі була створена інша група об'єктів, яка містила модель гаража, лавки, смітника і шини. Більшість моделей були створені на основі простих геометричних фігур. Для побудови пакета і сміття була застосована симуляція тканини. Для цього спочатку був створений плейн. Він був зазначений як лемент Cloth, а смітник – як Passive Collider. Після зміни налаштування тканини була ввімкнена симуляція і плейн впав таким чином, що розмістився всередині смітника. Для того, щоб пакет став мешем і симуляція не вмикалась знову, потрібно видалити історію об'єкта. Після цього були створені об'єкти сміття, але в налаштуваннях тканини їм була зазначена більша вага. Тепер смітник з пакетом були зазначені пасивними колайдерами. При ввімкненні симуляції об'єкти впали і натурально пом'ялись.

Була створена ще одна група моделей оточення – паркан і ліхтар. Ці об'єкти мають бути повторюваними в сцені, але для розроблюваного середиваща їх достатньо створити один раз, а потім продублювати в ігровому рушії за допомогою інструмента для створення масивів. Паркан був створений за допомогою простих геометричних фігур. В цілях оптимізації

моделі на місці сітки був залишений плейн – ефект сітки буде створений вже на етапі текстурування, використовуючи карту прозорості. Ліхтар також був створений шляхом полігонального моделювання з простих геометричних фігур, а для дротів спочатку були створені криві, а потім за допомогою інструмента Sweep mesh була отримана сітка.

Останні дві групи об'єктів – це фрагменти житлового будинку і школи. Для побудови таких великих об'єктів використовується технологія модульного моделювання. Вона заключається в тому, що спочатку визначається, які елементи будівлі будуть повторюватись, і вони моделюються один раз, щоб потім зібрати з цих елементів цілу модель шляхом їх дублювання. Для житлового будинку був створений фрагмент стіни з вікном двох розмірів, також стіни з вікном для сходового майданчика, двері під'їзду, два фрагменти стіни, штори, кондиціонер, два варіанта балкона і труба (рис. 3.2). Об'єкт школи буде розміщений на відстані від гравця, тож для нього створено менше фрагментів, серед них: стіна з вікном, ворота, а також два фрагмента стіни. Також окремо були створені плейни для скла в вікнах.

Для того, щоб в ігровому рушії було зручно зібрати з цих фрагментів будівлі і всі вони щільно прилягали один до одного, для кожного з них був переміщений Pivot point на крайню ліву точку. Таким чином, при включенні прив'язки до точок в ігровому рушії можна буде легко виставити всі стіни рівно і без пробілів.

3.3 Створення UV-розгортки

Всі елементи були розбиті на групи таким чином, щоб при проведенні UV-розгортки на всіх картах була приблизно одна щільність. Якщо виявляється, що якась з карт має занадто багато вільного місця чи частина UV-островів не вміщається, потрібно перегрупувати об'єкти і зробити розгортку повторно. Таким чином можна зменшити кількість карт у проєкті.

На початку для кожної групи робиться автоматична розгортка. Вона добре допомагає з простими об'єктами, але гірше рокладає складні моделі. Всі елементи, з якими програма не справляється, потрібно розкласти самостійно. Для цього зручно застосувати створення UV-розгортки на основі виду камери. Таким чином ми отримаємо зображення цілого об'єкта в редакторі. Після чого потрібно виділити потрібні ребра і розрізати об'єкт. При розгортці важливо враховувати, що всі Hard edge повинні бути розрізані, інакше при запіканні на їх місці можуть з'явитись артефакти у вигляді чорних полос. Для того, щоб отримані UV-острови мали правильну форму, для них можна застосувати інструмент Unfold Tool і Optimize Tool. А для всіх кривих об'єктів, як пирила, дроти чи гірка, будуть отримані криві і вигнуті UV-острови. Це може значно ускладнити процес текстурування, так як всі накладені зображення будуть викривлюватись. Тому для них варто застосувати Straighten UV. Цей інструмент випрямить UV-острів.

При складанні також варто враховувати відстані від краю карти та між UV-островами. Це потрібно для того, щоб під час запікання інформація не вилізла за межі одного UV-острова і не перейшла на інший. В розроблюваному ігровому середовищі створюються карти розміром 4096 x 4096 пікселів, тому для них допустими відстанями є 16 пікселів від краю і 32 пікселя між двома UV-островами. Для того, щоб мати можливість розмістити всі UV-острови правильно, можна ввімкнути сітку в редакторі і обрати кількість вертикальних та горизонтальних ліній рівними розміру карти. В роботі була додана сітка 1024x1024, щоб було краще видно дрібні деталі. Таким чином, мінімальна кількість клітинок від края і між островами складала 4 і 8 відповідно.

