

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)
«23» травня 2022 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

студентові _____ *Мельніченку Дмитру Олександровичу* _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ *Розробка технології створення тривимірних моделей для відеоігор* _____

Затверджена наказом по університету від _____ 21.05.2022 № 558 Ст _____


2. Термін подання студентом роботи до екзаменаційної комісії _____ 20 червня 2022 р. _____

3. Вихідні дані до роботи
документація розробки для ігрового движку Unreal Engine 5, документація до Autodesk 3DsMax, документація до Substance Painter, формати файлів: для імпорту 3D-моделі у Substance Painter – *.fbx, Unreal Engine 5 – *.fbx.

4. Перелік питань, що потрібно опрацювати в роботі
Вступ. 1. Аналіз предметної області та завдання до роботи. 2 Аналіз існуючих засобів моделювання. 3 Аналіз аналогів. 4 Пошук референсів для 3D-моделі. 5 Визначення програмного забезпечення. 6 Розробка 3D-моделі. 7 Тестування коректності роботи 3D-моделі у ігровому движку. 8 Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п. 5 включається до завдання за рішенням випускової кафедри)
Вступ. 1 Актуальність розробки 3D-моделі для AAA-відеоігор. 2 Мета роботи та аналіз технічного завдання. 3 Аналіз предметної області. 4 Аналіз аналогів. 5 Основні вимоги до 3D-моделей. 6 Вибір та обґрунтування програмного забезпечення. 7 Виконання практичної частини. 8 Розробка текстур. 9 Тестування 3D-моделі. 10 Економічна частина. 11 Висновки.


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	ст. викл. Парамонов А.К.		20.06.2022
Економічна частина	проф. Полозова Т.В.		17.06.2022


КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання	16.05.2022	виконано
2	Аналіз предметної області та завдання до роботи	17.05.2022	виконано
3	Аналіз існуючих способів моделювання	18.05.2022	виконано
4	Аналіз аналогів	20.05.2022	виконано
5	Визначення програмного забезпечення	29.05.2022	виконано
6	Повна розробка 3D-моделі	05.05.2022	виконано
7	Оформлення пояснювальної записки	06.06.2022	виконано
8	Попередній захист	17.06.2022	виконано
9	Захист перед ЕК	23.06.2022	виконано

Дата видачі завдання 23 травня 2022 р.

Студент 
(підпис)

Мельніченко Д.О.

Керівник роботи 
(підпис)

ст. викл. Парамонов А.К.
(посада прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка містить 69 стор., 49 рис., 3 табл., 3 дод., 15 джерел.

3D-МОДЕЛЬ, UV-РОЗГОРТКА, МОДЕЛЮВАННЯ, ПОЛІГОНИ, РЕНДЕРІНГ, РЕФЕРНЕС, ТЕКСТУРІНГ, ЗАПІКАННЯ, AAA-ВІДЕОІГРИ, NORMAL MAP, AMBIENT OCCLUSION, CURVATURE.

Об'єкт дослідження – технології розробки 3D-моделі для AAA-відеоігри. Мета роботи – розробка 3D-моделі для навколишнього середовища, з можливістю подальшого використання у AAA грі на пристроях з високими графічними характеристиками.

Області застосування – використання моделі у AAA іграх. Проведено аналіз аналогів існуючих ігор для консолей та комп'ютера, розглянуто особливості моделювання 3D-моделей у цих іграх. На основі зібраних референсів було створено low-poly, high-poly моделі, UV-розгортку, розроблені карти та був виконаний текстурінг. З high-poly моделі розроблено карти нормалей, карту тіней (ambient occlusion), криватура. Після текстурінгу було створено карти базового кольору, металік, рафнес.

ABSTRACT

The explanatory note has 69 p., 49 pic., 3 tabl., 3 app., 15 sources.

3D-MODEL, UV, MODELING, POLYGONS, RENDERING, REFERENCE, TEXTURING, BAKING, AAA VIDEO GAMES, NORMAL MAP, AMBIENT OCCLUSION, CURVATURE.

The object of research is the technology of developing 3D models for AAA video games. The purpose of the work is to develop a 3D model for the environment, with the possibility of using it in the future in an AAA video game on devices with high graphic characteristics.

Applications - the use of the model in AAA games. The analysis of analogues of games for consoles and computer is carried out, the features of modeling 3D models in these games are considered. Based on the collected references, Low-Poly, High-Poly models, UVs, baked maps and texturing were created.

Normal maps, shadow maps (ambient occlusion), curvature were developed with High - Poly models. After texturing, base color, metallic, roughness maps were created.

ЗМІСТ

	С.
СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	8
ВСТУП.....	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ	11
1.1 Загальні вимоги до 3D-моделей.....	11
1.2 Аналіз аналогів	14
1.3 Основні етапи створення 3D-моделей в Gamedev індустрії	16
1.4 Основні методи моделювання	17
1.5 Текстурування 3D-моделей.....	22
1.6 Основні вимоги при проектуванні 3D-моделі	25
2 ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	30
2.1 Програмне забезпечення для збору референсів.....	30
2.2 Програмне забезпечення для розробки 3D моделей.....	31
2.3 Програмне забезпечення для створення розгортки	36
2.4 Програмне забезпечення для створення Normal Map, Curvature, AO....	37
2.5 Програмне забезпечення для редагування та видалення артефактів на картах текстур.....	42
2.6 Програмне забезпечення для створення текстур	44
2.7 Програмне забезпечення для тестування моделі	46
3 РОЗРОБКА ТРИВИМІРНОЇ МОДЕЛІ.....	48
3.1 Пошук референсів	48
3.2 Розробка 3D моделі	49
3.3 Розробка Low – Poly моделі.....	50
3.4 Розробка розгортки	52
3.5 Розробка карт для текстурування	55
3.6 Розробка текстур моделі	57
4 ТЕСТУВАННЯ МОДЕЛІ В ІГРОВОМУ РУШІЮ	61
5 ЕКОНОМІЧНА ЧАСТИНА	63

ВИСНОВКИ	69
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	71
ДОДАТОК А Етапи розробки 3D-моделі для відеогри ...	Error! Bookmark not defined.
ДОДАТОК Б Аналоги моделей до AAA-відеоігор	Error! Bookmark not defined.
ДОДАТОК В Текстури до 3D моделі.....	Error! Bookmark not defined.

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

AAA-відеоігри – сучасні відеоігри з великим бюджетом на розробку, що розробляються великими студіями з сотнями або навіть тисячами працівників.

Курватура (Curvature) – текстура, яка дозволяє отримувати та зберігати увігнуту та опуклу інформацію, пов'язану з Mesh-сіткою.

Карта Roughness (Рафнес) – визначає мікронерівності поверхні, через це відбувається розсіювання світла.

Карта Metallic (Металік) – визначає фізичні властивості поверхні, метал або не метал.

Mesh (Меш) – сукупність ребер (Edge), вершин (Vertex), полігонів (Polygon), що створюють модель.

Alfa (Альфа) – текстура якою додається деталізація на меші або текстурі.

RMA текстура – текстура в котрій розміщено 3 текстури на кожний канал RGB.

Texel Density – щільність пікселів на розгортці.

PBR – фізичний коректний рендерінг. Метод відображення текстур у графіці.

Shell – окремий шмат розгортки об'єкту чи деталі.

Overlap (оверлап) – накладання однакових шелів розгортки.

ПК – персональний комп'ютер.

Артефакт – некоректне відображення карт та текстур.

ВСТУП

Світ сучасних комп'ютерних ігор останнім часом перетворився в велику індустрію. Для створення гри з реалістичною графікою, анімацією потребується багато часу, зусиль та коштів. Компанії можуть мати декілька тисяч працівників і крім того, ще існують фрілансери та аутсорсинг (outsource) компанії, котрі не враховуються в штат основної команди. Відеоігри стали настільки великими і складними, що у розробці беруть участь художники, програмісти, аніматори, сценаристи, режисери, дизайнери, маркетологи, актори, звукорежисери. Комп'ютерні ігри тримаються на сюжеті, графіці, якщо сюжет в грі відсутній або він присутній у вигляді квестів, то це – онлайн гра, де, наприклад, дві команди грають одна проти одної за окрему винагороду.

Програмісти працюють над тим, щоб графіка у грі була більш реалістична, вони пишуть код, створюють шейдера, щоб у ігровому рушії, текстурі, котрі створив художник відображались більш коректно та реалістично. Художники створюють об'єкти сцени, іконки, спеціальні позначки для гравців, меню гри, котрі гравець бачить на екрані під час гри.

На сьогодні індустрія AAA-відеоігор – один з найпопулярніших та наймасштабніших напрямів розробки. З кожним роком індустрія розвивається дуже стрімко. Причина в тому, що сьогодні існують багато якісних інструментів для створення привабливої реалістичної графіки, але це потребує більших ресурсів від консолей або персональних комп'ютерів (ПК). Також кожна компанія конкурує з іншими аби створити найкращу та цікавішу гру. Актуальність розробки полягає в тому, що сучасні ігри наближаються до реалістичної графіки, анімації тощо. Іноді складно зрозуміти в якому реальному чи віртуальному середовищі знаходиться гравець.

Об'єкт дослідження – технології розробки 3D-моделі для AAA-відеоігри.

Мета роботи – розробка 3D-моделі для навколишнього середовища, з можливістю подальшого використання у AAA грі на пристроях з високими графічними характеристиками.

Області застосування – використання моделі у AAA відеоіграх з високоякісною реалістичною графікою, 3D-анімаціях, презентаційних та рекламних роликах.

Розробка повинна бути достатньо привабливою та реалістичною, з використанням високоякісних текстур. Для досягнення даної мети необхідно вирішити такі завдання:

- а) провести аналіз аналогів;
- б) розробити концепт моделі для відеогри;
- в) обрати технологію моделювання;
- г) обрати програмне забезпечення для реалізації;
- д) розробити модель за допомогою обраного методу моделювання;
- є) виконати ретопологію моделі;
- ж) підібрати та розробити текстури для моделі;
- з) тестування моделі у рушій гри.

При розробці 3D-моделі, необхідно пам'ятати що гравець звертає увагу на оточення протягом всієї гри, реалістичність та привабливість моделей сприяє настрою та мотивації для гравців. За останні роки комп'ютерні ігри стали більш доступними для кінцевого користувача – ігри перестали розроблятися лише задля розваг. Спеціальні ігри можуть виступати в якості навчального матеріалу або для використання гравців в науково-дослідних цілях. Але слід зазначити, що сучасні користувачі стали більш вибагливі до комп'ютерної графіки самої гри, до оточення та самих ігрових 3D-моделей.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Перед тим, як розпочати основну роботу над тривимірним об'єктом, необхідно проаналізувати предметну область та провести аналіз існуючих аналогів. Слід зазначити, що важливою складовою будь-якої комп'ютерної гри є її візуальне представлення, користувач протягом гри постійно знаходиться у ігровому середовищі і контактує із предметами – об'єктами ігрової сцени.

Слід нагадати, що перші комп'ютерні ігри були дуже простими, недорогими продуктами, у яких могли грати окремі користувачі чи декілька людей. Графіка була мінімалістичною чи зовсім відсутня. Розвиток висококласних технічно складних консолей і всесвітньої павутини змінило все це, перетворивши відеоігри на складні розраховані на багато користувачів продукти, що включають високоякісну графіку, відео і музику [3].

До кінця 1990-х років такі компанії, як EA і Sony, випускали блокбастери, які мали охопити величезну аудиторію і принести серйозний прибуток. Саме тоді розробники ігор почали використати термін AAA – сучасні відеоігри з великим бюджетом на розробку, що розробляються великими студіями з сотнями і навіть тисячами працівників. Сьогодні цей термін можна застосувати до будь-якої гри, тому що ігри довго, складно та дорого робити. Їхня ідея полягала в тому, щоб створити передчуття, інтерес до гри, реальність всього, що відбувається навколо гравця.

1.1 Загальні вимоги до 3D-моделей

При створенні 3D-моделі для AAA – відеоігор для розробника відкриваються великі можливості для створення привабливого вигляду моделі. З сучасними ресурсами пристроїв ПК чи консолі, художники можуть

дозволити робити велику кількість полігонів, але витратити їх марно непотрібно, бо кожні зайві ребра, вершини – це додаткове обчислення для пристрою, що за собою приведе зниження частоти кадрів у грі. Тому навіть з сучасними інструментами та ресурсами необхідно слідкувати за оптимізацією але робити це так, щоб не впливало на загальний вигляд гри.

Також оптимізація реалізується за допомогою камери, все що гравець не бачить, тобто не потрапляє в поле зору камери – відсікається. Це стосується моделей, анімацій, світла, VFX та багато іншого. Професійною мовою це називається Frustum Culling [18].

Для створення моделей художнику необхідно підібрати програмне забезпечення для всіх етапів розробки. Вибір програм залежить від виду моделей, якщо створюється персонаж, то використовується скульптинг, якщо це робот, можна використовувати Cad-програми або класичне полігональне моделювання. Програми повинні забезпечувати максимальну швидкість та зручність при розробці. Ці фактори можуть значно вплинути на фінальний продукт, адже швидкість створення ігор – це один із найважливіших вимог в індустрії [9]. Фінальна модель повинна відповідати заданій ідеї гри, якщо це апокаліпсис, то моделі зазвичай виглядають старими та зруйнованими часом, все це додає атмосферу та унікальність грі.

Найголовнішим в моделі на етапі моделювання – це сітка мешу, вона складається з ребер, вершин та полігонів. Саме вона створює форму об'єкту.

Сітку моделі потрібно робити якомога рівною, із більш-менш полігонів, наближених до правильної геометрії, необхідно уникати довгих трикутників, особливо на округлих формах. В деяких місцях розробник сам триангулює меш за допомогою додаткових ребер. Робиться це для того, щоб запобігти довгих потяжок на меші при розробці карт та текстурованні (рис. 1.1).

Оскільки ідеально на сто відсотків розгорнути модель неможливо, необхідно користуватися текстурою – чекером, який допомагає виявити потяжки на розгортці. Чітких правил при створенні розгортки не існує, але

слід пам'ятати, що від якості розгортки залежать майбутні карти при бейку та текстурі. Рекомендується тримати розгортку в координатах від 0 до 1, якщо вийти за ці межі, то текстура буде повторюватись, тобто вона стане тайловою.

При тайловій текстурі необхідно пильно слідкувати за упаковкою шелів, так як тайлова текстура – незакінчена унікальні шели потрібно пакувати окремо від тайлових шелів (рис. 1.2).

