

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 122 «Комп'ютерні науки»
(освітня програма «Комп'ютерна графіка та розробка ігор»)
на тему:

**«Ігровий додаток "Forbidden Forest" у жанрі Rogue-
lite. Проектування, розробка та тестування 3D
моделей»**

Виконав студент групи КН-20-2
Тіхонов Максим Олегович

Керівник роботи:
МАРЧУК Галина Вікторівна

Рецензент:
МАРЧУК Дмитро Костянтинівич

Житомир – 2024

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук
Марина ГРАФ
«25» січня 2024 р.

ЗАВДАННЯ
на кваліфікаційну роботу освітнього ступеня «бакалавр»
за спеціальністю 122 «Комп'ютерні науки»
(освітня програма «Комп'ютерна графіка та розробка ігор»)

Здобувач вищої освіти: **ТІХОНОВ Максим Олегович**
Керівник роботи: **МАРЧУК Галина Вікторівна**
Тема роботи: «Ігровий додаток "Forbidden Forest" у жанрі Rogue-lite. Проектування, розробка та тестування 3D моделей»,
затверджена наказом закладу вищої освіти від «25» січня 2024 р., №22/с
Термін здачі закінченої роботи: «30» червня 2024 р.
Вихідні дані роботи (зазначається предмет і об'єкт дослідження):

Консультанти кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Дата	
		Завдання видав	Завдання прийняв
1	Галина МАРЧУК	01.09.2023	01.11.2023
2	Галина МАРЧУК	01.11.2023	01.04.2024
3	Галина МАРЧУК	01.04.2024	01.05.2024

Календарний план

№ з/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка задачі. Пошук, огляд та аналіз аналогічних розробок. Формулювання технічного завдання. Опрацювання літературних джерел.	2 лютого 2024 – 5 березня 2024	Виконано
2	Проектування структури системи	6 березня 2024 – 25 березня 2024	Виконано
3	Написання програмного коду	26 березня 2024 – 26 квітня 2024	Виконано
4	Тестування системи	27 квітня 2024 – 14 травня 2024	Виконано
5	Оформлення пояснювальної записки	15 травня 2024 – 1 червня 2024	Виконано
6	Попередній захист	17 червня 2024	Виконано

Здобувач вищої освіти **Максим ТІХОНОВ**
Керівник **Галина Марчук**

РЕФЕРАТ

Кваліфікаційна робота включає в себе процес створення тривимірних моделей за допомогою комп'ютерних інструментів, а також супровідну пояснювальну записку. Пояснювальна записка містить у собі вступ, три розділи, підсумкові висновки, список використаних джерел та додатковий матеріал.

Текстова частина викладена на 60 сторінках друкованого тексту.

Пояснювальна записка має 22 сторінок додатків. Список використаних джерел містить 22 найменувань і займає 3 сторінки. В роботі наведено 46 рисунків та 4 таблиці. Загальний обсяг роботи – 82 сторінка.

Ключові слова: ГРА, UNITY, ZBRUSH, SUBSTANCE PAINTER, TEXTURE, ROGUE-LITE, 3D.

					КН.КР.Б – 122 – 24 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.	М. О. ТИХОНОВ				«Ігровий додаток "Forbidden Forest" у жанрі Rogue-lite. Проектування, розробка та тестування 3D моделей», Пояснювальна записка	Літ.	Арк.	Аркушів
Керівник	Г.В.МАРЧУК						3	82
Рецензент	Д.К.МАРЧУК					Житомирська політехніка, група КН-20-2		
Н. контр.								
Зав. каф.	М.С. ГРАФ							

ABSTRACT

The final qualification of the robot includes the process of creating 3D models using additional computer tools, as well as a supporting explanatory note. The explanatory note consists of an entry, three sections, pouches, a list of related items and additional material.

The text part is published on 60 pages of the original text.

The explanatory note contains 22 pages of supplements. The list of favorites contains 22 names and occupies 3 pages. The work contains 46 drawings and 4 table. Zagalny obsyag roboti – 82 pages.

Key words: GRA, UNITY, ZBRUSH, SUBSTANCE PAINTER, TEXTURE, ROGUE-LITE, 3D.