Для того, щоб текстура відображалась однаково на всіх моделях різного розміру варто підтримувати однакове значення щільності текселя. Інколи допускається зменшення його значення, але тільки для тих елементів, що знаходяться у важко доступних для гравця місцях. Програма Autodesk Maya має інструменти, які назначають і розраховують це значення. (рис. 3.2).

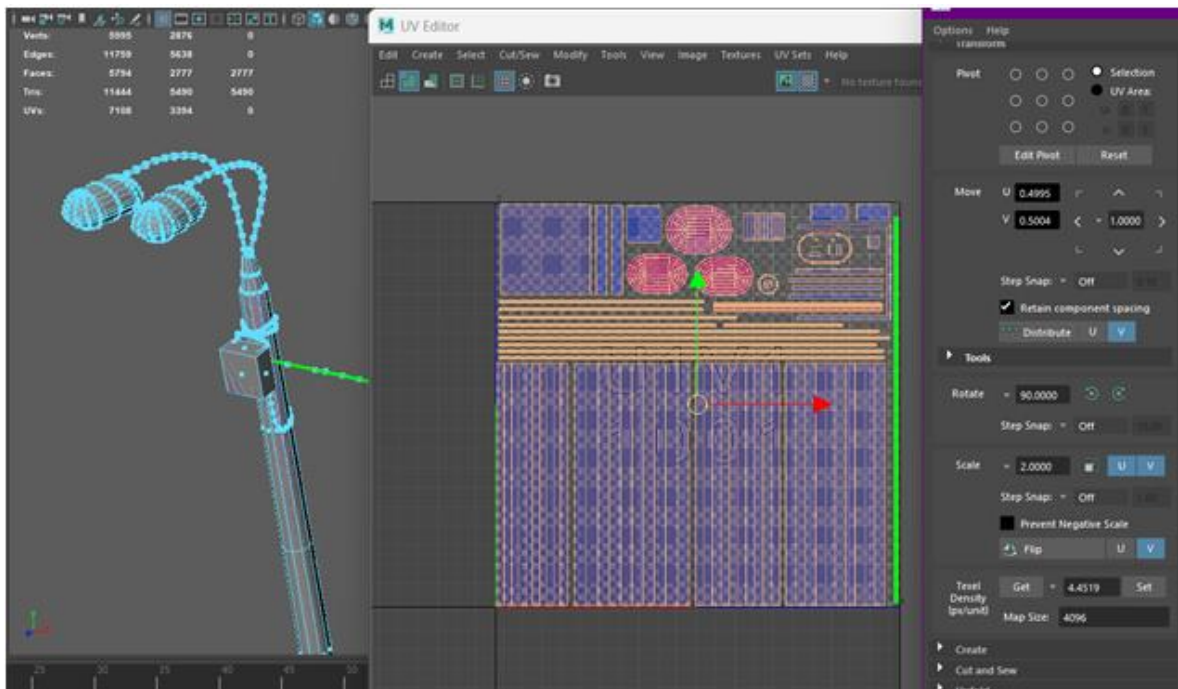


Рисунок 3.2 – UV-розгортка моделі

Для оптимізації UV-розгортки були зроблені оверлапи, тобто накладання однакових островів один поверх одного. Для коректного результату треба обов'язково перевіряти, чи немає неспівпадінь, інакше це вплине на відображення текстури. В роботі були застосовані оверлапи, але тільки для тих елементів, що майже не видні гравцю або мають маленький розмір. Ця техніка дозволяє звільнити місце на карті і збільшити значення щільності текселя. Але перед експортом моделі і запіканням потрібно прибрати всі оверлапи за межі карти, щоб уникнути артефактів при запіканні.

3.4 Розробка високополігональних 3D-моделей

Для того, щоб економити кількість полігонів у моделі і зменшувати навантаження на пристрої була придумана технологія, яка вплинула на пайплайн створення модеелй для всіх сучасних ігор. Ця технологія заключається в тому, що створюються дві версії об'єкта – низькополігональна і високополігональна. Далі інформація з high-poly

запікається на low-poly і ми отримуємо карти, які відтворюють рельєф і при цьому не змінюють геометрію моделі.