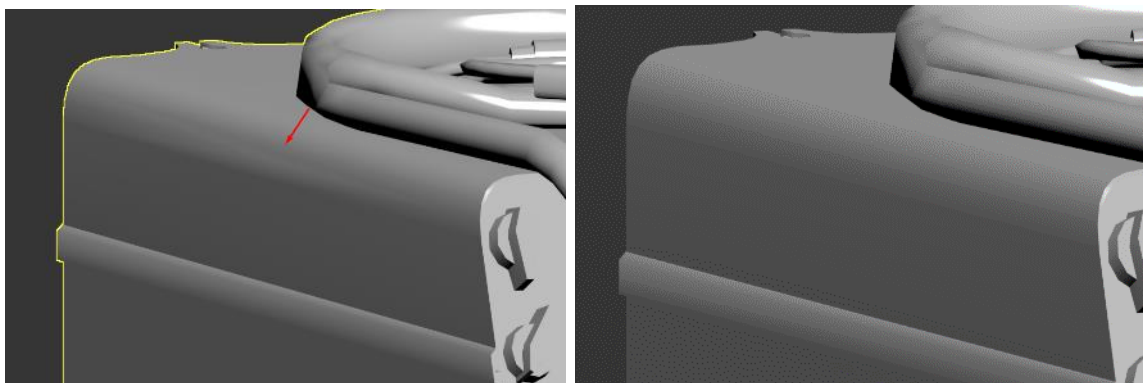


Рисунок 1.1 – Приклад впливу правильної триангуляції на шейдинг моделі (зліва – неправильно, справа – правильно)

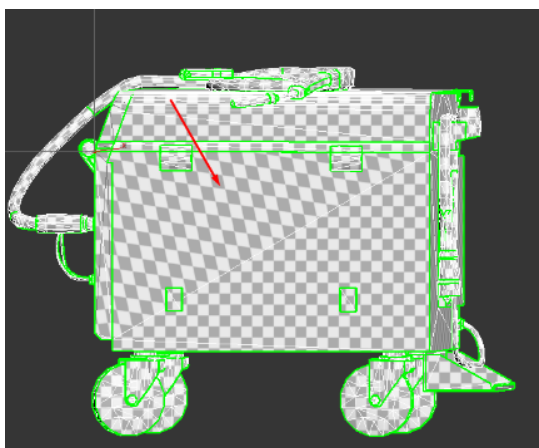


Рисунок 1.2 – Приклад потяжок на розгортці

У підсумку при розробці 3D моделі для гри необхідно дотримуватись певних вимог:

- а) мінімізація полікаунту, виконання низько деталізованої моделі;
- б) максимально можлива схожість Low Poly та High Poly моделей;

- в) правильні групи згладжування;
- г) правильна топологія;
- д) правильний розподіл сітки;
- е) тріангуляція після завершення етапу моделювання.

1.2 Аналіз аналогів

Сучасні AAA-відеоігри вражають своєю графікою та різноманітністю. На сьогодні індустрія AAA-відеоігор – один з найпопулярніших та наймасштабніших напрямів розробки ігор. З кожним роком індустрія розвивається дуже стрімко [10].

Серед аналогів, на які можна звернути увагу, це – Call of Duty Cold War (рисунок 1.3). Це популярна гра для ПК та консолей від Infinity Ward, що встигла стати популярною в світі порівняно з старими частинами гри. Одна з найцікавіших частин гри в усій франшизі, перед провальною іншою частиною гри, Call of Duty Cold War відновила увесь авторитет та значимість компанії у світі [1].



Рисунок 1.3 – Call of Duty Cold War

Call of Duty Cold War – це нова частина франшизи, що захоплює гравців своєю історією, атмосферою, сюжетом, чого наприклад бракувало в перших частинах ігор. Головні ідеї сюжету в серії Call of Duty особливо не відрізнялась від звичайного шутера, але саме в Call of Duty Cold War у вас є задача врятувати світ від ядерної загрози під час холодної війни. Вибраний

історичний період розробниками дав змогу відчуту гравцям ту напругу, котра була в ті часи, а саме створення берлінської стіни, пропаганда, агенти, вербування тощо.

Більшість цільової аудиторії гри – чоловіки від 17 – 30 років.

Переваги візуальної та технічної складової 3D-моделей Call of Duty Cold War:

- а) гра має реалістичну графіку;
- б) використання фотограмметрії;
- в) якісні та чіткі текстури.

Недоліки візуальної та технічної складової 3D-моделей Call of Duty Cold War:

- а) погана оптимізація мешу;
- б) погане відображення волосся у грі.

У висновку можна сказати, що візуальна складова в Call of Duty Cold War на високому рівні, персонажі виглядають дуже реалістично, а текстури котрі були створенні за допомогою фотограмметрії додають ще більшого реалізму та якості гри.

Наступний аналог – The Last of Us 2 (рис. 1.4). Одна з найвідоміших та найпопулярніших ігор в жанрі стелс – екшену та survival horror для Play Station 4 та Play Station 5. Розробка американської компанії Naughty Dog. Головна ідея гри – це знайти ліки від зомбування людей під час апокаліпсису.



Рисунок 1.4 – The Last of Us 2

Гра має дуже гарний сюжет з драматичними частинами, також сюжет передає всю напругу та складність життя людей під час зомбі апокаліпсису за допомогою візуальної частини, музики, розмов між персонажами, зовнішнього вигляду людей та навколишньої середовища.

Більшість цільової аудиторії гри – чоловіки та жінки від 17 – 30 років.

Переваги візуальної та технічної складової 3D-моделей: The Last of Us 2:

а) гра має реалістичну графіку;

б) стилістика вигляду навколишньої середовища та персонажів відповідає сюжету гри;

в) реалістичні анімації.

Недолік візуальної та технічної складової 3D-моделей The Last of Us 2 – погане відображення текстур шкіри під час гри.

Гра чіпляє гравців не тільки візуальною складовою, а ще драматичними та зворушливими сценами між персонажами, що змушує гравця ще більш перейматись за свого героя та його майбутнє в грі.

1.3 Основні етапи створення 3D-моделей в Gamedev індустрії

Першим кроком створення моделі – це створення високополігональної (High-Poly) моделі з великою деталізацією. Високо деталізована модель розробляється для того щоб, за допомогою її спроектувати усю деталізацію на Low-Poly модель таким чином розробляються Normal Map, Ambient Occlusion та Curvature. Ці карти потрібні для розробки текстур, але Normal Map використовується ще для оптимізації.

Наступне, що виконує розробник після High-Poly моделі – це ретопологію, або на основі вже існуючої високополігональної моделі створюється низька (Low-Poly) за кількістю полігонів, модель. Це набагато спрощена модель по полікаунту ніж High-Poly, що складається з мінімальної

кількості полігонів, достатнього для візуального сприйняття тривимірного світу гри, також саме вона використовується в самій грі. Тому що займає набагато менше ресурсів відео-пам'яті пристрою ПК чи консолі.

Коли створено обидві моделі, розробник починає робити розгортку Low – Poly. Розгортка – це майбутні текстури та карти, від її якості залежить відображення текстур та карт у грі. Тому розробнику слід уважно слідкувати за кожним розрізом на UV, щоб кількість розрізів була мінімальна та щоб вони були максимально сховані від очей гравця. Після створеної розгортки художник створює карти такі як: Normal Map, Ambient Occlusion та Curvature. У спеціальній програмі розробник повинен за допомогою інструментів проєкції спроектувати всю деталізацію від High-Poly моделі на Low-Poly.

Після всіх етапів розробки моделі (рис. 1.5) настає фінальний етап створення – це текстурування. Головна задача художника під час текстурування – це передати історію моделі та розробити модель під вимоги двійки.

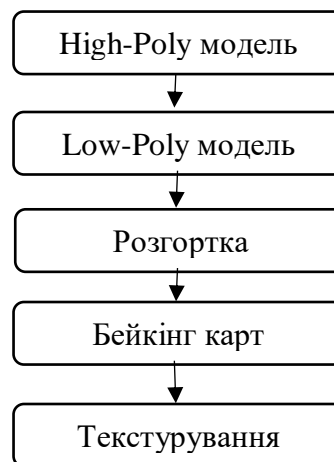


Рисунок 1.5 – Алгоритм створення 3D-моделі

1.4 Основні методи моделювання

На сьогоднішній день тривимірна графіка використовується в різних індустріях, це не тільки ігри але й ще кіно, CG, дизайн автомобілів, будівель та навіть в медицині [20]. Існує ціла низка методів моделювання та створення тривимірної візуальної частини. Можна використовувати класичне

полігональне моделювання, булеві операції, скульптинг, фотограмметрія, CAD моделювання. Всі ці методи допомагають створювати красиві та якісні 3D-моделі. В сучасних іграх розробники використовують всі доступні методи та інструменти аби прискорити процес створення контенту [7]. При створенні сучасних ігор всі методи моделювання тісно зв'язані між собою, розробник використовує багато різних програм для того щоб прискорити процес розробки та зробити цікаву та красиву модель.

Полігональне моделювання – це самий старий та класичний метод створення 3D-моделі. Його суть полягає в тому, що розробник використовує примітивні форми та додає полігони за допомогою додаткових вершин та ребер. Існує безліч інструментів для задання форми, наприклад, видавлювання полігонів (Extrud), створення округлих форм на гострих кутах (Chamfer), зшивання дірок між полігонами (Bridge) та ще багато інших, тільки в одній програмі існує декілька сотень різних інструментів. Звісно всі інструменти знати не обов'язково, але необхідно вміти використовувати ту базу інструментів або модифікаторів, котрі допоможуть створити фінальний продукт. Цей метод використовується більше при створенні твердих поверхонь, професійною мовою – це називається Hard Surface, існують навіть окремі художники для цього напрямку, вони розробляють наприклад зброю, автомобілі, мебелі та взагалі все, що не має пластики.

Переваги методу полігонального моделювання:

- а) великий контроль над щільністю полігонів;
- б) існує можливість створювати стек історії застосованих модифікаторів, що дозволяє швидко повернутись до попереднього етапу моделювання, без втрати часу.

Недоліки полігонального моделювання:

- а) необхідно завжди слідкувати за топологією, особливо при створенні High-Poly моделі;

б) витрачається більше часу при створенні High-Poly моделі, через постійну зміну модифікаторів та інструментів, а також збільшується час розробки через слідування за кожним полігоном при створенні форм.

Один із найцікавіших методів моделювання – це скульптинг. Суть його в тому, що це та сама ліпка з глини, тільки цифрова, за допомогою графічного планшету розробник може вільно створити модель з самої звичайної сфери. Такий метод частіше всього використовується для створення персонажів, тому що набагато легше і швидше створити пластичні форми за допомогою такого методу. Але також скульптинг використовується для Hard Surface, найчастіше це робиться для всяких вм'ятин, великих та унікальних подряпин, сколів. Ще деякі художники за допомогою скульптинга розробляють одяг для персонажів.

Переваги методу скульптинга:

- а) не потрібно слідувати за топологією;
- б) займає менше часу при створенні пластичних форм.

Недоліки скульптингу:

- а) потреба великих ресурсів пристрою;
- б) майже неможливий контроль полікаунту, через це модель потребує більше ресурсів пристрою;
- в) незручно створити рівні та прямі площини та лінії.

Булеві операції – це метод моделювання для Hard Surface, ідея цього методу, що розробник створює модель за допомогою відрізання форм від іншої форми і в підсумку художник отримує готову форму своєї моделі. Такий метод дуже часто використовують художники по зброї, бо можна швидко створити примітивні форми та повідрізати зайве [13]. Якщо розробник відрізав форму не в тому місці, то він може пересувати об'єкт у реальному часі, котрий буде відрізати від основної форми. Але після того, як всі зайві форми відрізані та вже створена фінальна форма моделі, розробнику лише залишається створити правильну топологію. Якщо брати на прикладі

створення зброї, то такий метод вбиває одразу двох зайців, тобто у вас є вже готова форма і для High-Poly, і для Low-Poly моделей.

Переваги булевих операцій:

- а) швидкість створення форм;
- б) не обов'язково слідкувати за топологією.

Недолік булевих операцій – після створення форм, фінальна форма має багато зайвих вершин.

Один із сучасніших методів створення 3D моделей – це фотограмметрія. Це дуже простий та цікавий метод розробки моделей. За допомогою великої кількості фотокамер, об'єкт фотографується з різних ракурсів в статичному вигляді. Після створення фотографій, яких знадобиться декілька десятків але залежить від складності форм об'єкту, спеціальна програма розробляє реконструкцію об'єкту з великої кількості вершин. На цьому етапі художник має хмару точок, потім програма зшиває всі ці вертекси і створює високо полігональну модель з кольором.

Переваги фотограмметрії:

- а) висока точність створення форм об'єкту;
- б) швидкість розробки High-Poly моделі.

Недоліки фотограмметрії:

- а) після створення реконструкції об'єкту, художник повинен почистити зайві полігони та видалити артефакти, котрі створила програма, тому що фотоскан не може створити на сто відсотків ідеальну рівну та чітку модель;
- б) після фотоскану модель має дуже багато полігонів.

CAD моделювання – це спосіб, котрий також використовується для Hard Surface моделей. Раніше таке моделювання використовувалось тільки для розробки моделей, котрі будуть створюватися на верстатах, тобто це болти, шайби, але і проектувалися і великі об'єкти, такі як ракети, вогнепальна зброя. Тільки з часом CAD моделювання перейшло і до гейм-деву, цей метод допомагає швидко та чітко створити High – Poly модель але без дрібних фасок. Також таке моделювання використовують концепт –

художники для Hard Surface моделей, особливо в стилі Sci-Fi, тобто моделі роботів, частин органів людей, які будуть в майбутньому, в загальному це стиль гри Cyberpunk [16]. Суть цього метода полягає в тому, що програма за допомогою булевих операцій та математичних розрахунків створює нескінченну сітку на округлих формах, саме через таку сітку, форми виглядають дуже плавно, тому цей метод використовували на виробництві для верстатів, щоб на реальному об'єкті не було заламаних форм в округлих місцях. Найчастіше CAD використовують для створення роботів, зброї, пропсів тому що вони мають дуже багато рівних та гострих форм, та також саме роботи мають дуже багато дрібних деталей, які швидко можна створити не думаючи за топологію та полікаунт.

Переваги CAD моделювання:

- а) швидкість розробки та можна особливо не слідкувати за топологією;
- б) чіткість створення прямих та гострих форм.

Недоліки CAD моделювання:

- а) дуже велика кількість полігонів на фінальному етапі моделі, полікаунт можна знизити але модель втратить рівність форм.

Основні переваги та недоліки найбільш популярних графічних редакторів для тривимірного моделювання наведено у таблиці 1.1.

Отже, проаналізувавши основні недоліки та переваги, було обрано для роботи: Autodesk 3DsMax.

Таблиця 1.1 – Порівняльна характеристика програмних засобів

Порівняння	Метод	Швидкість та ресурс-потрібність	Для яких об'єктів призначено	Топологія
3DsMax	Полігональне моделювання	Займає багато часу, особливо для High-Poly.	Hard Surface	Необхідно слідкувати за сіткою та її рівномірністю
ZBrush	Скульптинг	Займає менше часу, через вільне	Органіка та об'єкти	При створенні форм, ZBrush

		керування при створенні моделі	наколишньої середі	опрацьовує будь-яку топологію
CAD	Створення моделі на мат. розрахунках	Не займає багато часу але витрачає багато ресурсів	Hard Surface, особливо зброя та роботи.	Топологія не підлягає редагуванню
Boolean	Відсікання форм	Швидкий метод створення форм моделі	Hard Surface	Створюється багато зайвих вершин
Megascan	Скан реального об'єкту	Не займає багато часу але витрачає багато ресурсів	Органіка, Hard Surface та усе інше	Багато артефактів на меші

1.5 Текстурування 3D-моделей

Сучасна індустрія AAA-відеоігор зробила величезний крок у якості та реалістичного вигляду текстур. Якість та чіткість текстур залежить від вимог гри. Треба враховувати на яку платформу гра розробляється, якщо це ПК, то треба розробляти текстури під високі, середні, та низькі якості налаштувань графіки. Тому що не кожний гравець зможе собі дозволити придбати дорогий комп'ютер, саме тому розробники намагаються зробити текстури під різні налаштування та під різні ресурси комп'ютерів. Але якщо гравець збирається грати в гру яка вийшла наприклад в 2020 році, а він має пристрій, котрий придбав ще у 2010, то не слід розраховувати на якісну графіку в грі, можливо гра зовсім не запуститься через слабкі ресурси пристрою. Якщо гра розробляється як ексклюзив, тобто тільки на консоль: Play Station або Xbox то з текстурами для розробників трохи легша робота. Тому що консолі мають сталі ресурси відео пам'яті, оперативної пам'яті, тощо. Через це розробники мають чіткі вимоги на які ресурси пристроїв їм розраховувати під час створення контенту. Але сама якість графіки буде все одно краща на комп'ютері ніж на консолях. Тому що деякі технології для поліпшення якості графіки існують тільки на комп'ютері, такі як реалістична симуляція волосся – HairWorks, реалістичне затінювання та освітлення сцени гри – RTX [14].