		<i>М. О. ТИХОМОВ</i>			КН.КР.Б – 122 – 24 – ПЗ	<i>Арк.</i>
		<i>Г.В.МАРЧУК</i>				
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>4</i>

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	6
ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ.....	8
1.1 Постановка задачі.....	8
1.2 Аналіз існуючого програмного забезпечення за тематикою кваліфікаційної роботи.....	8
1.3 Обґрунтування вибору інструментальних засобів.....	14
Висновки до першого розділу.....	17
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	18
2.1 Проектування розробки 3D моделей за допомогою діаграм UML.....	18
2.2 Розробка 3D моделей	25
2.3 Основні етапи створення текстур.....	37
2.4 Реалізація програмного кода	45
Висновки до другого розділу	50
РОЗДІЛ 3. ОЦІНКА ФУНКЦІОНАЛЬНОСТІ ТА НАДІЙНОСТІ МОДЕЛЕЙ.....	51
3.1 Тестування Моделі	51
3.2 Тестування UV розгортки	56
Висновки до третього розділу	57
ВИСНОВКИ	59
СПИСОК ВИКОРАСТАННИХ ДЖЕРЕЛ	60
ДОДАТКИ.....	63

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ПЗ – Програмне забезпечення

WIP - Work In Progress – малюнок або рендер незавершеної роботи

UI – User Interface – інтерфейс, елементи, з якими взаємодіє користувач

NPC - Non-Player Character – персонажі, якими гравець не управляє, та з якими зазвичай можна взаємодіяти.

		М. О. ТІХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВСТУП

Актуальність теми. Rogue-lite - це жанр відеоігор, що виник в середині 1980-х років, із відомою грою "Rogue" як одним з його перших представників. Основна особливість Rogue-lite полягає в тому, що вони зазвичай мають процедурно генеровані рівні, де кожна гра відрізняється від попередньої, і немає можливості зберігання прогресу. Це означає, що при поразці гравця потрібно починати гру спочатку.

Стиль та графіка в Rogue-lite можуть бути реалізовані у різних стилях і з використанням різних графічних рішень. Деякі Rogue-lite використовують піксельну графіку, яка нагадує класичні ігри 8-бітних та 16-бітних епох, в той час як інші можуть мати більш сучасний, реалістичний вигляд. Однак графіка зазвичай не є головною фокусною точкою у Rogue-lite, і нерідко може бути досить простою чи абстрактною [5].

Метою кваліфікаційної роботи є проектування, створення та тестування 3D моделей для гри "Forbidden Forest" у жанрі Rogue-lite.

Об'єктом дослідження є технологій створення 3D моделей для подальшого використання в відеоіграх.

Предметом дослідження є розробка 3D моделей для гри "Forbidden Forest" в жанрі Rogue-lite у спеціалізованих програмних забезпеченнях Zbrush для створення скульптури моделі, Blender, для їх оптимізації та Substance 3D Painter для їх текстуровання.

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				
Змн.	Арк.	№ докум.	Підпис	Дата		7

РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ

1.1 Постановка задачі

Основна мета розробки стилізованих моделей для гри в жанрі Rogue-lite [18] полягає в створенні візуального стилю, який ефективно передає атмосферу та настрої гри, підсилює іммерсію гравців і сприяє відтворенню унікального світу гри.

- Створення унікальної атмосфери. Стилiзованi моделi дозволяють розробникам створити унікальну атмосферу, яка характеризується своїм власним стилем та естетикою. Це допомагає поглибити іммерсію гравців і зробити гру більш привабливою.

- Підкреслення особливостей жанру. Графічний стиль стилізованих моделей може відображати особливості жанру Rogue-lite, такі як процедурно генеровані рівні, висока випадковість та постійна загроза смерті. Це дозволяє гравцям зануритися в унікальність гри та відчути її унікальний характер.

- Забезпечення легкості в сприйнятті. Стилiзованi моделi можуть бути спрощеними та абстрактними, що допомагає покращити сприйняття гри гравцями. Вони дозволяють швидко розпізнавати об'єкти та враження, що важливо для інтенсивного геймплею [17, 4].

- Створення цілісної візуальної концепції. Графічний стиль стилізованих моделей допомагає створити єдиний, цілісний візуальний світ гри, де всі елементи гармонійно поєднуються між собою. Це створює довершений вигляд гри та підсилює враження від геймплею.

1.2 Аналіз існуючого програмного забезпечення за тематикою кваліфікаційної роботи

Графіка в жанрі Rogue-lite включає в себе різноманітні аспекти, що варіюються від класичного піксель-арту до більш сучасних 3D об'єктів. Характеристики риси графіки у жанрі Rogue-lite для проєктів у 3D.

- Стилiзацiя. Хоча графіка в 3D може бути деталізованою, вона все

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				
Змн.	Арк.	№ докум.	Підпис	Дата		8

одно зберігає всій стилізований стиль, що є характерним для жанру Rogue-lite. Це допомагає зберегти унікальну атмосферу гри та зосередити увагу гравця на головному - на геймплейній динаміці та стратегії [16].