Таким чином, для всіх створених моделей робиться дублікат. Для зручності ці групи перейменовуються для того, щоб випадково не сплутати моделі. Навайжливіше у високополігональній моделі – це створення м'яких ребер, тобто використання інструмента Bevel. Так модель буде виглядати реалістичніше, бо матиме більше відблисків і тіней. Також на high-poly моделі додаються деталі на площині.

Для моделі шини було вирішено зробити її рельєф саме за допомогою запікання з високополігональної моделі. Для того, щоб відтворити цей патерн, спочатку був створений один його фрагмент, а потім з нього був зроблений лінійний масив. Для того, щоб отриманий патерн йшов по колу, як на шині, його потрібно деформувати. Для цього в програмі Autodesk Maya існує інструмент Bend. Він вигинає об'єкти, і якщо правильно налаштувати кути деформації, то можна отримати коло.

Для отримання високополігональних моделей бардюру був застосований скульптинг. Програма пропонує інструменти, за допомогою яких можна зробити натуральний рельєф побитого каменю. Таким чином із застосуванням пензля, який підіймає чи опускає обрану площину, був відтворений натуральний рельєф.

3.5 Запікання процедурних карт

Для того, щоб запекти процедурні карти, потрібно створити новий проект в програмі Marmoset Toolbag і завантажити в нього дві версії моделі. Далі потрібно створити проект запікання, який матиме окреме місце для додавання низькополігональної і високополігональної моделі. Після цього в налаштуваннях потрібно вказати розмір карт, шлях зберігання і вид тих карт, які потрібно отримати [15]. Для цієї роботи були вказані Normal map, Ambient Occlusion map, Curvature map. Після цього треба натиснути кнопку Запекти і

Перегляд. Тепер можна переглянути результат. Після першого разу цілком можливі артефакти, які можна прибрати прямо в програмі. Для цього були застосовані два інструменти – Skew та Offset. Інструментом Skew виправляємо ті місця, в яких тінь має неправильне направлення, а інструментом Offset зменшуємо розміри «клітки» яка покриває модель і визначає зону, яка має вплив на поверхню моделі при запіканні, прибираючи непотрібну інформацію. Після виправлення артефактів запікаємо знову.

3.6 Розробка та додавання текстур на поверхню моделі

Для текстурування в програмі Substance Painter Були завантажені низькополігональні версії моделей, а також отримані після запікання Normal map, Ambient Occlusion map і Curvature map.

Загальна ідея текстурування полягає в тому, щоб спочатку відтворити чисті і правильні матеріали об'єкта, а потім додати до нього елементи і ефекти, які відобразять його історію і вік [23]. Тож спочатку для всіх моделей були створені шари, які визначали їх кольоровість, металічність, шорсткість і прозорість. Після того, як моделі мали правильний і натуральний вигляд, почався етап створення ефекту старості. Для цього були застосовані генератори, які створювали маску на основі інформації зі заздалегідь завантажених процедурних карт. Але той результат, що створюють генератори, виглядає надто акуратно і послідовно. Тобто патерни маски настільки рівномірні, що вони не виглядають реалістичними. Для того, щоб це виправити, були створені шари для малювання, в яких до отриманого результату була додана ще форма, зроблена вручну. Це зробило маску пилу та бруду більш хаотичною, щоб досягнути результату, який ми можемо побачити в реальному житті.

На моделі дитячого майданчика були розміщені надписи і наліпки. Ці текстури спочатку розроблялись в Adobe Photoshop, а потім імпортувались в Substance Painter. Для того, щоб вони також виглядали натурально, були

додані потертості пензлями. Окремо в програму були імпортовані чорно-білі зображення слідів від взуття, для того щоб використати їх як альфа-пензель і постирати ними піл. Це створило імітацію того, що по моделі майданчика ходили ногами люди (рис 3.3).