Сьогодні існує декілька способів текстурування тривимірних моделей, для цієї задачі також існує багато програм.

Найстаріший метод текстурування – це ручний, такий спосіб використовувався з часів першої частини WarCraft. Розробник малює в 2D редакторі кожен подряпинку та пилінку у програмі, використовуючи координати розгортки аби розуміти в яких місцях художнику необхідно додати той чи інший елемент в текстурі. Також такий метод витрачає багато часу, та неможливо створити стек історії створення текстур на кожному кроці.

Ручний спосіб в сучасних AAA-відеоіграх вже майже не використовується, тому що існують більш нові та якісні інструменти для створення красивих та цікавих текстур. Зараз один із найвідоміших методів текстурування – це створення текстур за допомогою процедурних текстур та генераторів. Такий більш сучасний метод має декілька шейдерів, багато різних інструментів та саме головне – це PBR. Саме через PBR текстурі реалістична відображаються у грі та якісно показують фізичні властивості різних матеріалів, тобто метал має свої відблиски від сонця, пластик інші відблиски, всі матеріали по-різному відображаються в грі. До створення PBR такого якісного та реалістичного відображення текстур не було. Використовуючи генератори та процедурні текстурі, котрі називаються гранж, художник створює текстурі на моделі. Розробник може вибрати різні пензлики, додати різних подряпин, іржу, змішувати різні кольори. За допомогою генераторів можна швидко додати бруд або пил, або пошкодження на моделі. Також кожний шар текстурі має свої налаштування, можна додати ефект висоти, тобто щоб подряпина виглядала наче вона дуже глибока і це все на текстурі без додаткових полігонів. Цей спосіб більш швидкий та творчий, тому що текстура малюється не по розгортці, а вже по самій моделі. Але такий метод ще й більш технічний, через сучасні вимоги PBR, шейдерів та інших підпрограм, котрі виконуються на відео-прискорювачі. Для досягнення реалістичного вигляду текстурі, слід не забувати, що генератори та гранжи – це лише процедурні інструменти, котрі допомагають створити кожний шар, базу та образ текстурі. Тому необхідно

слідкувати аби процедурні інструменти не видавали себе у фінальному вигляді, тобто якщо просто накидати без всякої логіки текстур, то буде каша замість реалістичного вигляду текстури. В реальному житті не існує ідеального зносу по краям або чітких плям, тому щоб цього запобігти, художник багато домальовує пензликами, додає різних відтінків кольору, різних типів гранжів, тощо.

Текстурування за допомогою фотограмметрії – це затратний на ресурси спосіб текстурування але такий метод допомагає добитися найбільшого реалізму в текстурах. Ідея такого текстурування в тому, що після реконструкції справжнього об'єкту, модель має колір, таким чином розробник створює з цього кольору текстуру базового кольору, а потім карту рафнесу. Після цього розробник накладає ці дві текстури на розгортку моделі. Потім вже в програмі для текстурунгу художник використовуючи пензлики підмальовує усі нерівності та потяжки, бо неможливо накласти ідеально текстуру, котра з самого початку не підходить під унікальну розгортку моделі.

Цей спосіб має суттєві недоліки:

а) не все можна відсканувати, тобто якийсь камінь або стару дошку легко відсканувати і артефактів після реконструкції буде небагато (це залежить від якості скану та складності форм об'єкту), але якщо сканувати авто, зброю, будівлі, то скан не зможе ідеально реконструювати форми та текстури через великий масштаб об'єкту, складність форм та велику кількість деталей;

б) якщо розробник створює фантастичного героя, наприклад орка, то художнику не звідки взяти скан, так як орків в реальному житті не існує;

в) бувають відеоігри, котрі мають стилізовану графіку, тобто мультяшну, тому також не звідки робити скан, щоби застосувати скановані текстури на моделі;

г) не всі компанії мають мегаскан, тому що це дуже дорогий пристрій і потрібні люди, котрі вміють його вправно використовувати.

Існує також більш швидкий метод текстурування, суть його полягає в тому, що можна використовувати справжні фотографії і за допомогою

спеціального інструменту просто штампувати фотоматеріал на модель. Перед тим як накладати фото на модель необхідно вирівняти перспективу фотографії. Але такий спосіб далеко не завжди підходить для використання, наприклад, якщо розробник створює зброю, одяг, персонажа, то неможливо ідеально штампувати фото на модель, так як ці моделі мають складні форми і в них є різні карти, котрі допомагають текстурувати. Такий спосіб краще використовувати для великих та простих об'єктів, наприклад, цистерна, в неї дуже проста розгортка і тому непотрібно шукати багато різних та дрібних шелів на координатах текстури. Тому складні моделі в сучасних іграх текстурують процедурно або використовують частково текстури з фотограмметрії [17].

1.6 Основні вимоги при проектуванні 3D-моделі

Усі тривимірні моделі складаються з вертексів та ребер. При будь-якому методі моделювання, будь то скульптинг, полігональне моделювання або фотограмметрія. Усі моделі мають полігони. Тому при моделюванні необхідно слідкувати за топологією, навіть при скульптингу це треба робити, аби запобігти сходження полігонів і не виходили так звані «зірочки», особливо цьому приділяють увагу при створенні органіки, округлих форм об'єкту та одягу, топологія повинні бути максимально рівномірною. При створенні High-Poly методом полігонального моделювання – це називається Subdivision modeling, необхідно аби сітка мешу складалась майже тільки з чотирикутних полігонів, бо якщо полігон матиме 3, 5 або більше вершин, то меш буде з неправильним шейдингом. Тільки в деяких випадках можна створювати трикутні полігони, найчастіше це на рівних площинах, бо площина немає випуклих та опуклих форм, тому слідкувати за шейдингом в місцях з трикутними полігонами не варто (рис. 1.6).

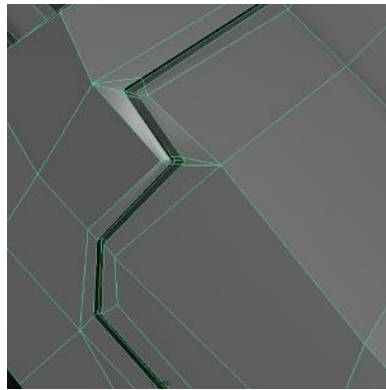


Рисунок 1.6 – Невірна топологія на округлих формах для Subdivision через присутні трикутники

Також необхідно слідкувати за товщиною стін при створенні High-Poly. Не допускаються дуже тонкі стіни, бо при віддаленні камери від моделі, площини, котрі створюють товщину, будуть створювати ефект наче б вони перетинаються. Допустима мінімальна товщина стін в моделі – це три відсотки від усього об'єму моделі, але такі дані залежать від вимог гри, наприклад в грі STALKER 2 товщина стінок - це вище сказані три відсотки. Треба слідкувати час від часу за напрямком векторів кожного полігону – це називається нормалі. Робиться для того аби не було Flip Normal, полігон ззаду не відображається для полегшення об'єму пам'яті моделі, тому якщо полігон буде вивернутий в іншу сторону, то в грі гравець побачить отвір на місці цього полігона. Високо-деталізована модель створюється для запікання фасок на Low-Poly та створення інших карт для текстурингу. Аби фаски були помітні на Low-Poly навіть при низькому форматі текстури та при Mir-Mapping, розробник створює більш жирні скруглення на краях, ніж котрі має реальний об'єкт у житті. Також фаски потрібно робити різними, бо реальний об'єкт немає однакових округлень на гострих кутах, тому при створенні моделі під Subdivision розробник додає додаткові ребра – називаються Support Loop. Вони стримують форму об'єкту, інакше інструмент Subdivision занадто сильно згладить всю модель (рис. 1.7-1.8).

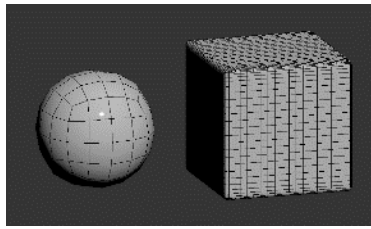


Рисунок 1.7 – Бокс із Support Loop та без них (при Subdivision)

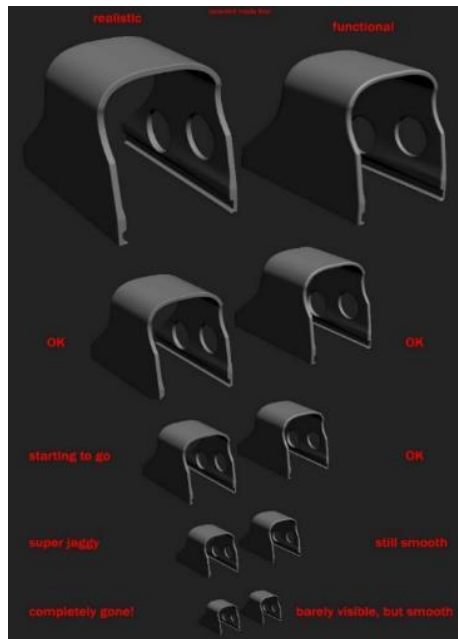


Рисунок 1.8 – Приклад фасок на тривимірній моделі
(Справа – вірно, зліва – невірно)

Існують окремі види моделей, для котрих топологія повинна бути максимально, майже на сто відсотків рівномірною, наприклад моделювання автомобілів. Так як, майже кожний автомобіль має округленні форми, тому сітку роблять так, щоб кожен полігон мав форму квадрату аби запобігти потяжкам на шейдингу та не втратити форму моделі. Також такий додатковий об'єм сітки дає змогу робити більш дрібну деталізацію, наприклад, дверна рукоятка автомобіля. Вона складається з декількох деталей – це сам корпус рукоятки та отвір для ключа. Отвір дуже маленький але його необхідно моделювати, тому на початку додаються додаткові полігони аби розробнику мав змогу розпустити сітку для базової топології під отвір для ключа. Таке ж саме правило рівномірної топології стосується і для створення Low-Poly

моделі. Так це зайвий полікаунт але шейдинг на округлих формах автомобіля буде коректний, тому при запіканні карт не буде артефактів.

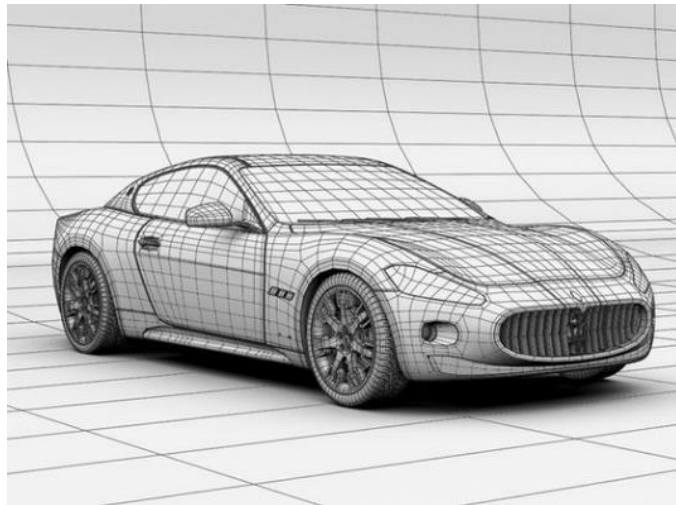


Рисунок 1.9 – Приклад рівномірної топології на High-Poly моделі

На відміну від автомобілів, створення High-Poly моделі зброї або об'єктів, котрі складаються суттєво тільки з площин та мають гострі кути під 90° , не обов'язково дотримуватися рівномірної сітки. Але також необхідно слідкувати за шейдингом моделі. Потрібно звертати увагу на габарити моделі аби модель не була набагато менша персонажа. Наприклад, холодильник, середня висота холодильника сто вісімдесят – двісті сантиметрів. Тому поруч з моделлю можна створити звичайний бокс з правильними габаритами та налаштувати масштаб моделі під розмір боксу або просто поставити поруч персонажа для розуміння розмірів моделі, котра створюється [15].

При створенні Low-Poly моделі, задача полягає в тому щоби зробити максимально низький полігонаж моделі та зберегти форми об'єкту. В сучасних іграх майже кожний гвинт моделюється на Low-Poly, бо на даний час ігри приближаються до реалістичної картинки та ресурси пристроїв дозволяють використовувати великий полігонаж на Low-Poly моделі. Якщо зрівняти моделі ігор, котрі вийшли 5-7 років назад, то можна помітити різницю, що багато дрібних об'єктів запікались на Normal Map, зараз Normal Map запікає тільки фаски кожної деталі та дрібні стики різних об'єктів, тому

що вони дуже маленькі та непомітні оку гравця. Необхідно слідкувати за рівною відстанню між полігонами в Low-Poly моделі на округлих формах аби запобігти зламаності та нерівності форм на об'єкті. Наприклад, розробник створює спусковий гачок пістолета, він має круглу форму якщо дивитися на нього з бокової сторони. Тому необхідно слідкувати за рівномірністю відстані між полігонами (рис. 1.10).

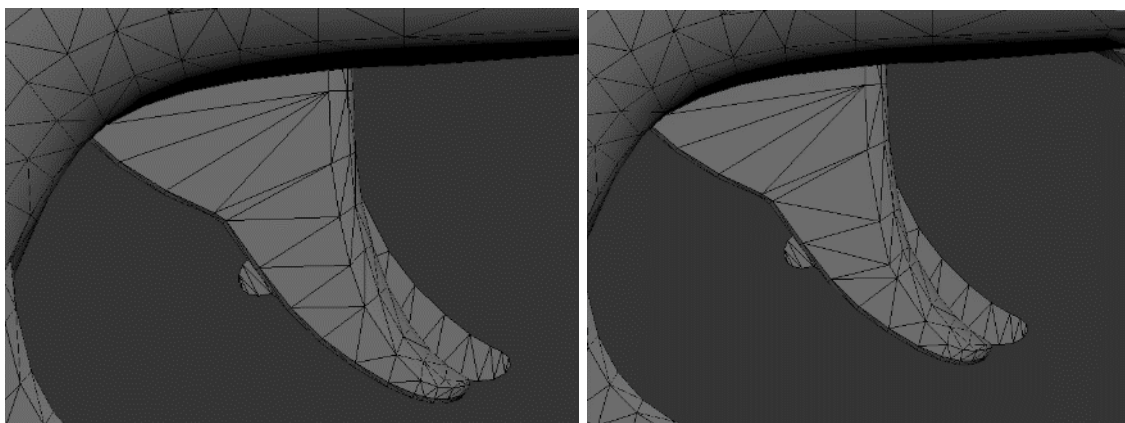


Рисунок 1.10 – Приклад впливу на форму від відстані між полігонами
(Зліва – правильно, справа – неправильно)

2 ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Програмне забезпечення для збору референсів

Першим кроком розробки 3D-моделі для відеоігри є пошук референсів, на які художник опирається для подальшого моделінгу та текстуровання. Знаходження референсів – це відповідальна та не швидка робота, чим більше їх буде тим краще, але вони не повинні повторюватися. Існує безліч програм для збереження фотоматеріалу – це може бути звичайний 2D редактор, можна загрузити фотографії на пристрій та переглядати їх, або навіть не завантажувати фото і дивитись на референси в інтернеті. Референсом не завжди рахується фото, це також може бути відеоматеріал. На сьогодні дуже легкий спосіб збереження референсів в одному файлі – це використання програми PureRef. В цій програмі можна швидко перетягувати фото в файл і додавати туди нескінченну кількість референсів, не завантажуючи фото з інтернету.