- Відтворення різних стилів. Графіка в Rogue-lite може відображати різноманітні стилі - від містики до наукової фантастики. Це дозволяє створювати унікальні світи з власною атмосферою та характером.

- Стилiзовані персонажі та монстри. В Rogue-lite персонажі та монстри часто мають стилізований вигляд, що може бути яскравим або темним, але завжди цікавим та виразним. Це дозволяє вирізнити їх на фоні локацій та робить їх легко впізнаваними [15].

Для аналізу було обрано три ігри. Borderlands 3, Rage 2 та The Outer Worlds, спираючись на їх графіку та жанр.

Borderlands 3 [1]. Графіка в Borderlands 3 вражає своєю виразністю та стильністю. Вона поєднує в собі комікський стиль з елементами кіберпанку та постапокаліптичного настрою. Кожен елемент гри виглядає як малюнок з графічного роману, що надає їй неповторності та оригінальності (рис.1.1).

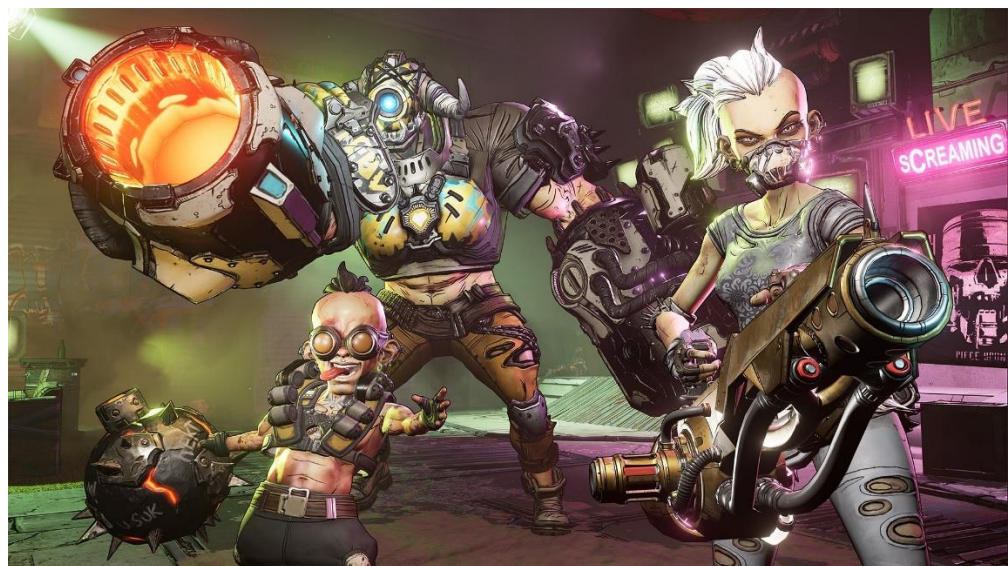


Рис.1.1 Borderlands 3 перший кадр вступного відео

Особливості графічного оформлення «Borderlands 3».

- Комікський стиль. Основним помітним елементом у графіці Borderlands 3 є комікський арт-стиль. Світ гри та персонажі виглядають так, ніби вони були намальовані рукою художника коміксів. Цей стиль дає грі унікальний

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				
Змн.	Арк.	№ докум.	Підпис	Дата		9

зовнішній вигляд, що запам'ятовується.

- Яскраві кольори та контури. Кольори у Borderlands 3 яскраві та насичені, що додає грі динамізму та енергії. Контури об'єктів та персонажів чіткі та виділені, що робить їх більш виразними у цьому коміксовому світі.

- Унікальний дизайн персонажів та зброї. Персонажі та зброя в Borderlands 3 мають унікальний та фантастичний дизайн, який поєднує в собі елементи наукової фантастики, фантастичних істот та психоделічних елементів.

- Ефекти та анімації. Borderlands 3 пропонує різноманітні спецефекти та динамічні анімації, які роблять бої та екшн у грі захоплюючими та видовищними. Вибухи, вогонь, електричні розряди та інші спецефекти створюють яскраве та вражаюче видовище.

- Деталізовані локації. Світи та локації в Borderlands 3 деталізовані та різноманітні. Від постапокаліптичних пусток до футуристичних міст та екзотичних планет – кожна локація у грі має свій унікальний стиль та атмосферу.

Недоліки графічного оформлення.

- Недоліки оптимізації. Деякі гравці відзначають проблеми з оптимізацією продуктивності на різних платформах, особливо на старих або менш потужних комп'ютерах. Це може виявлятися у ривках, падіннях кадрів під час гри або тривалих завантажувальних екранах.