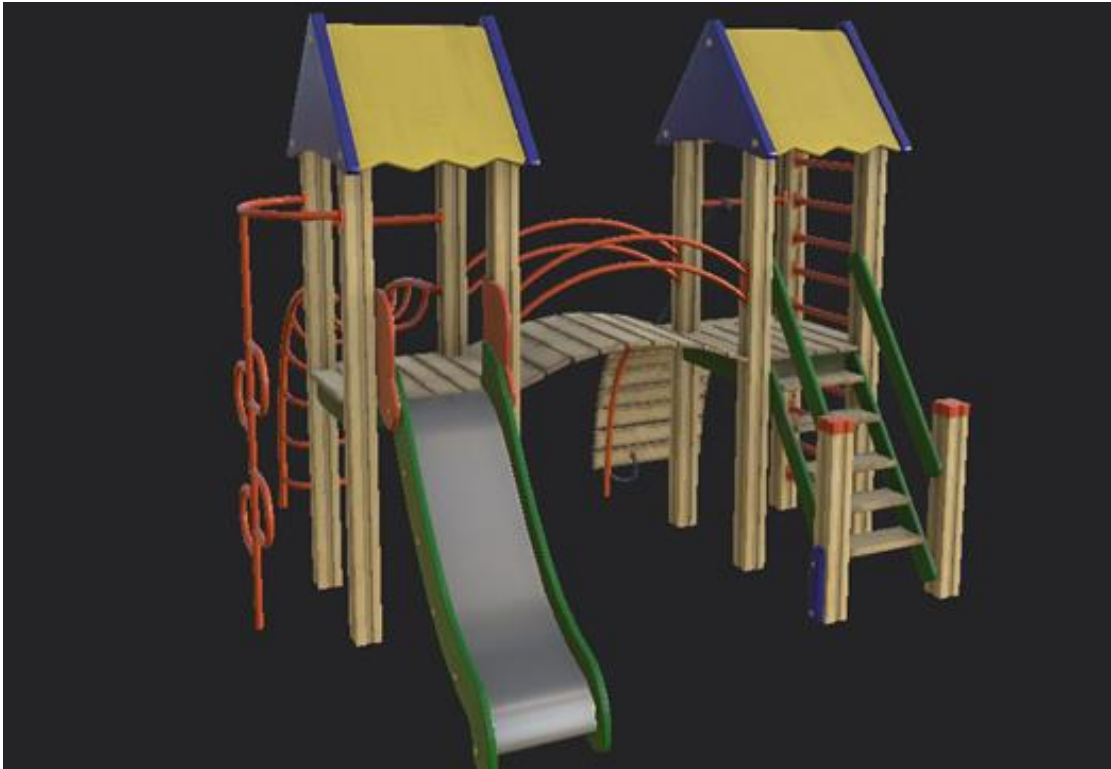


Рисунок 3.3 – Модель з текстурою

Для моделі ліхтаря був створений ефект свічення за допомогою властивості Emissive, тобто свічення. Був обраний колір світла, інтенсивність і за допомогою пензля намальована область, яка повинна світитись. Цієї однієї карти не вистачить, щоб відтворити ефект свічення ліхтаря, але в команді з джерелом світла, яке можна додати в ігровому рушії, можна добитись потрібного результату.

Після закінчення роботи всі текстури були експортовані. Карти Ambient Occlusion, Roughness, Metallic були об'єднані в одне зображення, в якому кожен канал відповідає одній з карт. Таким чином було зменшено кількість графічних файлів, які будуть завантажені до ігрового рушія.

3.7 Збірка середовища в ігровому рушії і тестування

Перед імпортом моделей і текстур були створенні окремі папки. Для зручного доступу вони мають зрозумілі назви. Після цього потрібно налаштувати матеріали об'єктів. Це відбувається за допомогою редактора матеріалів, що заснований на нодах.

Для того, щоб об'єкт з'явився на сцені, його достатньо перетягнути на неї. Всі моделі оточення були розташовані у середовищі за запланованою композицією, а моделі будинків були зібрані шляхом прив'язки Pivot Point моделі до точок інших. Таким чином були вибудовані цілі поверхи, що не мають дірок, всі фрагменти співпадають по ширині і висоті. Так як через вікна гравець має змогу досить добре бачити, за склом були додані плейни з текстурами кімнат, щоб створити ілюзію того, що за вікном знаходиться квартира (рис. 3.4).



Рисунок 3.4 – Вікна будинків

Для більш натурального ефекту були використані інструменти рушія для створення ландшафту. Інструмент «Ландшафт» має три режими: Керування, Ліплення та Малювання. За допомогою спеціальних пензлів був доданий

рельєф до землі, а також була згенерована трава. Окрім цього були завантажені моделі дерев і кущів, які пропонує бібліотека Fab від Epic Games. Після завантаження моделі ми отримуємо меш, а також Material Instance, який дозволяє змінювати деякі налаштування матеріалу.