Деякі розробники використовують реальні об'єкти для референсів (рис. 2.1).

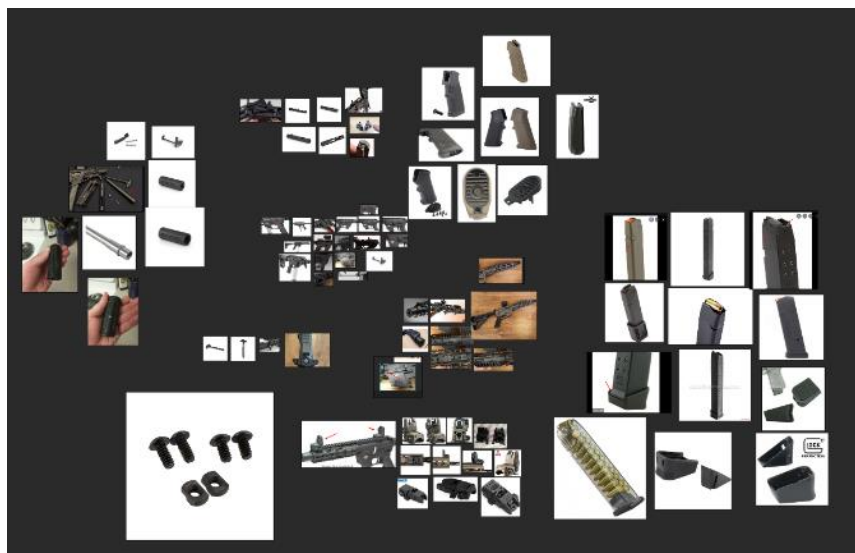


Рисунок 2.1 – Приклад референсів до моделі

Якщо, наприклад, художник створює зброю, він може придбати або взяти в оренду репліку гвинтівки, пістолету або автомату, котрий розробляє.

Таким чином він має у себе біля робочого місця справжній об'єкт з реальними габаритами та з усією деталізацією, бо не завжди можна знайти фото з різних ракурсів. Всю цю інформацію, котру отримує художник з референсів він використовує для створення моделі. Кількість референсів на один об'єкт може досягати до декількох десятків або сотень. Від них залежить майбутня історія та стилістика моделі котру художник передасть гравцям. Також референси допомагають зберегти відповідність форм мешу до реального об'єкта у житті, тому що художник бачить з різних ракурсів об'єкт, котрий моделює.

Необхідно вміти читати референси, це не просто зображення об'єкту з пошкодженнями. Розробник повинен умовно розбивати кожний референс як окремий крок, тобто де знаходиться бруд, пил, подряпини, іржа та вміти зробити правильну базу глянцу та шорсткості при текстуруванні. Для створення текстур не обов'язково шукати референси об'єкту який буде розроблятися, можна брати за образ пошкоджень різних об'єктів, головне щоб вони відповідали образу майбутньої моделі.

2.2 Програмне забезпечення для розробки 3D моделей

На сьогоднішній день в CG-індустрії (Computer Graphics) існує безліч програм для 3D моделювання. Такі як: Maya, 3DsMAX, ZBrush, 3D Coat, Blender, Fusion 360, Marvelous Designer – це стандартні програми індустрії, котрі кожний художник використовує для розробки моделей, кожна з цих програм призначена для створення окремих задач. Наприклад, в Marvelous Designer неможливо створити якогось робота чи будинок, ця програма призначена для симуляції тканин. Всі програми мають одночасно схожі та різні інструменти, але кожний редактор допомагає швидше та якісніше розробити модель. Всі програми створені для окремих задач, нажаль в сучасній індустрії не існує одного унікального софта, котрий може

виконувати всі етапи створення 3D моделі: від полігонального моделювання – до симуляції тканин. Особливо дуже схожі між собою 3DsMax, Maya та Blender.

Вони мають майже однаковий інструментарій, однак різниця існує та звичка до використання програмного забезпечення також присутня [5].

Першим був створений 3DsMax він розроблявся більш для створення дизайну будівель та різних приміщень, з часом його почали використовувати в GameDev, в цій програмі можна створювати моделі та анімацію. 3DsMax має неймовірно простий інтерфейс, усі основні інструменти, котрі найчастіше використовують для створення моделі, знаходяться в окремій колонці, котру неможливо не помітити, тому цей інтерфейс буде зрозумілий навіть тим, хто відкрив програму вперше. Але з виходом нового редактору Maya, анімація перейшла до неї, бо Maya має більше якісних та зручних інструментів для анімації, тому майже всі аніматори виконують свою роботу в Maya. В цій програмі також можна створювати 3D моделі, але Maya має свій недолік – це інтерфейс, він складніше ніж у 3DsMax, тому для новачків, котрі переходять з 3DsMax на Maya, складно зрозуміти де знаходяться інструменти, особливо це стосується моделювання. Також 3DsMax та Maya мають свій спільний недолік. Всі ці програми небезкоштовні, вони дуже коштовні, тому не кожна людина, котра хоче навчитися створювати 3D моделі або анімацію може дозволити купувати дороге програмне забезпечення. Але час йде і програми розвиваються, з'являється більше інструментів, котрі пришвидшують час розробки моделі. 3DsMax на сьогодні рахується більше як вже застаріле програмне забезпечення, але ще досі ним користуються дуже багато художників, котрі звикли до нього. Тому вже з'явився набагато оновлений інструментарієм редактор по розробці 3D моделей, створений анімаційною студією NeoGeo.

Blender – це більш новий пакет по зрівнянню 3DsMax та Maya. Особливо останні 3 роки програма зробила великий крок у своєму розвитку. Також програма по зрівнянню з 3DsMax та Maya безкоштовна. Blender має

дуже простий та зрозумілий інтерфейс, швидкість створення моделі швидше приблизно на 30 відсотків по зрівнянню 3DsMax та Maya, завдяки зручній навігації в інтерфейсі та швидкості знаходження потрібного інструменту серед сотень інших модифікаторів. Також Blender має інструментарій скульптингу, тепер не обов'язково експортувати модель в ZBrush але все це залежить від складності поставленої задачі перед художником при створенні моделі. Якщо це не складна форма, то можна скульптити в Blender, але якщо необхідно створювати складні форми з дрібною деталізацією, то необхідно експортувати модель в ZBrush, так як ця програма більш призначена для скульптингу та вона має більше різних інструментів, котрі допоможуть створити якісну фінальну модель [1].

Zbrush – програмне забезпечення для скульптингу. За допомогою Zbrush можна швидко створювати форми органіки не звертаючи увагу на топологію. Для розробки органіки в Zbrush, тобто створення персонажів, тварин – це ідеальне програмне забезпечення для таких задач. Zbrush має доволі не складний інтерфейс для використання, так як найголовніший інструмент в програмі – це пензлики, котрі мають свої характеристики для моделювання. Пензлики знаходяться в окремій вкладці або можна відкрити вкладку з пензликами за допомогою гарячих клавіш. Можна вибирати розмір, жорсткість, фактуру та багато різних налаштувань пензликів. З кожним мазком розробник додає форм на моделі. По суті – це цифрова ліпка з глини. На початку художник вибирає примітивний об'єкт, будь то бокс, сфера чи циліндр та починає з грубих форм, додаючи масу на об'єкт, полігони самі по собі вслід за кожним мазком додаються.

Також Zbrush дуже творчий та зручний для створення старих дощечок, котрі мають глибокі прожилки від старіння, елементи одягу, та майже все можна скульптити в Zbrush. Як і любе програмне забезпечення Zbrush має недоліки. Програма має два типу моделювання, основний – скульптинг, та другий – це класичне полігональне моделювання. Але в Zbrush дуже не зручно через навігацію в програмі, користуватись полігональним

моделюванням. Так як інтерфейс та головна задача програми більш розраховані для вільного скульптингу, тому через це страждає зручність, швидкість та головне якість моделювання. В Zbrush не так зручно створювати ідеально рівні площини, по зрівнянню як це можна робити в Blender, 3DsMax та Maya. Так, існують пензлики та деякі інструменти котрі все ж таки дозволяють зробити рівні форми, але на це вийде більше часу, та ще знадобиться зробити багато зайвих дій аби виконати задачу. Zbrush також використовують ще для створення High-Poly моделей зброї. Програма має дуже зручне керування булевими операціями. Тобто при створенні художник розробив форми відсікання в 3DsMax чи Maya та потім експортує в Zbrush [4]. За допомогою модифікатора Live Boolean в Zbrush можна в реальному часі створювати необхідні форми. Це дуже зручно для того якщо розробник помилився з габаритами або з місцем отвору в меші, то об'єкт, котрий відсікає форму, можна перетаскувати, не боявшись, що залишиться зайвий отвір, котрий вийшов при помилці. Також Zbrush може перерахувати любую топологію, за допомогою інструменту ZRemesher. В програмі неймовірно зручно та дуже просто створювати фаски. В полігональному моделюванні фаски на High-Poly моделі створюють за допомогою сапорт лупів, в Zbrush такого немає, програма має інструмент під назвою Polish. Софт створює фаски сам, художник вибирає на скільки велику фаску треба зробити на об'єкті. Це дуже сильно заощаджує час при розробці High-Poly моделі. Також в Zbrush доволі просто та легко створювати різні фактури, наприклад, сколи, ржавіння металу, вм'ятини, тріщини та багато різних пошкоджень на об'єкті.

Marvelous Designer – це програмне забезпечення використовується для симуляції тканин. Суть полягає в тому, щоби розробник створив максимально реалістичні складки на одязі. Художник створює форми с площин, наприклад, спідниця, футболка. Створюються шви між площинами, як на реальному одязі. Потім програма за допомогою симуляції зшиває ці площини та симулюється на персонажа, котрий стоїть в сцені. Після симуляції на тканині створюються складки. Таким чином створюється High-

Poly модель одягу, таку модель можна використовувати вже для ретопології, а можна ще допрацювати з деталізацією в Zbrush. Додати розірвані стрічки, дрібні складки та додати фактуру тканини.

3D Coat – українське програмне забезпечення для розробки 3D моделей. Головна фішка цієї програми - це створення воксельної графіки. Тобто кожна модель складається з маленьких кольорових боксиків, приклад такої графіки – це гра Minecraft. Також в програмі можна скульптити але інструментарій не такий великий та якісний як Zbrush. Тому Zbrush вважається еталоном для скульптингу. В 3D Coat можна працювати з текстурами, можна редагувати готові текстури та текстурити модель з нуля. При редагуванні текстур програма має свої переваги по зрівнянню зі звичайним 2D редактором – змінювати текстури можна одразу на моделі, а не на розгортці. Такий спосіб заощаджує багато часу, бо розробник одразу бачить, що відбувається на моделі при редагуванні текстур. Також в 3D Coat присутній інструментарій для текстурування, але в цій програмі рідко текстурять бо існують набагато якісніше програми для створення текстур. Існує так званий стандарт програм для розробки тривимірної графіки. 3D Coat не такий ще розвинений як інші програми, але деякі невеликі речі в ньому зручніше робити.

Fusion 360 – програмне забезпечення для моделювання та проектування. За допомогою цієї програми можна швидко та зручно створювати різні моделі для Hard Surface моделей. Дуже зручно використовувати Fusion 360 для створення 3D концептів, особливо зброя та роботи. Роботи мають дуже велику деталізацію, тому можна швидко створювати деталі не думаючи о правилах топології, це дуже сильно заощаджує час. Також цю програму використовують архітектори та дизайнери, котрі в 3D створюють свої проекти. На сьогоднішній день це дуже зручно, так як є можливість побачити дизайн будівлі чи квартири в тривимірній графіці.

У загалом художники в гейм-деві використовують всі доступні програми, котрі їм можуть допомогти швидко та якісно створити модель. Не

існує чітких рамок, що кожна програма потрібна тільки для чогось одного, кожен використовує софт для своїх цілей та виконує свою роботу під вимоги гри. Зараз аби створити красиву модель персонажа необхідно як мінімум 3 програми тільки для моделювання. Але ще існує багато програм, в котрих створюють текстури, розгортки, бейкінг карт, анімацію та тестування. Існують навіть окремі програми для ретопології, але їх дуже рідко використовують, бо це можна робити в звичайному 3DsMax, Blender, Maya. Насправді програм існує ще більше, але всі не потрібні, так як для виконання роботи достатньо знати базу програмного забезпечення. Кожний розробник вибирає програмне забезпечення під свої навички, художник розуміє в яких програмах та інструментах він сильніший, а де більш слабкий. Також в великих студіях художники розділяються на види: художник по зброї або по персонажам.

2.3 Програмне забезпечення для створення розгортки

Розгортка – необхідний етап створення будь-якої моделі, без неї не існуватимуть текстури та карти моделі. На етапі розгортки необхідно завжди думати в якому місці створювати розрив текстури. Головне правило розгортки – це створити її максимально безшовною. Дуже важливо слідкувати за потяжками на розгортці, необхідно пильнувати за чекером, саме він допомагає розробнику побачити щільність пікселів на окремому шелі та допомагає виявити потяжки на розгортці. На сьогодні розгортку можна створювати в багатьох програмах, будь то 3DsMax, Blender, Maya, Zbrush.

Існує програмне забезпечення, котре створено тільки для розробки розгортки, наприклад, UVLayout та RizomUV. Різниця між цими програмами в тому, що UVLayout та RizomUV мають більше інструментів для зручної та швидкої розробки розгортки.

UVLayout на перший погляд має застарілий інтерфейс але в цьому є переваги, в програмі немає зайвих вкладок, кнопок, нічого не відволікає від

роботи, та з таким інтерфейсом можна розібратись повністю з програмою буквально за декілька хвилин. Також за допомогою цієї програми можна дуже швидко створити розрізи на розгортці, для цього знадобиться тільки один інструмент, це дуже заощаджує час. Єдиний недолік UVLayout в тому, що програма слабо відображає потяжки на розгортці, тому слід перевіряти коректність розгортки в RizomUV або вже в програмі для моделювання, наприклад 3DsMax.

RizomUV – більш нове програмне забезпечення для розгортки. Він трохи складніший, але має більше інструментів. На перший погляд може здаватися навіщо тоді UVLayout якщо існує RizomUV? RizomUV має свої переваги перед UVLayout, такі як: щільність пакування шелів розгортки у координатах від 0 до 1, а також програма краще відображає потяжки на розгортці та набагато краще релаксить шели аби запобігти помилок на майбутній текстурі. Але RizomUV складніший по навігації, тому деякі художники розгортку роблять в обох програмах, використовуючи тільки сильні їх сторони. Наприклад, в UVLayout створюють тільки розрізи шелів, а релакс розгортки, пакування та створення оверлапів роблять в RizomUV.