- Текстури, що повторюються. У деяких випадках в Borderlands 3 можна помітити повторювані текстури на навколишніх об'єктах, що може трохи підірвати імєрсію і створювати відчуття одноманітності в деяких локаціях.

- Недостатня різноманітність анімацій. Незважаючи на те, що анімації персонажів і спецефекти в грі загальному вигляді вражають, деякі анімації можуть здаватися недостатньо різноманітними або повторюваними, особливо при тривалому проходженні гри.

- Проблеми з інтерфейсом. Інтерфейс гри, такий як меню та налаштування користувача, може бути трохи незручним або перевантаженим

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				10
Змн.	Арк.	№ докум.	Підпис	Дата		

для деяких гравців, особливо на ПК.

- Відсутність фізично-коректних ефектів. У Borderlands 3 використовується стилізований підхід до графіки, і це може викликати бажання деяких гравців побачити більш фізично коректні ефекти, такі як більш реалістичні фізичні реакції на руйнування навколишнього середовища або об'єктів.

Rage 2 [2]. Графіка в грі "Rage 2" є яскравим і динамічним візуальним досвідом, який поєднує в собі постапокаліптичний світ з кінематографічною естетикою та безкомпромісним екшеном. (рис.1.2).



Рис.1.2 Rage 2 показ геймплею

Особливості графічного оформлення «Rage 2».

- Постапокаліптичний стиль. Світ Rage 2 відображає постапокаліптичну пустелю, наповнену руїнами та небезпеками. Графічні елементи, такі як зруйновані будівлі, висушені краєвиди та агресивні мутанти, допомагають створити атмосферу хаосу та виживання.

- Яскраві та насичені кольори. "Rage 2" відрізняється використанням яскравих та насичених кольорів, що підкреслюють динамічність та драматизм ігрового світу. Це створює контраст з апокаліптичним оточенням і надає грі особливий візуальний стиль.

- Динамічні спец ефекти. Бойові сцени в "Rage 2" пожвавлюються динамічними спецефектами, такими як вибухи, ефекти світла та тіні, та різні

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				
Змн.	Арк.	№ докум.	Підпис	Дата		11

візуальні елементи, які роблять кожен бій епічним та захоплюючим.

- Деталізовані локації. Світи та локації в "Rage 2" відтворені з увагою до деталей, що додає глибину та реалізму ігрового світу. Кожна локація має свої унікальні особливості та атмосферу, відкриті для дослідження.

- Плавні анімації та ігровий процес. Анімації у грі "Rage 2" плавні та реалістичні, що робить персонажів та навколишній світ більш живими та реалістичними. Це важливо для створення занурюючого ігрового досвіду.

Недоліки графічного оформлення.

- Недостатня оптимізація. Деякі гравці зіткнулися з проблемами оптимізації, особливо на деяких комп'ютерах та консолях, що призводить до ривків, падіння кадрів та інших технічних проблем.

- Текстури та елементи, що повторюються. У деяких випадках в "Rage 2" можна помітити повторювані текстури та елементи оточення, що може створити відчуття одноманітності в різних локаціях.

- Недопрацьовані анімації. Деякі анімації персонажів та об'єктів можуть виглядати недопрацьованими або неприродними, що може відволікати від занурення в світ ігрового світу.

- Недостатня фізика оточуючого світу. Деякі гравці вказують на нереалістичну фізику оточення, таку як недостатнє руйнування об'єктів або дивну поведінку предметів при взаємодії з гравцем.

- Проблеми з освітленням та тінями. У деяких випадках у грі можуть виникати проблеми з освітленням та тінями, що може негативно позначатися на візуальній якості та атмосфері деяких сцен.

The Outer Worlds [3]. Графіка в грі "The Outer Worlds" розроблена студією Obsidian Entertainment і є унікальним і привабливим візуальним стилем, який поєднує в собі елементи наукової фантастики, комедії та атмосфери ретро-футуризму. (рис.1.3).

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				
Змн.	Арк.	№ докум.	Підпис	Дата		12



Рис.1.3 The Outer Worlds сцена демонстрації моба

Особливості графічного оформлення The Outer Worlds.