Для того, щоб відтворити заплановану атмосферу сцени, було ретельно налаштовано освітлення з використанням кількох ключових типів джерел світла в Unreal Engine. На сцені присутнє джерело Sky Lights, що імітує освітлення від неба, та Directional Light, яке імітує Сонце. Для ефекту світла ліхтарів і лампочок застосовувались джерела Spot Light і Point Light.

Далі була проведена ретельна візуальна перевірка всіх моделей безпосередньо в режимі програвання, який має назву Play Mode (рис. 3.5). При виборі цього режиму вмикається симуляція проекту, в якій розробники і художники зможуть протестувати результат і за потреби швидко повернутись до режиму редагування.



Рисунок 3.5 – Тестування середовища

На цьому етапі важливо візуально оглянути всі моделі на наявність дефектів, які можуть виникнути при неправильній сітці чи під час текстурування. Також в режимі програвання було перевірено, чи може гравець пробігати між об'єктами. Особливу увагу було приділено таким проблемам, як розтягнуті текстури, нерівномірне накладання матеріалів, артефакти або "дірки" в геометрії, що можуть суттєво вплинути на візуальну якість та занурення гравця.

В результаті було отримане середовище, моделі якого мають правильну сітку та якісні текстури, що не мають дефектів. Після завершення роботи над середовищем були зроблені його рендери. Результат представлений на рисунках А.1 і А.2.

4 ЕКОНОМІЧНА ЧАСТИНА

У результаті виконання кваліфікаційної роботи створено тривимірне ігрове середовище. Впровадження якісного оптимізованого середовища у гру є ключовий етапом при розробці, оскільки він забезпечує захопливий досвід для гравців.

Економічна ефективність проекту розраховується перед проектуванням середовища, у результаті чого можливо спрогнозувати потенційний ефект і доцільність впровадження середовища у гру. Спочатку розраховується собівартість розробки, потім визначається ціна.

Розглянемо переваги проектного середовища.

Розроблюване середовище має якісні і оптимізовані моделі, що забезпечить можливість грати на різних пристроях. Також правильна сітка об'єктів мінімізує можливі баги чи проблеми з завантаженням рівня.

Середовище містить універсальні елементи будівель, які можна використовувати як для створення різних моделей будинків, так й імпортувати їх у інші проекти.

Збірка середовища проводиться у безкоштовному ігровому рушії, що значно зменшує загальну вартість проекту.

Отриманий проект в ігровому рушії має організовану структуру, що дасть можливість іншим спеціалістам продовжити роботу над ним.

Розглянемо конкурентне середовище відповідно до розроблюваної сцени. Аналогічні середовища можна зустріти на сайтах для інді-розробників, на яких вони поширюють свої ігри. Більшість з них мають неякісні або неоптимізовані моделі, а їх текстури зазвичай виглядають дуже просто і нереалістично. У зв'язку з цим у середовища, що проектується, передбачається явна перевага перед конкурентами з наступних причин:

– всі моделі оптимізовані відповідно до загальних правил побудови сітки ігрових моделей;

- всі текстури зроблені якісно і виглядають реалістично;
- забезпечено гнучку інтеграцію та розширюваність, що дозволяє легко адаптувати створене середовище під різні ігрові механіки та додавати нові елементи без значних технічних складнощів.

Розглянемо джерела економії, доходу, джерела фінансування.

Для фірми-розробника джерелом потенційного доходу є інтеграція середовища в ігрові проекти та подальша підтримка й розширення функціональності розробленого 3D-середовища. Витрати фірми-розробника на даному етапі включають витрати на моделювання, запікання, текстурування. Основним джерелом фінансування на початковому етапі є власні кошти фірми-розробника.

Для підприємства-замовника джерелом економії виступає заміна ручної праці машинною, що в значній мірі скорочує час. Витрати підприємства складаються з придбання потрібного ліцензованого програмного забезпечення і технічного обладнання.

Розглянемо порядок проектування ігрового тривимірного середовища.

У загальному випадку проектування ігрового тривимірного середовища містить у собі наступні етапи:

- етап формулювання ідеї і концепції середовища;
- збір і організації референсів;
- моделювання і UV-розгортка;
- запікання процедурних карт;
- створення текстур;
- збірка середовища у ігровому рушії.