2.4 Програмне забезпечення для створення Normal Map, Curvature, AO

Розробити карти для оптимізації моделі та текстурювання можна за допомогою різних програм. Наприклад, Normal Map створюється як і в Maya, 3DsMax, Substance Painter, Substance Designer, Marmoset Toolbag та XNormal. Всі ці програми з легкістю дозволяють розробити карти. Normal Map – це сама основна карта для всіх моделей, майже кожна модель в грі має таку карту. Але кожний підхід для запікання Normal Map використовується для різних пайплайнів розробки тривимірного контенту. Наприклад, в грі Farming Simulator від швейцарських розробників Giants Software карта нормалей запікається в Maya або XNormal. Ці програми дуже якісно запікають рівні форми, як наприклад балки, рама трактора та інше. Так як

карта нормалей повинна не мати градієнтів – це рахується артефактом та такі помилки будуть псувати коректне відображення моделі з Normal Map у двіжку або Substance Painter при текстуруванні. Також такий метод запікання передбачає відображення стиків карти нормалей на Low-Poly моделі. Normal Map в грі Farming Simulator використовувалась тільки для фасок. Художник розробляє модель подібну High-Poly, але не створюються різні сколи, подряпини та різноманітні пошкодження – це такі вимоги пайплану при розробці гри. Високо полігональна модель розробляється за використанням сапорт лупів але інструментів Subdivision не використовується. Головна задача при розробці High-Poly – це створення коректного шейдинга за допомогою смуз груп в програмі для моделювання. Також такий метод запікання Normal Map вимагається для коректного відображення карт нормалей у двіжку, котрий компанія створювала сама. Через низький розмір карт у грі та щільність пікселів на шелах розгортки запікання Normal Map в Maya та особливо XNormal дозволяє легше редагувати артефакти на карті після запікання.

Самий найстаріший, не зручний та не якісний інструмент для розробки карт – це використання 3DsMax. Через застаріли та не якісні інструменти в програмі дуже важко створити коректну Normal Map, не кажучи вже про Ambient Occlusion. Normal Map та вся решта карт створюється за допомогою проєкції з High-Poly моделі на Low-Poly. На жаль в 3DsMax дуже погано виконується проектування через постійний перетин променів під час запікання карти. Тому в результаті як би розробник не намагався зробити все акуратно та витрати на це багато зайвого часу, після запікання на карті виявляється дуже багато артефактів, помилок, градієнтів та інших похибок на текстурі, котрі потім необхідно видаляти за допомогою інших програм та витрачати на це багато зайвого часу.

Більш сучасний та зручний метод розробки карт – це використання Substance Painter. Substance painter – програмне забезпечення, котре призначене набагато більше для створення текстур ніж для розробки карт.

Але так як процедурне текстурування, прикладом котрим являється Substance Painter, використовує всі розроблені карти. То якість запікання в програмі дуже велика. Однак як будь-який софт програма має недоліки зі сторони запікання карт, такі як: немає можливості контролювати висоту проекції над Low – Poly при проектуванні деталізації з High-Poly, неможливість вибирати кількість променів для створення карти, бо від більшої кількості променів на один піксель збільшується якість та чіткість карти, що за собою веде меншу кількість помилок після розробки але велика кількість променів витрачає більше ресурсів пристрою та потребує більше часу. Також після запікання витрачається час на експорт карти в будь-який редактор для видалення помилок на текстурі. В свою чергу експорт карт та текстур з Substance Painter трохи складніший ніж звичайний експорт моделі з 3DsMax або Maya.

Рідше використовують для запікання карт Substance Designer. Програмне забезпечення використовується більш для створення базових матеріалів, наприклад, потріскана фарба на дереві або вовняна тканина. Потім розроблені базові матеріали застосовують при подальшому текстуруванні в Substance Painter для прискорення процесу. Програма має свій підхід для створення текстур, матеріалів та карт. Ідея полягає в тому, що всі текстури та карти створюються за допомогою нодів (рис. 2.2).

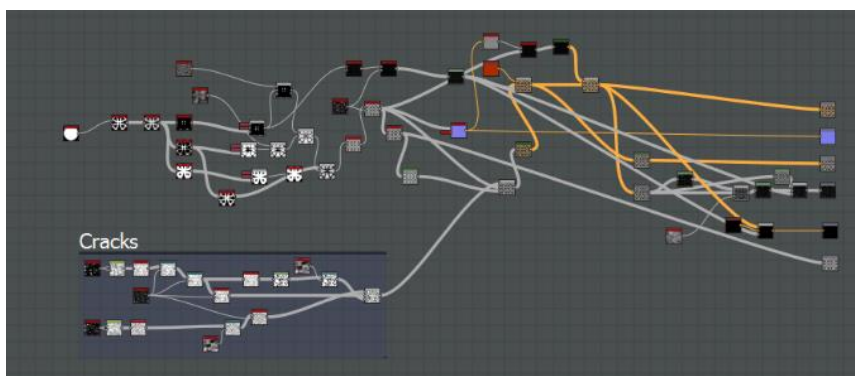


Рисунок 2.2 – Приклад нодів в Substance Designer

Метод створення текстур за допомогою нодів складний та його використовують більш досвідчені розробники, котрі знають не тільки як

створювати текстури та карти, а ще й розуміють їх технічну складову, наприклад, які канали необхідно підключати до окремого нода аби текстура відображалась належним чином.

Самий передовий та найбільш використовуваний метод запікання карт – це використання програмного забезпечення як Marmoset Toolbag (рис. 2.3). Програма має великий та досить простий інструментарій для створення карт.

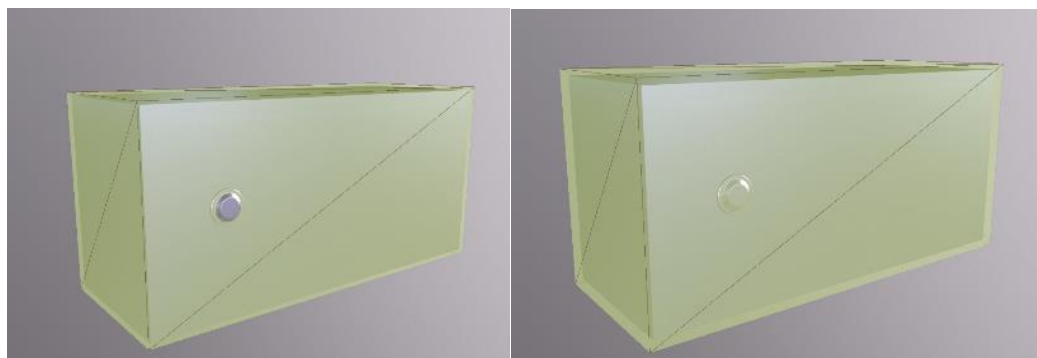


Рисунок 2.3 – Контроль висоти проекції перед запіканням
(Зліва – невірно, справа – вірно)

Кожен етап запікання карт можна контролювати, підлаштовуючись під вимоги гри для якої розробляється модель. В Marmoset Toolbag можна розробляти всі карти, котрі використовуються в сучасній індустрії, їх дуже багато. Програма також дозволяє запікати карту базового кольору та рафнес з реконструйованої моделі, котра була створена за допомогою фотограмметрії. Існує контроль над кількістю променів на один піксель під час запікання, легко вирівнювати нормалі після першого бейку карт за допомогою інструмента Paint Skew (рис. 2.4).

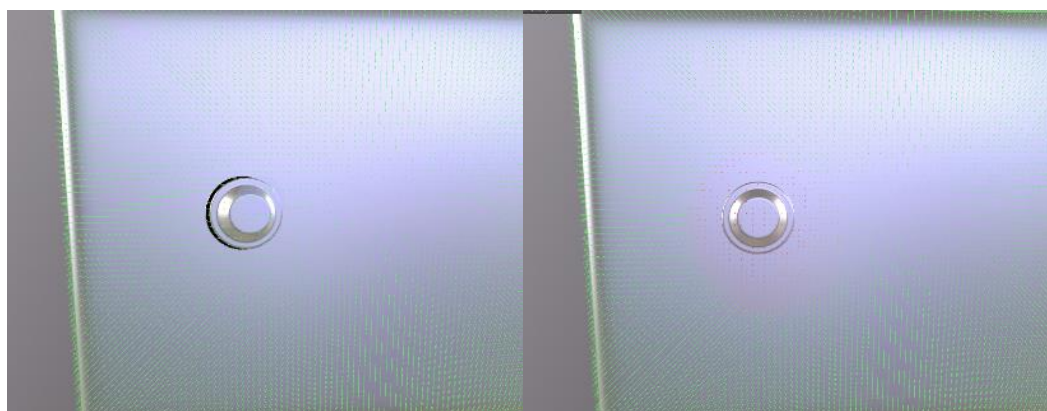


Рисунок 2.4 – Приклад впливу нормалей на запікання деталізації
з використанням інструменту Paint Skew

(Зліва – до використання інструменту, справа – після використання)

Нормалі необхідно вирівнювати в тих випадках, коли карта запікається на великий та деталізований об'єкт. При стандартних налаштуваннях нормалей Normal Map, котра забейкала наприклад гвинт на моделі, то він буде відображатись не рівно на текстурі. Тому слід використати інструмент Paint Skew для вирівнювання нормалей під 90 градусів відносно кожної площини моделі. Також програмне забезпечення дозволяє вибирати бітність карт, тобто 8 біт, 16 біт, 32 біт. Чим більше число – тим буде якісніше відображення карти на моделі але потребує більше ресурсів та часу на опрацювання пристроєм.

Marmoset Toolbag має можливість створювати бейк групи, робиться це для того коли запікається модель з великою кількістю під об'єктів, (наприклад у зброї це приціл та гвинтівка) щоби запобігти перетину проекції кожного меша. Якщо відбувається перетин проекції від кожного об'єкту, то на фінальному етапі карта матиме артефакти, найчастіше – це чорні плями на фіолетовій карті нормалей. Тому слід робити бейк групи (рис. 2.5) для тих об'єктів, котрі стоять поруч або просто розсувати об'єкти подалі один від одного.

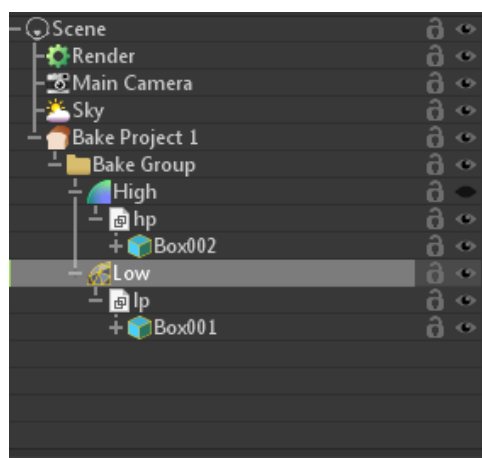


Рисунок 2.5 – Приклад стеку бейк групи

Але розсування має свій недолік, так як меш, котрий розсували знаходиться в одній бейк групі, а значить нормалі у всіх об'єктів спільні, то вони можуть ще сильніше стати не рівними відносно кожної площини до об'єкту. Через це розробник втрачає час на вирівнювання нормалей після першого запікання карти. Таким чином створюються вся решта карт, котрі необхідні для текстурування моделі. Ambient Occlusion необхідно максимально акуратно запікати, намагаючись уникнути перетину проєкцій та уникнути нерівності нормалей, бо карта тіней має усі допустимі відтінки від білого до чорного, також має дуже багато переливів кольору тіні, котрі не легко редагувати та видаляти артефакти на карті за допомогою інших інструментів.

2.5 Програмне забезпечення для редагування та видалення артефактів на картах текстур

Сучасна індустрія розробки AAA-відеоігор розвиваються дуже стрімко. Але навіть на сьогоднішній день розробникам необхідно робити багато рутинної, технічної роботи, на котру необхідно витратити додатковий час. Одним із етапів підготовки моделі до текстурування – це редагування карт та видалення артефактів. Існує два способи виконувати таку не цікаву та нудну роботу – використання 3Dcoat та Photoshop. Всі ці два методи підходять для такої роботи та використовуються для окремих задач.

3Dcoat – дуже зручно та швидко використовувати для редагування основного об'єму карти. Наприклад, при редагуванні карт, як Ambient Occlusion та Curvature виправляти артефакти можна одразу на моделі, крутити в сцені меш і продивлятися усі необхідні місця, де є помилки, дуже зручно, також такий метод заощаджує час та сили розробника [19]. За допомогою пензликів можна одразу розмити жорсткі переходи тіней на карті Ambient Occlusion та зробити більш чіткими лінії опуклої та випуклої форми на карті Curvature або навпаки домальовувати необхідні тіні, котрі погано

запеклись під час створення карт у Marmoset Toolbag. Нажаль 3DCoat має свої суттєві недоліки в редагуванні карт:

а) неможливість редагування градієнтів карт нормалей, тому що кожний відтінок та колір Normal Map передає свою інформацію на меш. Редагування такої карти в 3DCoat займе набагато більше часу. Тому якщо наважиться на такий метод видалення помилок на Normal Map, [2] то необхідно знати усі відтінки в окремих випадках на моделі для досягнення правильного виду карти. Але в більшість випадків складно сказати, який колір буде в окремому місці на меші, так як таких різних випадків може траплятись дуже багато раз та також неможливо передбачити який колір Normal Map буде використовуватись на окремому згині мешу;

б) 3DCoat не має можливості редагувати хоча б один піксель зображення за межею шела на розгортці, тобто виконувати редагування на падінгах. Тому якщо, розробник малює перелив кольорів на карті та замальовує артефакт на самому краю шела, то крайні пікселі шела не будуть замальовуватись і помилка залишиться.

Наступний метод редагування карт – це використання Photoshop. Таке програмне забезпечення використовується через використання програмою растрової графіки, такий вид графіки складається з пікселей. Тому редактори векторної графіки такі як, Illustrator або CorelDraw не використовуються для редагування карт, бо при переводі з векторної в растрову графіку втрачається якість зображення. Саме Photoshop використовують для редагування Normal Map, так як легше підібрати відтінок градієнту на карті для виправлення артефакту або більш коректного уточнення градієнту на текстурі. Photoshop має змогу редагувати кожний піксель зображення, тому піклуватись про неможливість виправлення крайніх пікселів на шелі текстури не варто. Також найчастіше програму використовують після того як зробили основну роботу в 3DCoat, бо в Photoshop можна ідеально рівно вирівняти лінію аж до розміру пікселя або чіткіше підібрати відтінок кольору для переливу тіней через

можливість бачити розробнику одразу всю карту на екрані (рис. 2.6).

Photoshop має свої недоліки при редагуванні карт:

- а) витрачається більше часу при редагуванні ніж в 3D Coat;
- б) Photoshop іноді погано відображає контрасти кольору Normal Map, необхідно використовувати додаткові інструменти для створення контрасту;
- в) стандартний формат зображення при збереженні займає більше місця, необхідно постійно переводити в формат Targa;
- г) неможливість переглядати карти в 3D сцені одразу в програмі.