- Арт-стиль. Гра використовує стилізований арт-стиль, що нагадує комікси чи художні роботи середини ХХ століття. Це створює унікальну та впізнавану візуальну атмосферу.
- Колірна палітра. Графіка The Outer Worlds відрізняється насиченими та яскравими кольорами, які надають грі мальовничого та незабутнього вигляду. Кожна локація має свою унікальну колірну схему, що відображає її характер та атмосферу.
- Деталізовані локації. Світи та локації у грі "The Outer Worlds" деталізовані та різноманітні. Від космічних станцій та колоній до загадкових планет та спустошених ландшафтів, кожне місце у грі представлено з унікальними деталями та атмосферою.
- Дизайн персонажів та істот. Персонажі та істоти у "The Outer Worlds" мають оригінальний та характерний дизайн, який відображає їх характеристики та ролі в ігровому світі, що поєднує елементи наукової фантастики, футуризму і гумору.
- Ефекти та анімації. "The Outer Worlds" пропонує різноманітні спецефекти та динамічні анімації, які роблять ігровий світ більш живим та цікавим. Вибухи, вогонь, світлові ефекти та інші деталі створюють переконливу атмосферу гри.

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				
Змн.	Арк.	№ докум.	Підпис	Дата		13

Недоліки графічного оформлення The Outer Worlds.

- Технічні проблеми. Деякі гравці повідомляли про проблеми з оптимізацією гри, які можуть призводити до ривків, падіння кадрів або навіть зависань, особливо на менш потужних комп'ютерах або консолях.
- Недостатня деталізація навколишнього світу. У деяких випадках світ може здаватися недостатньо деталізованим або монотонним, особливо при дослідженні відкритих планет.
- Анімації та арт-активи, що повторюються. Деякі анімації та арт-активи можуть повторюватися, що може створити відчуття одноманітності під час гри та знизити імерсію.
- Недостатнє освітлення та тіні. У деяких сценах гри може бути недостатнє освітлення або проблеми з тінями, що впливає на візуальну якість та атмосферу.
- Графічний інтерфейс користувача. Деякі елементи інтерфейсу користувача, такі як інвентар або карти, можуть здаватися незручними у використанні або неоптимізованими для ігрового процесу.

На основі аналізу, було вирішено розробити власну гру в стилізованому стилі. Стилїзована графіка - це художній підхід до створення зображень, в якому форми, кольори та текстури спотворюються або перебільшуються для досягнення певного ефекту чи настрою. В розробці моделі під цей стиль буде використано Zbrush. Програмне забезпечення для моделювання в тривимірному просторі, яке відоме потужною системою скульптури і текстурування. Blender. Потужний інструмент, який надає можливість моделювати, анімувати, текстурирувати та рендерити. Substance Painter. Програмне забезпечення для текстурування тривимірних моделей, яке дозволяє художникам створювати реалістичні текстури для об'єктів.

1.3 Обґрунтування вибору інструментальних засобів

Популярні на даний час рушії для створення скульптур. Zbrush, Blender [19], та для створення текстур Substance Painter.

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				14
Змн.	Арк.	№ докум.	Підпис	Дата		

Blender плюси:

- Широкий функціонал для скульптування. Blender має набір інструментів для скульптування, що дозволяє створювати складні форми та додавати деталі до моделей.
- Інтеграція з іншими інструментами. Blender інтегрується з іншими функціональними компонентами, такими як моделювання, текстурування та анімація, що робить його зручним для повного циклу роботи над проектом.

Blender мінуси:

- Обмежені можливості скульптування. У порівнянні з ZBrush, Blender може бути менш потужним для складних скульптурних завдань через відсутність спеціалізованих інструментів та оптимізації.

Zbrush плюси:

- Професійний рівень скульптування. ZBrush вважається стандартом у галузі скульптування та має потужні інструменти для роботи з деталями.
- Динамічний інтерфейс для скульптування. Інтерфейс ZBrush спеціально розроблений для скульптування, що робить його дуже ефективним та приємним у використанні.

Zbrush мінуси:

- Вартість та ліцензування. ZBrush є комерційною програмою і вимагає придбання ліцензії, що може бути високим бар'єром для деяких користувачів.
- Відсутність інтеграції з іншими програмами. При використанні ZBrush, може знадобитися інтеграція з іншими програмами для завершення проекту, що може бути не так зручно, як працювати всередині одного середовища, як у Blender.

Розглядаючи всі аспекти та переваги обох програм для створення скульптур, можна зробити висновок, що обрати ZBrush було найкращим вибором. Перш за все, ZBrush спеціалізується на скульптуванні та має вражаючий набір інструментів для цієї задачі. Його потужність та можливості

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				
Змн.	Арк.	№ докум.	Підпис	Дата		15

дозволяють створювати деталізовані та реалістичні скульптури з вражаючою точністю.

Крім того, інтерфейс ZBrush дуже інтуїтивний для скульптурування, що робить процес створення скульптур більш приємним та продуктивним. Це дозволяє мені краще зосередитися на моїй творчості та виразності, замість того, щоб витратити час на навчання складних інструментів.