Здійснимо розрахунок собівартості і ціни розробки веб-сайту.

У собівартість розробки веб-сайту входять наступні статті витрат:

- основна заробітна плата;
- додаткова заробітна плата;
- єдиний соціальний внесок;
- інші витрати.

Розробкою 3D-середовища займаються два фахівці: 3D-моделер і художник ігрового середовища. Зарплата 3D-моделера становить 370,00 грн/год, художника ігрового середовища – 430,00 грн/год. При цьому тривалість робочого дня кожного з них становить 8 годин. Середовище розроблялось 10 днів.

Розрахунок основної заробітної плати наведено у таблиці 4.1.

Додаткова заробітна плата – це винагорода за працю понад установлені норми, за трудові успіхи та винахідливість і за особливі умови праці. Вона включає доплати, надбавки, гарантійні та компенсаційні виплати, передбачені чинним законодавством; премії, пов'язані з виконанням виробничих завдань і функцій. У даному випадку додаткова заробітна плата становить 20 % від основної:

$$31280,00 * 0,2 = 6256,00 \text{ грн.}$$

Ставка єдиного соціального внеску становить 22 % від величини основної і додаткової заробітної плати:

$$(31280,00 + 6256,00) * 0,22 = 8257,92 \text{ грн.}$$

Таблиця 4.1– Розрахунок витрат на заробітну плату

Етап	Вид робіт	Виконавець		Годинна ставка, грн	Тривалість виконання, дні	Заробітна плата, грн
		кількість, ос.	посада			
1	2	3	4	5	6	7
1. Формулювання ідеї і концепції середовища	Заповнення письмового брифу	1	3D-моделер	370,00	0,5	1480,00
2. Збір і організації референсів	Завантаження референсів і їх організація у спеціалізованій програмі	1	3D-моделер	370,00	1,5	4440,00

Продовження таблиці 4.1

1	2	3	4	5	6	7
3. Моделювання і UV-розгортка	Моделювання low-poly і high-poly моделей та UV-розгортка	1	3D-моделер	370,00	2	5920,00
4. Запікання процедурних карт	Перенесення деталізації з високополігональної моделі на низькополігональну за допомогою технології запікання	11	3D-моделер	370,00	0,5	1480,00
5. Створення текстур	Надання поверхням візуальних властивостей	1	3D-моделер	370,00	2	5920,00
6. Збірка середовища у ігровому рушії	Імпортування моделей і текстур, налаштування матеріалів, створення композиції, налаштування світла	1	Художник ігрового середовища	430,00	3,5	12040,00
Разом					10	31280,00
Додаткова заробітна плата (20 %)						6256,00
Усього						37536,00

До інших витрат слід віднести витрати на обслуговування комп'ютерної техніки, що використовується виконавцями проєкту, плату за електроенергію і плату за програмне забезпечення.

Витрати на електроенергію розраховуються виходячи зі споживаної потужності пристрою і тарифу на електроенергію. У даному випадку передбачається використання двох комп'ютерів з потужністю 0,8 кВт/год. Вартість 1 кВт/год електроенергії прийнято у розмірі 4,32 грн. Час використання електроенергії в процесі розробки:

$$8 \times 10 = 80 \text{ год.}$$

Отже, плата за електроенергію складе:

$$0,8 \times 4,32 \times 80 \times 2 = 552,96 \text{ грн.}$$

Витрати на обслуговування техніки визначаються виходячи з її вартості та часу експлуатації, після закінчення якого, вона підлягає заміні (зазвичай цей час не перевищує 3-х років). Отже, враховуючи, що вартість кожного комп'ютера дорівнює 40000,00 грн, а протягом року техніка використовується 254 робочих дні, отримаємо наступну суму витрат на обслуговування за час виконання проєкту:

$$(80000,00 / (3 \times 8 \times 254)) \times 80 = 1049,87 \text{ грн.}$$

Витрати на програмне забезпечення визначаються виходячи з його вартості та часу використання. Отже, вартість ліцензії на програму для моделювання Autodesk Maya складає 82812,00 грн/рік, програму для текстурування Substance Painter – 515,00 грн/місяць, програму для запікання складає 53518,80 грн/рік.