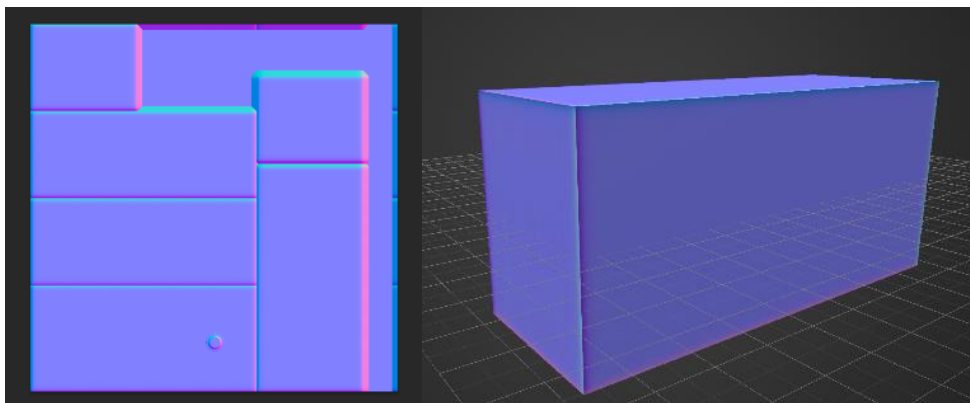


Рисунок 2.6 – Приклад редагування карт нормалей
(Зліва – Photoshop, справа – 3D Coat)

2.6 Програмне забезпечення для створення текстур

Нині в індустрії розробки відеоігор існує стандарт програм, котрі використовують розробники для створення текстур. В найбільшій кількості випадків художники використовують Substance Painter. Програма має дуже велику кількість інструментів та підходів до різних задач. Головна ідея створення текстур за допомогою Substance Painter – це процедурне текстурювання. Такий метод створення текстур рахується найновішим та найякіснішим, що зараз має індустрія GameDev [11]. Substance Painter дозволяє створювати текстури для фото-реалістичних відеоігор та розробляти текстури для ігор з стилізованою графікою. Процедурне текстурювання – дозволяє швидко та якісно створювати текстури для моделі водночас переглядаючи модель в 3D

сцені. Також можна використовувати для зручності другий вид в'юпорта – це перегляд текстури на розгортці, тобто звичайний 2D вигляд.

Substance Painter має велику кількість стандартних шейдерів, найчастіше – це шейдер під робочий процес Metal/Rough так як цей підхід більш новий та легший для розуміння технічних вимог для PBR. Раніше розробники використовували робочий процес Spec/Gloss, він більш тяжкий з точки зору логічного розуміння процесу роботи. Але цей метод має деяку перевагу в створенні текстур, особливо для розробники текстури металу. За допомогою каналів Specular та Diffuse можна створити дуже красиві та різні градієнти металу в один крок, так як Specular має всю палітру кольорів, по зрівнянню з методом Metal/Rough. Розробити градієнти та переливи кольору на металі в один крок неможливо, через те, що канал Metallness – має тільки відтінки від чорного до білого.

Substance Painter дуже легко опрацьовує текстури, котрі були створенні з фотограмметрії. Розробнику залишається тільки загрузити текстуру та додати кожний канал до відповідного каналу в створеному стеку текстур. Також програма має так звані мости до двіжка, тобто, аби художник не займався експортом кожний раз для перевірки якості відображення текстур у двіжку, в Substance Painter мається інструмент для швидкого експорту до двіжка – це дуже сильно заощаджує час, тому що, на реальному продакшені кожна хвилина має свою цінність [6].

У програмі також існує можливість створювати текстури методом Hand-Paint. Це метод текстурування за допомогою тільки пензликів. Не використовуються гранжи, альфи, але використовуються інколи генератори. Такий спосіб створення текстур використовують найчастіше для мультиплекційних, казуальних та стилізованих ігор. Усі процедурні інструменти розробники використовують більш для фото реалістичних текстур. Тому, що реалістичні текстури виглядають більш одного кольору, немає різких контрастів у відтінках та насиченості кольору. Тому стилізовані відеоігри і виглядають більш яскравими та кольоровими.

В Substance Painter деякі розробники додають свої шейдера та сцену освітлення аби чіткіше розуміти одразу як будуть виглядати текстури у грі – це також заощаджує час, але інколи все ж таки приходиться експортувати текстури в движок, бо властивості рушія та різні його ефекти відображення та освітлення змінюють програмісти.

Substance Painter постійно оновлюється через швидкий розвиток індустрії і з кожним разом додається більше інструментів, котрі допомагають швидше та якісніше розробляти текстури.

2.7 Програмне забезпечення для тестування моделі

На етапі тестування немає чітких вимог для виконання такої задачі. Розробник може використовувати любі інструменти. Найчастіше тестування моделі з текстурами проводиться в двіжку гри, це може біти Unity, Unreal Engine 4 або 5, FrostBite або самостійно створений двіжок компанії. При тестуванні часто звертають увагу в першу чергу на розмір моделі, необхідно аби меш був реалістично відповідний до габаритів інших моделей. Тестувальники перевіряють також відображення, якість та логіку текстур, тобто щоб пил або бруд був в тих місцях, де вони можуть бути у реальному житті. Єдине, що розробник може ще перевірити так це валідацію кольору і рафнесу у двіжку та перевіряється колізія мешу [12].

Розробники перевіряють також анімацію, необхідно аби анімація була якомога реалістична, не було різких рухів у персонажів та перевіряється згин мешу на суглобах або тканинах при анімації. Будь-який асет проходить декілька разів перевірку до того моменту як його приймуть до продакшену. Також модель проходить таку процедуру перевірки через постійні зміни у двіжку, тому слід час від часу перевіряти кожний асет до дрібниць. Наприклад, дуже важливо перевіряти логіку асета, не допускається щоби рослини або дерева були на домівках – це не логічно (іноді трапляються навіть і такі випадки, через неуважність художника або левел-дизайнера та також таке трапляється через великий об'єм відкритого миру в грі, чим

більше об'єм та складність розробки гри – тим легше зробити помилік).
Перевіряється також повторювання текстур (тайл), навіть якщо, текстура
була розроблена тайлова за вимогами пайплайну необхідно максимально
приховувати повторюваність бруду, пилу, іржі та багато іншого на текстурі.
Не допускається копіювання одного асета багато разів в одному місці – це
одразу кидається в очі гравцю та виглядає не реалістично.

3 РОЗРОБКА ТРИВИМІРНОЇ МОДЕЛІ

3.1 Пошук референсів

Мета пошуку референсів – рефборд (рис. 3.1) повинен мати в собі відповіді на більшість питань, які можуть виникнути на всіх етапах створення асета. У цьому кожен окремий референс має відповідати на кожне конкретно питання.



Рисунок 3.1 – Приклад рефборду

Основне завдання пошуку референсів – зібрати дані та сформувати картинку в голові. Референси дають цілісне розуміння того, як потрібно реалізувати асет: на що звернути увагу, що слід опустити, які проблемні моменти та як їх вирішати.

Збір референсів йде від загального до приватного: від силуету до середніх та малих форм, до матеріалу та деталей. Часто на початку збору важко визначитися, що потрібно шукати. Тут допоможе аналіз виду об'єкта:

- а) яким буде силует;
- б) цікавість форми силуету;
- в) оживання та реалізм силуету.

Основні правила збору референсів:

- а) зібрані у результаті референси мають закрити більшість питань з візуальної складової;
- б) не збираються референси з унікальною варіацією;
- в) необхідно уникати різних референсів на один схожий об'єкт, тобто якщо розробник збирає референси для автомобіля, то потрібно шукати фото відповідно моделі автомобіля;
- г) уникати хаосу у рефборді, необхідно намагатися використати якісні зображення, можна додавати нотатки.

3.2 Розробка 3D моделі

Після завершення етапу пошуку референсів починається етап розробки моделі. Для створення меша використано полігональне моделювання та скульптинг. Робота починається від High-Poly моделі до Low-Poly. High-Poly робиться окремими деталями відповідно до реального об'єкту. Під час створення високо деталізованої моделі потрібно створити всі деталі аж до маленьких гвинтиків, гайок та унікальних подряпин, сколів, вм'ятин. Для підвищення ще більшої деталізації під час скульптингу використовуються різні пензлики, текстури (alfa) та інструмент шуму (noise), все це додає мікро нерівності на меші, що робить модель більш живою та унікальною.

В High-Poly моделі (рис. 3.2) дуже важливо зробити фаски, тому що в реальному житті нічого не має гострих країв та кутів. Під час створення високо деталізованої моделі можна не перейматись за кількість полігонів, треба розуміти вимоги пайплану та вмикати художню складову, слідкувати за композицією та передачею кольору асета. Потрібно постійно розуміти як розбити однотонність мешу та форм за допомогою різних інструментів.

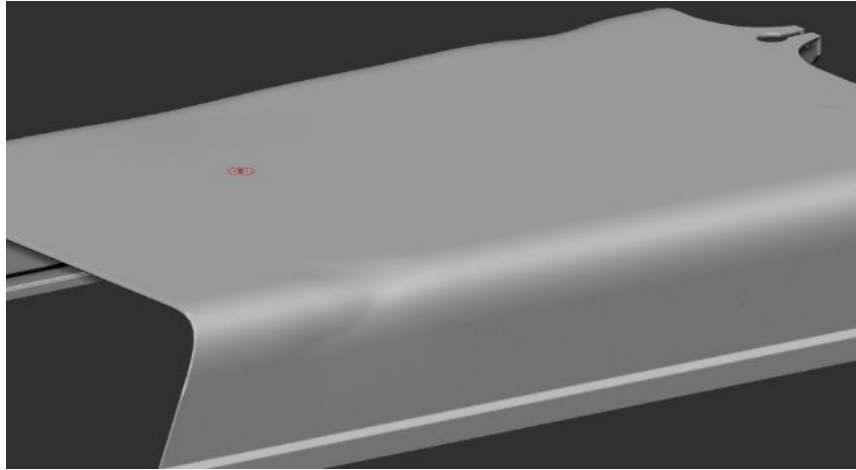
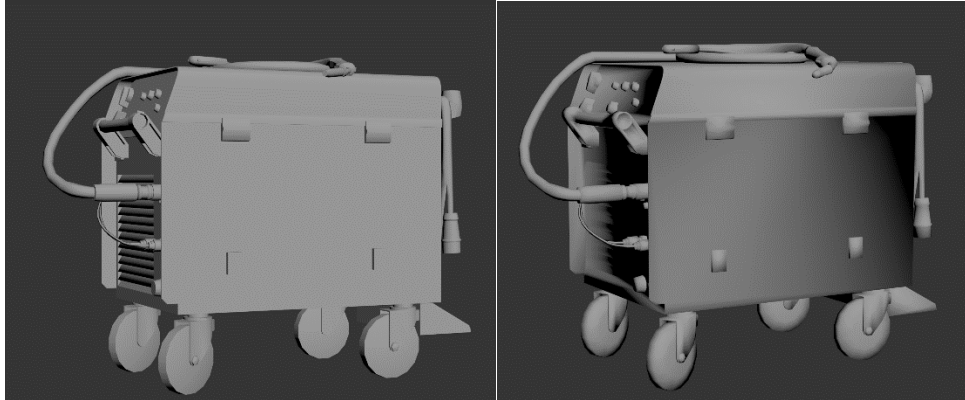


Рисунок 3.2 – Частина High Poly моделі

Під час розробки High Poly моделі можна витратити дуже багато часу та зусиль, але головна задача розробки ігор – це швидкість та якість, тому витратити увесь час, виділений на весь пайплайн створення моделі, не варто, навіть наскільки би не була красива в фіналі High Poly модель.

3.3 Розробка Low – Poly моделі

Після High-Poly моделі створюється Low-Poly. Низько деталізована модель створюється одним куском, тому майже усі об'єкти зшиваються для запобігання артефактів, окрім деяких деталей як дроти, колеса, кнопки, великі гвинтики, гайки та шланги. Під час створення Low-Poly моделі необхідно слідкувати за шейдингом поверхні мешу аби запобігти артефактів на картах. Головна задача при створенні Low-Poly зробити меш дуже оптимізованим, тобто щоб моделька мала як най менше полігонів, але щоб форми не були дуже заламані. На великих округлених формах додається максимально велика кількість полігонів, але все треба дивитись за ситуацією, не треба перебільшити. На рис. 3.3 представлено приклад коректного шейдингу Low-Poly моделі. Коректний шейдинг – це необхідна вимога для розробки карт, текстурування та для коректного відображення моделі в ігровому рушії.



Русинок 3.3 – Приклад шейдингу

(зліва – правильно, справа – неправильно (заламаний меш))

На площинах необхідно витратити як найменше полігонів, бо на площинах немає об'єктів, які потребують більше полігонів. Також багато залежить від допустимої кількості полігонів на всю модель. Якщо художник створює об'єкт, котрий буде далеко від персонажа, то витратити багато мешу непотрібно, часто такі моделі робляться для навколишньої середи, бо кожний полігон, кожне ребро, кожна точка – це додаткова затрата ресурсів пристрою (ПК чи консоль), навіть с сучасними технологіями Nanite, котрі дозволяють робити багато полігонів майже на кожній моделі все одно не допускається марна трата мешу бо оптимізацію і логічне використання полігонів ніхто не відмінює. Але є різні ситуації при створенні Low-Poly. Наприклад, якщо художник створює модель зброї. В сучасних іграх зброя має дуже багато полігонів бо пістолет, автомат чи щось інше завжди перед камерою гравця, зброя займає 30 відсотків всього екрану, тому заламаність геометрії не допускається. Якщо зрівняти кількість полігонів у зварювальному апараті (рис. 3.4-3.6), котрий має 2500 полігонів або 5000 трикутників, то кількість полігонів у зброї може бути у 10 раз більше, все залежить від розмірів моделі. Сучасні моделі пістолетів досягають до 30000 трикутників, а якщо це автомат або гвинтівка з різною кастомізацією, то такі моделі можуть мати від 60000 до 100000 трикутників.

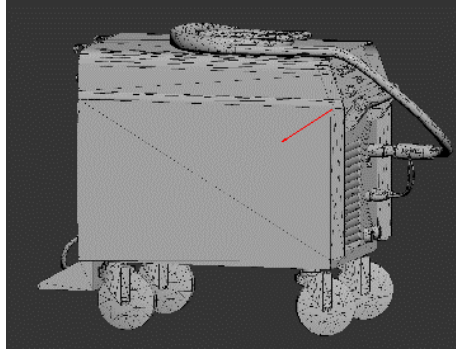


Рисунок 3.4 – Приклад розподілу полігонів на площинах

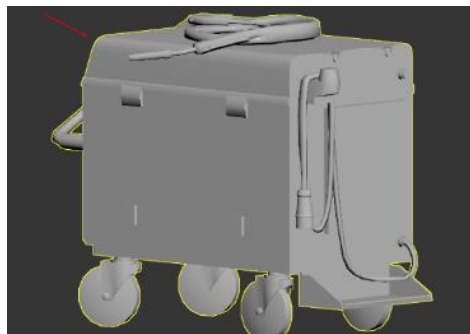


Рисунок 3.5 – Приклад правильного розподілу полігонів

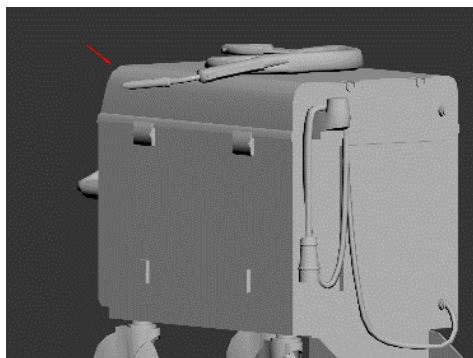


Рисунок 3.6 – Приклад неправильного розподілу полігонів

3.4 Розробка розгортки

Далі по звершенню розробки Low-Poly, художник створює розгортку (UV). Чітких правил при створенні розгортки не існує але треба завжди пам'ятати, що від якості розгортки залежить майбутні текстури та карти.