Отже, розглядаючи усі переваги та недоліки, при виборі ZBrush, можна максимально ефективно реалізувати творчі ідеї і створити вражаючі скульптури з високою деталізацією та рівнем реалізму.

Substance Painter плюси:

- Real-time rendering. Substance Painter має вражаючий режим попереднього перегляду в реальному часі, що дозволяє користувачам бачити зміни у текстурі миттєво та швидко налаштовувати ефекти та шари.
- Унікальні інструменти. Substance Painter пропонує розумні матеріали, генератори шарів та інші унікальні інструменти, які спрощують та прискорюють процес створення складних текстурних ефектів.
- Інтеграція з іншими програмами. Легка інтеграція Substance Painter з іншими програмами, такими як Maya, Blender, Unity та іншими, забезпечує зручну роботу з текстурами у різних проектах та платформах.

Substance Painter мінуси:

- Вартість ліцензії. Одним з головних недоліків Substance Painter є висока вартість ліцензії, що може бути недоцільним для деяких користувачів, особливо для початківців або невеликих студій.

3Dcoat плюси:

- Повний цикл роботи. 3Dcoat пропонує повний цикл роботи з текстурами, починаючи від їх створення та редагування до аплікації на моделі та експорту.
- Інтуїтивний інтерфейс. Простий та інтуїтивний інтерфейс 3Dcoat робить процес створення текстур приємним та доступним для користувачів будь-якого рівня.

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				
Змн.	Арк.	№ докум.	Підпис	Дата		16

3Dcoat мінуси:

- Обмежені можливості. 3DCoat може бути менш потужним у порівнянні з Substance Painter у термінах унікальних інструментів та ефектів для створення текстур.
- Відсутність режиму реального часу. Відсутність режиму реального часу у 3DCoat може збільшити час, необхідний для попереднього перегляду та внесення змін у текстури.

Розглядаючи всі аспекти та переваги обох програм для створення текстур, обрання Substance Painter є найбільш доцільним вибором. У цій програмі вражаючий набір інструментів та ефектів, що дозволяє створювати деталізовані та реалістичні текстури для будь-яких проектів. Режим попереднього перегляду в реальному часі дозволяє оперативно оцінювати результати змін та вносити корективи на льоту. Інтерфейс Substance Painter дуже інтуїтивний, що полегшує процес створення текстур і дозволяє зосередитися на творчості. Таким чином, на основі всіх плюсів та мінусів, обрано Substance Painter замість 3DCoat для створення текстур, оскільки вона надає більше можливостей для створення високоякісних та деталізованих текстур, а також забезпечує зручний та продуктивний робочий процес.

Висновки до першого розділу

Визначено мету розробки та поставлено завдання. Переглянуто три популярні гри жанрів Rogue-lite, визначено їх переваги та недоліки. Проаналізовано графіку в обранному жанрі, його характерні риси для відтворення у грі.

Обґрунтовано вибір інструментарію (Zbrush, Blender, та Substance Painter), розглянувши їх переваги, що будуть в нагоді для проекту.

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				17
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Проектування розробки 3D моделей за допомогою діаграм UML

Проектування розробки 3D-моделей є складним процесом, що потребує ретельного планування та координації. Для ефективного управління та візуалізації цього процесу використовуються діаграми UML (Unified Modeling Language).

Діаграма варіантів використання (Use Case Diagram) надає вичерпне уявлення про процес створення та інтеграції текстур для ігрових моделей, показуючи взаємодію між різними учасниками процесу та етапами роботи (рис.2.1).