Таким чином отримуємо вартість використання програми Autodesk Maya за 2 дні:

$$82812,00 / 254 \times 2 = 652,06 \text{ грн.}$$

Вартість використання програми Marmoset Toolbag за 0,5 дня:

$$53518,80 / 254 / 0,5 = 421,41 \text{ грн.}$$

Вартість використання програми Substance Painter за 2 дні:

$$515,00 \times 12 / 254 / 2 = 12,17 \text{ грн.}$$

Розрахуємо загальні витрати на платне ліцензійне програмне забезпечення для 10-денного проєкту:

$$652,06 + 421,41 + 12,17 = 1085,64 \text{ грн.}$$

Проєкт впроваджується для однієї компанії, тому собівартість розробки буде становити:

$$(37536,00 + 8257,92 + 1049,87 + 552,96 + 1085,64) / 1 = 48482,39 \text{ грн.}$$

Розрахуємо суму прибутку від реалізації розробки (виходячи з рівня рентабельності 30 %):

$$48482,39 \times 0,3 = 14544,72 \text{ грн.}$$

Розрахуємо ціну проєктування середовища без податку на додану вартість (ПДВ):

$$48482,39 + 14544,72 = 63027,11 \text{ грн.}$$

Розрахуємо суму ПДВ, що дорівнює 20 % від ціни без ПДВ:

$$63027,11 \times 0,2 = 12605,42 \text{ грн.}$$

З урахуванням проведених розрахунків ціна розробки ігрового середовища з ПДВ складає:

$$63027,11 + 12605,42 = 75632,53 \text{ грн.}$$

Результати розрахунків наведено у таблиці 4.2.

Таблиця 4.2 – Розрахунок витрат на розробку та ціни веб-сайту

№ з/п	Стаття витрат	Сума, грн
1	Основна заробітна плата	37536,00
2	Додаткова заробітна плата	6256,00
3	Єдиний соціальний внесок	8257,92
4	Витрати на обслуговування техніки	1049,87
5	Витрати на електроенергію	552,96
6	Витрати на програмне забезпечення	1085,64
7	Собівартість 3D-середовища	48482,39
8	Прибуток	14544,72
9	Ціна без ПДВ	63027,11
10	Податок на додану вартість (ПДВ)	12605,42
11	Ціна з урахуванням ПДВ	75632,53

Таким чином, повна вартість проектування 3D-середовища складе 75632,53 грн. Термін виконання усіх етапів проектування становить 10 днів для команди, до якої входять 3D-моделер та художник ігрового середовища. Очікувана сума прибутку складе 14544,72 грн, що свідчить про доцільність використання середовища у грі.

ВИСНОВКИ

У кваліфікаційній роботі були проаналізовані візуальні складові тривимірних інді-ігор, сформована концепція середовища, зібрані референси, спроектовані тривимірні моделі, зроблена UV-розгортка, запечені процедурі карти, зроблені текстури, зібране і протестоване середовище в ігровому рушії. Зроблені тривимірні моделі мають правильно і оптимізовану сітку і текстури, все ресурси в ігровому рушії структуровані, що робить середовище можливим для подальшої роботи для створення гри. Усі поставлені задачі були виконані, а саме:

- проведено аналіз предметної області і завдання на кваліфікаційну роботу;

- проведено аналіз аналогів;

- вибрано і обґрунтовано використання ігрового рушія, а також програмного забезпечення для моделювання, запікання, текстурування;

- зібрано і структурувано референси;

- розроблено низькополігональні моделі;

- зроблено UV-розгортку;

- розроблено високополігональні моделі;

- проведено запікання і текстурування;

- розташовано моделі у ігровому рушії;

- зроблено рендери.