Спочатку художник просто розрізає кожен деталь, тобто створює шви між шелами. Шви повинні надрізатися в тих місцях, де гравець не може побачити розрив розгортки, а значить внизу або в інших складних місцях, до яких гравець не добереться, – це робиться ще для того, якщо текстура буде в маленькому розмірі, наприклад, 512×512 пікселів то при таких обмеженнях текстура стає не такою чіткою і більш мутною, через це шов може відобразитись ще сильніше. Після розробник виконує вирівнювання шелів, тобто кожне ребро повинно бути чітко вертикальними або горизонтальними – це робиться аби текстура на краях шелів не мала піксельних зубчиків. Якщо не вирівнювати кожне ребро на шелах то при запіканні карт, текстури на краях будуть мати нерівності – це посує якість відображення текстур у грі.

Після нарізання та вирівнювання робиться оверлап – це створюється для того, щоб заощадити місце на UV але оверлапи не завжди потрібні, якщо художник буде текстурити кожний об'єкт як унікальний тоді оверлап не використовуються. За допомогою оверлапів можна збільшити щільність пікселів на текстурі але накладені шели будуть мати однакову текстуру. Перед упаковкою розгортки, розробник масштабує шели до меншого розміру, а деякі шели до більшого розміру – це виконується для того, щоб збільшити щільність пікселів на дрібних деталях, при масштабуванні шели маленьких об'єктів будуть мати більше площу на розгортці, а отже матимуть більше пікселів на шелах, тому якість зображення на дрібних місцях буде якісніша. Така сама процедура виконується для місць, котрі гравець дуже рідко або ніколи не побачить, наприклад, дно автомобіля але такі шели навпаки максимально зменшують, такий підхід до різниці щільності пікселів створює новий простір для пакування в координатах від 0 до 1 – це називається Texel Density (рис. 3.7).

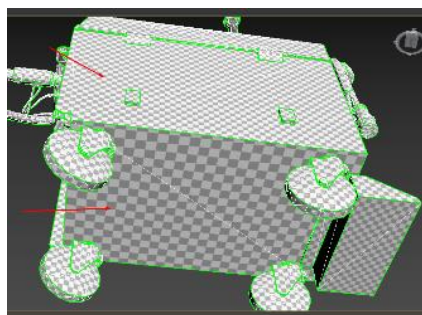


Рисунок 3.7 – Приклад різниці Texel Density

Після всіх етапів – робиться упаковка розгортки, її можна робити як вручну так і автоматично. Ручний спосіб більш нудний та займає більше часу але художник після упаковки легше орієнтується на розміщенні кожного шела. Більшість художників пакують розгортку автоматично – це заощаджує час та сучасні інструменти для упаковки роблять це якісно. При пакуванні розгортки необхідно слідкувати за відстанню між шелами – це називається падінг. Не дозволяється аби шели накладались один на одного хоча б одним пікселем, якщо це не оверлап.

Створення оверлапів (рис. 3.8) заощаджує місце на розгортці в координатах від 0 до 1 та також оверлап створюють для того, якщо модель має однакові об'єкти, наприклад, колеса зварювального апарату то шели одного колеса будуть знаходитися в координатах розгортки, а решта три чітко накладаються піксель в піксель один на одного та всі шели інших трьох коліс відсуваються на одну координату від основної розгортки. Відсування виконується аби запобігти артефактів та помилок при запікання Ambient Occlusion та Normal Map щоби програма розуміла на який шел запікати тінь або фаски.

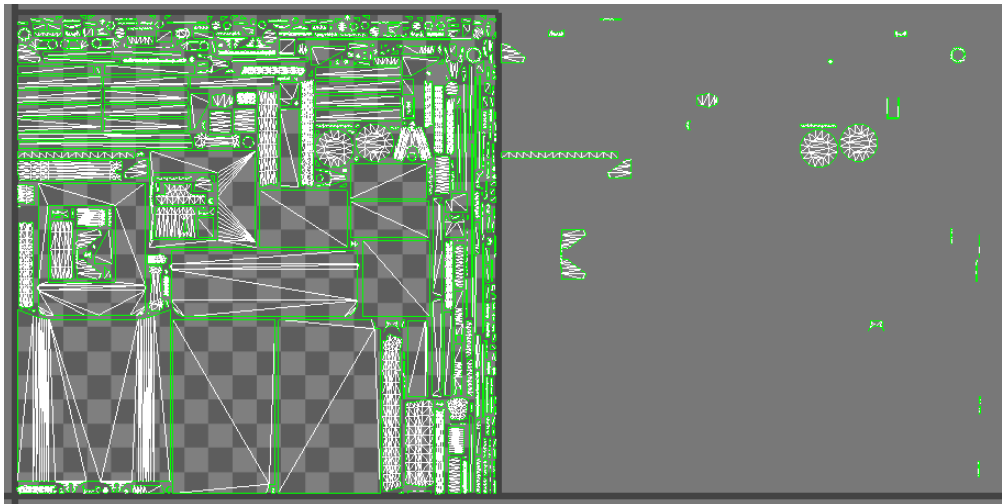


Рисунок 3.8 – Розгортка з оверлапами

Відстань між кусками розгортки залежить від розміру текстури:

- а) текстура 4096×4096 має відстань від 16 до 32 пікселів;
- б) текстура 2048×2048 має відстань від 8 до 16 пікселів.

Після упаковки необхідно перевірити розгортку на наявність потяжок – це виконується за допомогою стандартної текстури (чекер – рис. 3.9), якщо квадратики рівні, то потяжок немає, а отже розгортка коректно вирівняна та нарізана.

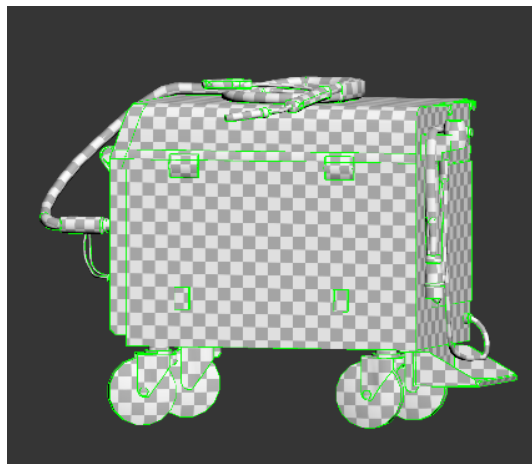


Рисунок 3.9 – Приклад правильного чекера

3.5 Розробка карт для текстурування

Після упаковки розгортки художник створює Normal Map, карту тіней (ambient occlusion), curvature – це основні карти які допомагають при текстуруванні. Всі ці карти створюються використовуючи розгортку. Спочатку створюється Normal Map, за допомогою нормалей виконується проєкція всієї деталізації з High-Poly моделі на Low-Poly бо високо деталізована модель дуже тяжка і не підходить для гри, а низько деталізована з Normal Map виглядає майже як High-Poly (рис. 3.10-3.11).

Після запікання у кінці розробник може побачити як Low-Poly модель отримала деталізацію від High-Poly, та без додаткових полігонів. Все це робиться для оптимізації, бо в грі таких моделей може бути тисячі і всі вони різного розміру та полігонажу. Далі розробник створює карту тіней та криватуру, вони створюються таким же чином як і Normal Map (рис. 3.12-3.13).



Рисунок 3.10 – Low-Poly модель без Normal Map

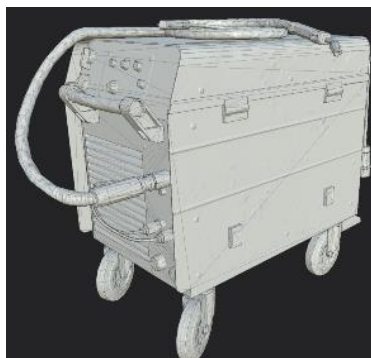


Рисунок 3.11 – Low-Poly модель з Normal Map



Рисунок 3.12 – Ambient Occlusion карта на моделі



Рисунок 3.13 – Карта Curvature на моделі

3.6 Розробка текстур моделі

Останній та самий творчий крок розробки моделі – це створення текстур. Головна задача під час текстурування – це передати історію моделі та розробити модель під вимоги ігрового рушія бо далі команда програмістів створює шейдера для покращення освітлення, відображення фізичних властивостей текстур. Також суворо забороняється використовувати на текстурах повністю чорний та білий колір, бо в реальному житті таких кольорів не існує і в рушію ці кольори будуть некоректно відображатись. До таких суворих вимог відноситься канал металік, рафнес та базовий колір, не дозволяється робити рафнес та металік в значеннях 1 або 0. Для розуміння, які максимальні значення дозволяються під окремий шейдер – використовується інструмент валідатор, він відображає правильні зони значень на текстурі, якщо якась зона підсвічується червоним, то значення неправильні, якщо зелений – все добре. Це необхідно для розуміння, де трапилась помилка, на якому каналі

(рафнес, металік або колір). Валідатор має свої налаштування, по яким можна обирати відповідні канали. На початку розробки текстур створюється база із рафнесу, додоється дрібний шум, різні текстури, які підходять для образу моделі, базу бруду, розводів, пилу і все це без кольору, і все на кожній деталі. База рафнесу будується на дуже грубих та ледве примітних гранжах. Кожний етап текстуровання нагадує написання картини – починається з фону та великих плям, потім додаються деякі елементи, за цим створюється на текстурі знос та бруд, а в самому кінці додаються дрібні деталі, такі як: унікальні подряпини, надписи та тощо. Робиться це для того щоб передати логіку забруднення моделі.

Наприклад, буде не логічно виглядати, якщо необхідно створити модель старого зварювального апарату і в нього не буде пилу, котрий осідає з часом або в загальному він виглядатиме дуже чистим. Під час текстурингу завжди необхідно експериментувати з кольором, на прикладі апарату, котрий стояв десятки років без роботи на вулиці, то верхня кришка об'єкту буде мати вигорання кольору, тобто стане світлішим і матиме інший відтінок.

Дуже цікавий елемент в текстурах – це іржа. Дуже рідко, коли існує один колір іржі. Усього в середньому може бути 4 основних типи іржі, все залежить від умов впливу навколишнього середовища на метал. Червоний колір (гідратований оксид) зустрічається у місцях де є підвищена вологість (рис. 3.14, зліва). Жовта іржа (рис. 3.14, справа) утворюється в місцях постійного контакту металу з водою, калюжа, стояча вода, місця, де метал постійно перебуває під впливом води.



Рисунок 3.14 – Червона та жовта іржа

Бура іржа, з'являється у місцях високого вмісту кисню та низької вологості. Виглядає у вигляді неоднорідних плям або лише у певних областях (забруднення та дефекти на поверхні металу).

Чорна іржа, буває у місцях низького вмісту кисню та низької вологості. Схожа на шар покриття зверху, що виникає при окисдуванні металу. Важливо! У процесі текстурування вкрай рекомендується використовувати тільки точкові крапління чорної іржі, щоб не надто затемнювати модель (рис. 3.15).

Усі текстури створюються за допомогою програми Substance Painter, ця програма дозволяє створювати текстури процедурно, тобто вона має свої інструменти для прискорення процесу, бо раніше текстури малювали руками до кожної подряпини. Процедурна магія це звісно добре, але руками художник все ж таки багато чого малює, він убирає зайві місця пошкоджень, котрі створив процедурний генератор. Робиться це для того щоб модель виглядала більш живою, а не як об'єкт з купою різних плям.



Рисунок 3.15 – Бура та чорна іржа

На фінальному етапі додаються дрібні деталі – це можуть бути маленькі бризки фарби, моторного масла, унікальні подряпини, або якісь надписи та наліпки, все це додає антуражності моделі – рис. 3.16.



Рисунок 3.16 – Фінальна модель з текстурами

4 ТЕСТУВАННЯ МОДЕЛІ В ІГРОВОМУ РУШІЮ

У даному випадку, основна ціль об'єднання – провести тестування коректності відображення текстур при стандартних шейдерах рушія. У редакторі матеріалів створюємо ID матеріал у рушію, котрий матиме відповідні канали для розроблених текстур. При імпорті моделі в сцену рушія – помилок та багів відображення моделі не виявлено, габарити моделі відповідають реальним габаритам об'єкта у житті, тонкі елементи моделі не видають помилок при віддаленій камері від мешу – не виявлено Z – Fighting.

Таким чином у процесі роботи було розроблено коректну розгортку, правильно створені карти та текстури – усе це доводить тестування готової моделі у рушію (рис. 4.1-4.2).

Дана тривимірна модель використовується в ігровій локації з іншими моделями для оточення в ігровому просторі – рис. 4.3.

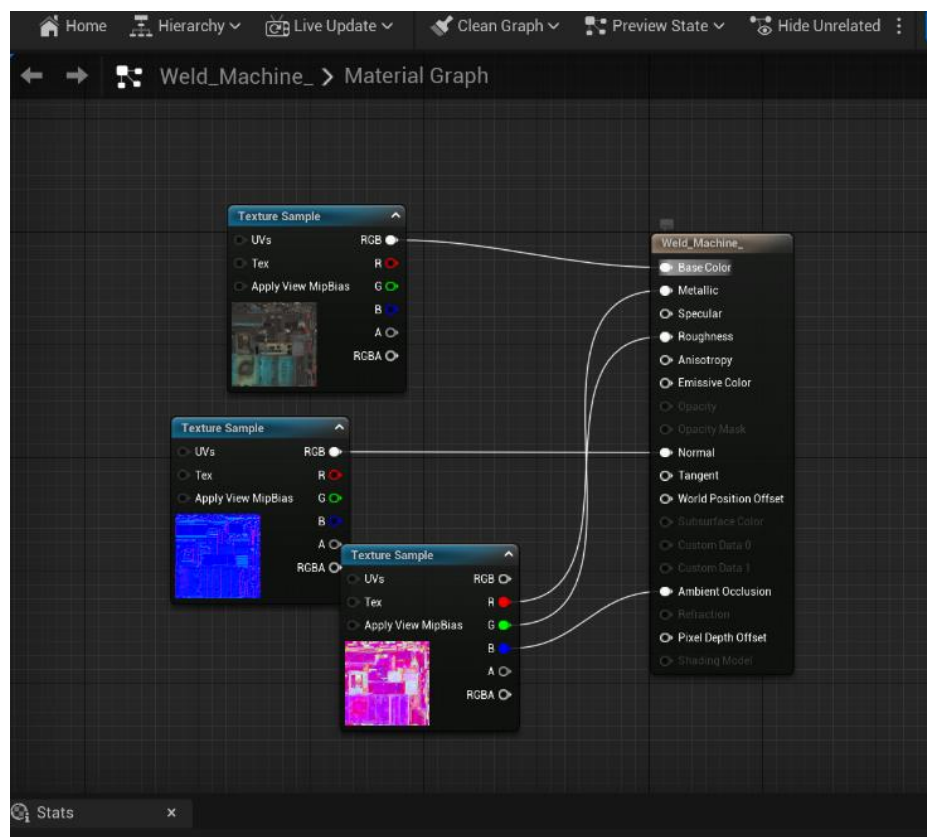


Рисунок 4.1 – Створення ID матеріала у рушію



Рисунок 4.2 – Створена модель у сцені рушію



Рисунок 4.3 – Використання моделі в ігровій локації

5 ЕКОНОМІЧНА ЧАСТИНА

У результаті виконання кваліфікаційної роботи створено 3D модель для відеогри STALKER 2 від компанії «GSC Game World». Розробка 3D моделей для ігор є важливим кроком для поліпшення якості графіки в грі, з виходом нових оновлень гри, розробники виправляють та видаляють деякі недоліки в графічній складовій. Таким чином, розробники залучають нових гравців до покупки гри в інтернет-магазині. Економічна ефективність розробки моделі розраховується перед створенням і розробкою контенту, у результаті чого можливо спрогнозувати потенційний ефект і доцільність впровадження даного контенту в компанії. Спочатку розраховується собівартість розробки, потім визначається ціна.