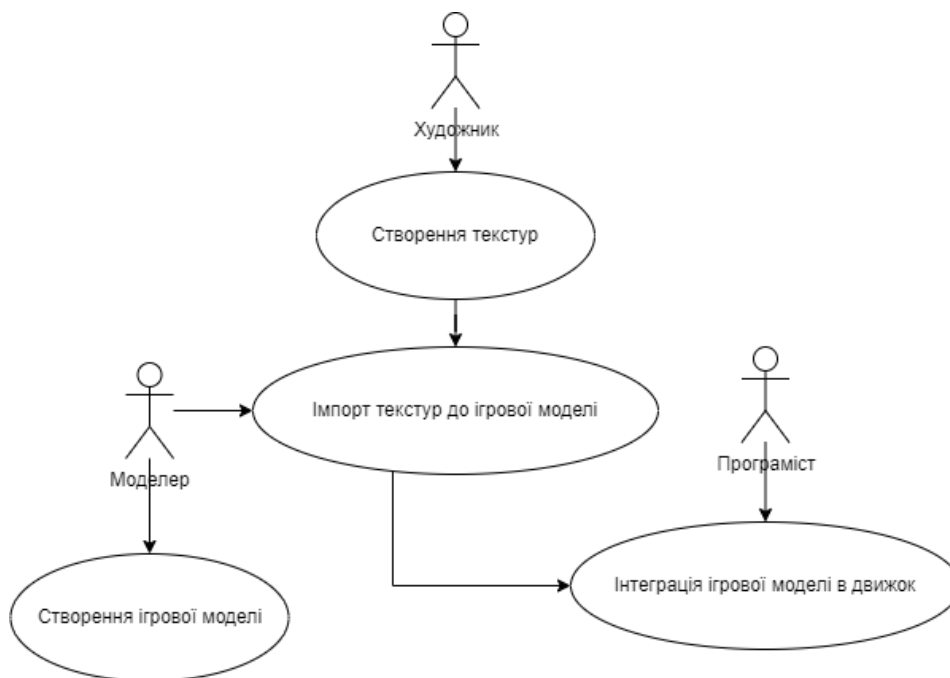


Рис.2.1 – Діаграма варіантів використання створення та інтеграція текстур для ігрової моделі

Таблиця 2.1 описує етапи процесу створення та інтеграції текстур для 3D-моделей включаючи ролі кожного учасника (акторів) і їхні завдання.

Таблиця 2.1

Створення та інтеграція текстур для 3D-моделей у ігровий рушій

Виконавець	Випадок використання	Опис	Залежності
Моделер	Створення ігрової моделі	Моделер створює 3D-модель для гри.	Немає
Художник	Створення текстур	Художник створює текстури для ігрової моделі.	Немає
Моделер	Імпорт текстур до ігрової моделі	Моделер імпортує текстури, створені художником, до моделі.	Включає створення текстур
Програміст	Інтеграція ігрової моделі в движок	Програміст інтегрує готову модель з текстурами в ігровий движок.	Включає імпорт текстур

Діаграма послідовності (рис. 2.2) описує процес створення ігрової моделі, починаючи від збору вимог і закінчуючи передачею моделі в ігровий проєкт. Вона охоплює етапи збір вимог, розробку концепції, моделювання, текстурування, та анімацію, відображаючи взаємодію між різними об'єктами у процесі розробки.

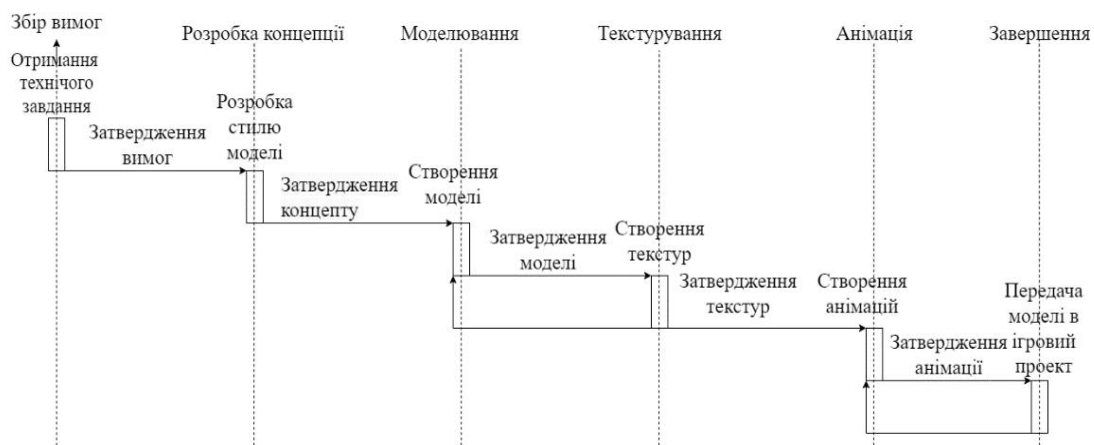


Рисунок.2.2 – Діаграма послідовності етапи розробки ігрової моделі

Таблиця 2.2 описує процес створення моделі при розробці гри, розділений на ключові етапи, кожен з яких включає певні завдання.