Готове середовище має коректно побудовані моделі та якісні текстури. Отриманий результат перевірений на наявність артефактів і проблем, тому готовий до подальшого перенесення в гру для налаштування ігрових механік.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. 3D game art pipeline for beginners and professionals. Pixune. URL: <https://pixune.com/blog/3d-game-art-pipeline/> (date of access: 26.05.2025).
2. 3D model optimization: everything you need to know – 3d-ace. Studio. URL: <https://3d-ace.com/blog/3d-model-optimization/> (date of access: 26.05.2025).
3. McDermott W. The PBR guide: навчальний посібник. 3rd ed. Clermont-Ferrand, 2018. 104 p.
4. McGowan D. The stages of environment art in gamedev. 80 Level. URL: <https://80.lv/articles/the-stages-of-environment-art-in-gamedev> (date of access: 26.05.2025).
5. ThePro3DStudio. Overview on various types of 3D modeling techniques. ThePro3DStudio. URL: <https://professional3dservices.com/blog/3d-modeling-techniques.html> (date of access: 26.05.2025).
6. Козловець С. та ін. Аналіз можливостей створення ігрового контенту за допомогою Unreal Engine 5 // Радіоелектроніка та молодь у XXI столітті. 2023. С. 102-103.
7. Биковський Г.А., Кононенко М.Ю., Ібулаєв В.В. Моделювання game-ready пропсів у геймдизайні: технології та методи // Радіоелектроніка та молодь у XXI столітті. 2025. Т. 3. С. 116-118.
8. Дєдх Т. Сучасні засоби 3d-моделювання об'єктів. Сучасні інформаційні технології в освіті та науці // Матеріали доп. VII Всеукр. науково-практ. конф. з міжнар. участю. 2022. С. 59-60.
9. Драчко Є.О. Аналіз засобів створення 3D моделей для ігор // Радіоелектроніка та молодь у XXI столітті. 2023. С. 88-89.
10. Дробот А.П., Сергейко О.Ю. Розробка 3D ассетів // Радіоелектроніка та молодь у XXI столітті. 2025. С. 156-158.
11. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. Київ, 2016. 31 с.

12. Колгатіна Л., Першина О. Науково-дослідна робота студентів як чинник удосконалення професійної підготовки майбутнього вчителя // Огляд графічних редакторів для створення 3d об'єктів. 2020. № 19. С. 62-63.

13. Манскова Ю., Кулішова Н. Використання фізично коректного рендерингу (pbr) під час створення фотореалістичної 3d моделі для комп'ютерних ігор // Поліграфічні, мультимедійні та web-технології. 2024. Т. 2. С. 166-168.

14. Манскова Ю., Табакова І. Етапи створення ігрового 3d середовища // Поліграфічні, мультимедійні та web-технології. 2025. Т. 2. С. 241-242.

15. Манченко А.В., Солодов В.Д. Оптимізація 3d моделей для ігрових рушіїв // Радіоелектроніка та молодь у ХХІ столітті. 2024. С. 331-333.

16. Манченко А.В. Процес запікання процедурних карт для ігрових 3D моделей та використання програми Marmoset // Радіоелектроніка та молодь у ХХІ столітті. 2025. С. 237-239.

17. Маслік Д.Є., Ібулаєв В. «Texel density» в індустрії комп'ютерних відеоігор // Радіоелектроніка та молодь у ХХІ столітті. 2025. С. 240-243.

18. Методичні вказівки з виконання кваліфікаційної роботи для студентів денної та заочної форми навчання першого (бакалаврського) рівня вищої освіти спеціальності 186 "Видавництво та поліграфія" за освітньою програмою "Видавничо-поліграфічна справа" / В.П. Ткаченко, А.В. Бізюк, О.В. Вовк, І.М. Єгорова, В.Ф. Челомбійко. Харків: ХНУРЕ, 2020. 68 с.

19. Полозова Т.В. Методичні вказівки до виконання економічної частини кваліфікаційної роботи для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 186 Видавництво та поліграфія усіх форм навчання. Харків: ХНУРЕ, 2022. 47 с.

20. Пронін О., Колісник В. Використання оптимального графічного програмного забезпечення при створенні відеоігор // Радіоелектроніка та молодь у ХХІ столітті. 2024. С. 268-269.

21. Табакова І., Бедрата Р., Русінов Ю. Створення 3D-моделей для ігор // Поліграфічні, мультимедійні та web-технології. 2024. Т. 2. С. 120-123.

22. Табакова І., Кучеренко Д. Особливості побудови 3D моделей // Поліграфічні, мультимедійні та web-технології. 2024. Т. 2. С. 34-36.

23. Станіславенко Є. та ін. Текстурування при формуванні тривимірних персонажів // Інформаційні технології і автоматизація. 2023. С. 449-450.

24. Шеїн С. Порівняння пакетів Blender та Maya // Радіоелектроніка та молодь у XXI столітті. 2024. С. 237-238.

25. Що таке PBR (фізично коректний рендеринг)? – Adobe Substance 3D. URL: <https://www.adobe.com/ua/products/substance3d/discover/pbr.html> (дата звернення: 26.05.2025).