Розглянемо переваги розробленої моделі.

Впроваджена модель в гру має максимально низький полікаунт та якісні текстури, що відповідають сучасним вимогам AAA-відеоігор. Це означає, що модель потребує мінімальну кількість ресурсів пристрою при сучасній якості графіки, а отже, гра не буде вимагати багато технічних ресурсів, що приведе за собою більше покупців гри з слабим залізом на ПК чи консоль. Також частота кадрів гри не буде падати нижче рівня, при якому можна спокійно грати без затримок кадрів.

Модель статична, вона не має анімації, що рятує від необхідності створювати скелети анімації, витратити часові та грошові ресурси на створення анімації. При розробці моделі використовується коштовне програмне забезпечення, що нажаль підіймає загальну вартість гри. Дана модель займає невеликий об'єм пам'яті пристрою, що може заощадити частоту кадрів та час на обробку пристроєм моделі.

Розглянемо конкурентне середовище стосовно до тематики розробленої моделі. Гра, в якій впроваджена розроблена модель, буде продаватись в наймасштабніших інтернет-магазинах у світі для відеоігор, порівняно з іншими іграми, котрі мають власні майданчики для продажу, але не мають такої великої

аудиторії, котрі мають інтернет магазини як Steam, Epic Game Store. Відповідно до великого розвитку GameDev індустрії ігри розробляють все більш з реалістичною графікою за допомогою нових технологій, нових ігрових рушіїв, у зв'язку з цим гра має явні переваги в графіці перед конкурентами з ряду причин:

- проект виконаний на найновішому ігровому рушію Unreal Engine 5 від компанії Epic Games, що набагато поліпшує графіку в грі ніж на старих двійках, котрі використовують решта компаній. Плюсом нового рушія також є технічні спрощення під час розробки 3D моделей для гри, а також він дозволяє використовувати більше полікаунту для моделей при цьому зберігаються вимоги до технічного забезпечення пристроїв як на іграх старого покоління;
- ігровий рушії не вимагає плати за використання;
- гра, в котрій впроваджена розроблена модель буде першою в світі, котра використовує новий рушії Unreal Engine 5.

Розглянемо джерела економії, доходу, джерела фінансування.

Для компанії-розробника джерелом доходу є продаж продукту, реклама, вдосконалення контенту в грі з виходом оновлень гри та продаж за окрему плату доповнювання сюжету гри (DLS). Витрати фірми містять у собі витрати на розробку гри, зарплати. Джерелом фінансування є кошти від спонсорів Microsoft. Для компанії-розробника джерелом економії є заміна ручної праці машинною, що в значній мірі скорочує час розробки та грошові ресурси. Витрати студії складаються з витрат на розробку, зарплати розробникам, транспортування дисків для консолей по світу.

Розглянемо порядок розробки 3D моделі.

У загальному випадку розробка 3D моделі містить наступні етапи:

- початковий етап, на якому формується основні вимоги, пропонувані до моделі, описуються основні цілі й розробляється концепт;
- етап розробки мешу, де необхідно розробити форму моделі, створити розгортку для майбутніх текстур;
- етап запікання карт для генераторів під час створення текстур;

- етап видалення недоліків та редагування розроблених карт;
- етап текстурингу, найтворчий крок розробки;
- заключний етап, на якому проводиться остаточне тестування моделі в рушію, перевіряється коректність відображення текстур при освітленні гри.

Здійснимо розрахунок собівартості і ціни розробки 3D моделі.

У собівартість розробки 3D моделі входять наступні статті витрат:

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

Розробку 3D моделі проводять чотири фахівці: арт-директор, концепт-художник, 3D художник та тестувальник. Зарплата арт-директора становить 140 грн./год, концепт-художника – 102 грн./год, 3D художника – 127 грн./год, тестувальника – 113 грн./год. При цьому тривалість робочого дня кожного з них становить 8 годин. Модель розроблялась 11 днів.

Розрахунок основної заробітної плати наведено в таблиці 8.1.

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. Додаткова заробітна плата становить 20 % від основної:

$$1374 * 0,2 = 269 \text{ грн.}$$

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

$$1634 * 0,22 = 359,48 \text{ грн.}$$

До інших витрат слід віднести витрати на обслуговування ЕОМ і плату за електроенергію.

Таблиця 5.1 – Розрахунок витрат на заробітну плату

Етап	Вид робіт	Виконавець		Годинна ставка	Дні	Заробітна плата, грн.
		Кількість	Посада			
1. Початковий	Формулювання вимог до моделі	1	Арт-директор	140	0,5	70
2. Графічна частина	Розробка концепту моделі	1	Концепт-дизайнер	102	2	204
3. Розробка	Розробка 3D моделі	1	3D художник	127	6	762
4. Основний етап	Тестування компонентів	1	3D художник	127	0,5	64
	Комплексне тестування	1	Тестувальник	113	1	113
	Оформлення програмної документації	1	3D художник	127	0,5	64
5. Заключний етап	Корекція програмної документації	1	Арт-директор	140	0,5	70
Разом					11	1374
Додаткова заробітна плата (20 %)						269
Усього						1634

Витрати на електроенергію розраховуються виходячи зі споживаної потужності устрою і тарифу на електроенергію. У даному випадку передбачається використання 4 комп'ютерів з потужністю 1,3 кВт/год. Вартість однієї кВт/год електроенергії прийнято у розмірі 1,68 грн. Час використання електроенергії в процесі розробки:

$$11 * 8 = 88 \text{ годин}$$

Отже, плата за електроенергію складе:

$$3 * 1,3 * 1,68 * 88 = 576,58 \text{ грн.}$$

Витрати на обслуговування ЕОМ визначаються з вартості ЕОМ і часу її експлуатації, після закінчення якого, вона підлягає заміні (звичайно цей час не перевищує 3-х років), протягом року ЕОМ використовує 254 робочих дні.

$$(75000 / (5 * 8 * 254)) * 88 = 649,6 \text{ грн.}$$

Проект впроваджується на 1 компанію, тому собівартість розробки становить 3219,7 грн.:

$$1634 + 359,48 + 576,58 + 649,6 = 3219,7 \text{ грн.}$$

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

$$3219,7 * 0,3 = 965,9 \text{ грн.}$$

Розрахуємо ціну розробки моделі без податку на додану вартість (ПДВ):

$$3219,7 + 965,9 = 4185,6 \text{ грн.}$$

Розрахуємо суму ПДВ, вона складає 20% від ціни:

$$4185,6 * 0,2 = 837,12 \text{ грн.}$$

Розрахуємо вартість розробки з урахуванням ПДВ:

$$4185,6 + 837,12 = 5022,72 \text{ грн.}$$

Результати розрахунків наведено у таблиці 5.2.

Таблиця 5.2 – Розрахунок вартості розробки

Стаття витрат	Сума, грн.
Основна заробітна плата	1374
Додаткова заробітна плата	269
Єдиний соціальний внесок	359,48
Витрати на обслуговування ЕОМ	649,6
Витрати на електроенергію	576,68
Собівартість розробки	3219,7
Прибуток (рівень рентабельності 30 %)	965,9
Ціна без ПДВ	4185,6
Податок на додану вартість (ПДВ)	837,12
Ціна з урахуванням ПДВ	5022,72

Таким чином, виходячи з виконаних розрахунків, повна вартість розробки моделі складе 5022,72 грн. Термін виконання всіх етапів розробки становить 11 днів для команди з одного арт-директора, концепт-художника, 3D-художника і тестувальника. Очікувана сума прибутку складе 965,9 грн.

ВИСНОВКИ

Будь-яка відеогра починає свою розробку з ідеї та створення механік геймплею, але моделювання, зовнішній вигляд будь-яких об'єктів тривимірної графіки, дуже важливе. Розробники комп'ютерних ігор намагаються кожний рік створити якомога красивішу та реалістичну графіку з новітніми технологіями, котрі існують сьогодні в CG-індустрії. Графіка в грі – це перше на, що звертає увагу гравець, так як звертати увагу на візуальну складову набагато легше ніж розбиратись в технічних проблемах. Розробка відеоігор займає багато часу через великий об'єм роботи, тільки одна модель має декілька етапів створення перед тим як потрапити в гру.

Кожний етап розробки моделі дуже важливий, розробку 3D моделей можна зрівняти з карточним будинком. Якщо все рівно та правильно – будинок буде стояти, якщо хоч одна карта впаде – будинок розрушиться. Так сама і розробка моделей, кожний етап створення 3D моделі – відповідальна робота, кожний крок розробки пов'язано з наступним кроком.

У ході виконання бакалаврської роботи було розроблено ігрову модель згідно до сучасних вимог для AAA-відеоігор опираючись на стандартний пайплайн: від збору референсів до тестування готової моделі в ігровому рушію Unreal Engine 5. Вирішені такі проблеми як знаходження референсів аби вони відповідали вимогам до моделі, котра буде розробляти, вирішено проблему перевищення полікаунту Low – Poly моделі – це дозволить зменшити об'єм пам'яті, котра займає модель на опрацювання пристроєм, помилки в шейдингу моделі для запобігання градієнтів та артефактів на картах, потяжки на розгортці – необхідно уникати для коректного відображення текстур та карт на моделі, оптимізація текстур при експорті з Substance Painter (створено RMA текстуру) – текстури рафнес, металік та АО експортувались не на кожен окрему карту, а на одну текстури з трьома каналами RGB, де канал R-рафнес, G-металік, B-ao та вирішені проблеми значення текстур до вимог валідації з налаштуваннями шейдера у рушію.

Проведено тестування готової моделі в сучасному ігровому рушію Unreal Engine 5.

Був створений ID матеріал для текстур моделі, перевірено якість відображення текстур та карт на моделі у рушію, враховуючи деяку різницю в освітлені сцени ніж при розробці текстур. Вирішена проблема, як логічне знаходження різних елементів у текстурах – цей етап перевірки дуже сильно зв'язаний з якістю рефборду, чим правильніші будуть зібрані референси, тим краще буде логіка пошкоджень та передачі пошкоджень при створенні текстур моделі. Вирішені проблеми з потяжками мешу через триангуляцію. Навіть при гарному шейдингу моделі після триангуляції меш може мати потяжки через довгі та тонкі трикутники. Такі міста на моделі було відредаговано або видалено не впливаючи на фінальний вигляд моделі. Перед пакуванням розгортки було створено оверлап та відредагований Texel Density для заощадження місця на розгортці в координатах від 0 до 1. Перед експортом моделі в рушій було перевірено габарити моделі на відповідність до реальних розмірів об'єкту за допомогою створення примітивних форм з правильними даними висоти та ширини.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Grafting Physically-based materials. Unreal Engine Documentation. URL: [https://docs.unrealengine.com/4.26/en-US/RenderingAndGraphics/Materials/Physically Based/](https://docs.unrealengine.com/4.26/en-US/RenderingAndGraphics/Materials/Physically%20Based/) (дата звернення 10.06.2022).
2. Десять кращих ігрових рушіїв. URL: <https://senfil.net/index.php?newsid=321> (дата звернення 10.06.2022).
3. Lyashenko, V., & et al.. (2021). Wavelet ideology as a universal tool for data processing and analysis: some application examples. International Journal of Academic Information Systems Research (IJASIR), 5(9), 25-30.
4. Unreal Engine 4 Tutorial: rabota s Iskusstvenny`m Intellectom. URL: <https://stdpub.com/unrealengine/unreal-engine-4-tutorial-rabota-s-iskusstvennym-intellektom> (дата звернення 10.05.2022).
5. Матеріал по Unreal Engine Знайомство з движком і його основними функціями. URL: <https://habr.com/ru/post/344394> (дата звернення 15.05.2022).
6. Матеріал по Unreal Engine Blueprints та його функції. URL: <https://habr.com/ru/post/344446/> (дата звернення 15.05.2022).
7. Сіденко Л.А. Комп'ютерна графіка та геометричне моделювання: Навчальний посібник. Київ: ArtHuss, 2009. 224 с.
8. Deineko, Zh., & et al.. (2021). Color space image as a factor in the choice of its processing technology. Abstracts of I International scientific -practical conference «Problems of modern science and practice» (September 21 -24, 2021). Boston, USA, pp. 389-394.
9. Unreal Engine 4 Documentation. URL: <https://docs.unrealengine.com/en-US/index.html>. (дата звернення 15.05.2022).
10. Тьюторіал по Unreal Engine. 2017. URL: <https://habr.com/ru/post/344394/>. (дата звернення 15.05.2022).

11. Optimized View Frustum Culling Algorithms: Ulf Assarsson and Tomas Möller. URL: <http://www.cse.chalmers.se/~uffe/vfc.pdf>. (дата звернення 25.04.2022).

12. 3D Coat 2021-Volume #1 /Adam Gibson. Get Started I. URL: <https://www.liberty3d.com#2021/12/new-3d-coat-2021-volume-1-get-started-i/> (дата звернення: 12.05.2022).

13. Gundolf S. Igrý. Gejmdizajn. Issledovanie igr. Київ: Гуманитарний центр. 2021. 250 с.

14. Використання NURBS-моделювання для створення тривимірних об'єктів в 3D Studio Max. URL: http://esate.ru/uroki/3d-max/kurs_modelirovaniya/3D_max_urok_7_1/ (дата звернення 10.06.2020).

15. Lee J. Unreal Engine: Game Development from A to Z / J. Lee, J. P. Doran, N. Misra., 2016. 837 с.

16. Edmonds M. Mastering Game Development with Unreal Engine 4 / Matt Edmonds., 2018. 356 с. Home – 3DCoat Online Documentation.

17. Kovalyova D.F., Dejneko Zh.V. Optimizacziya 3D-modelej kak sposob umen`sheniya zagruzki na graficheskij prozessor // Поліграфічні, мультимедійні та web-технології: матеріали молодіжної школи-семінару V Міжнародної науково-технічної конференції. 2020. Т. 2. С. 27-30.

18. Гаманець Є.О., Дейнеко Ж.В. Особливості створення анімованого тривимірного персонажу // Поліграфічні, мультимедійні та web-технології: матеріали молодіжної школи-семінару IV Міжнародної науково-технічної конференції. 2021. Т 2. С. 76-78.

19. 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.

20. 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. 48 с.