Процес створення ігрової моделі

Етап	Завдання	Опис
Збір вимог	Отримання технічного завдання	Збір і документування всіх технічних вимог до моделі.
	Затвердження вимог	Підтвердження всіх зібраних вимог з командою або клієнтом.
Розробка концепції	Розробка етапу моделі	Створення первісних ескізів і концептуальних ідей для моделі.
	Затвердження концепту	Погодження та затвердження концептуальних ескізів з командою або клієнтом.
Моделювання	Створення моделі	Власне процес 3D-моделювання, створення базової геометрії моделі.
	Затвердження моделі	Перевірка та затвердження створеної моделі на відповідність вимогам та концепту.
Текстурування	Створення текстур	Створення та застосування текстур до моделі.
	Затвердження текстур	Перевірка та затвердження текстур, переконання, що вони відповідають очікуванням і технічним вимогам.
Анімація	Створення анімації	Розробка анімацій для моделі, таких як рухи, дії та інші анімаційні ефекти.
	Затвердження анімації	Перевірка та затвердження анімацій з командою або клієнтом.

Завершення	Передача моделі в ігровий проект	Останній етап, на якому затверджена модель разом з текстурами і анімаціями інтегрується в ігровий проект для подальшого використання.
------------	----------------------------------	---

Діаграма станів (рис.2.3) відображає етапи створення скульптури моделі, включаючи основні стани та переходи між ними. Кожен етап деталізує процес створення та удосконалення моделі.

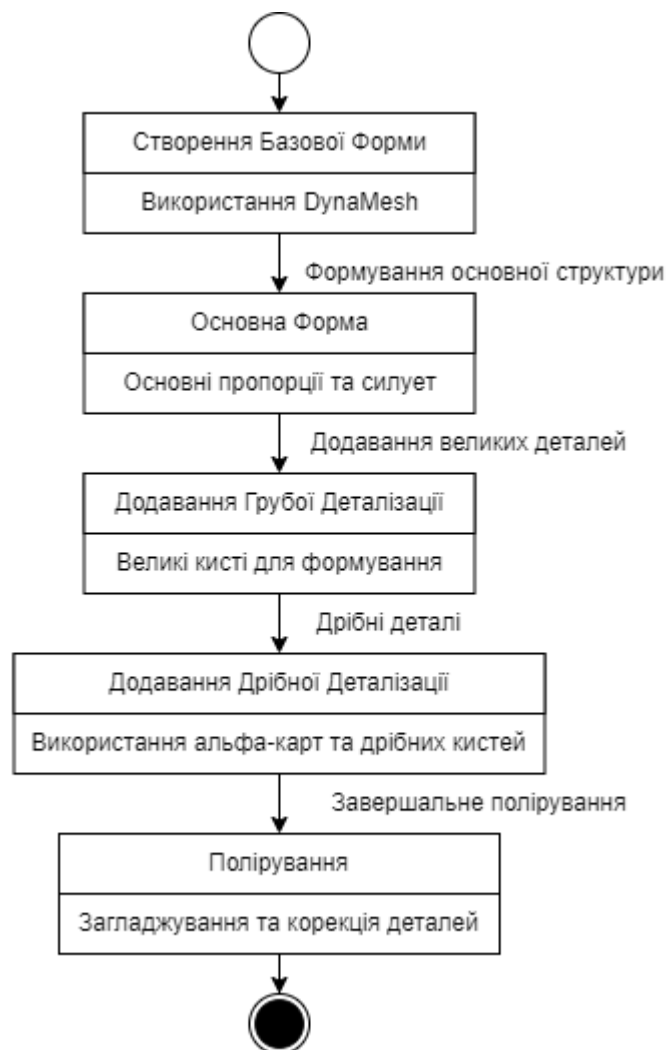


Рис.2.3 – Діаграма станів створення скульптури моделі

Таблиця 2.3 допомагає структурувати процес створення базової форми моделі, дозволяючи чітко слідкувати за кожним етапом роботи .

Таблиця 2.3

Процес створення базової форми моделі

Стан	Дія	Опис
Використання DynaMesh	Початкове створення базової форми моделі за допомогою технології DynaMesh	Використовується для швидкого створення основної форми без деталей, яка стане основою для подальшої роботи.
Основна Форма	Визначення основних пропорцій та силуету моделі	На цьому етапі встановлюються ключові пропорції моделі, забезпечуючи правильну анатомію та загальний вигляд.
Додавання Грубої Деталізації	Використання великих кистей для формування крупних деталей	На цьому етапі додаються великі деталі, такі як м'язи, основні форми обличчя та інші великі елементи.
Додавання Дрібної Деталізації	Використання альфа-карт та дрібних кистей	Додавання більш детальних елементів, таких як текстура шкіри, дрібні складки та інші дрібні деталі, що додають реалістичності моделі.
Полірування	Загладжування та корекція деталей	Останній етап, на якому виконується полірування моделі для усунення недоліків та забезпечення гладкості поверхні.

Діаграма активності (рис.2.4) допомагає зрозуміти загальний робочий процес та визначити послідовність дій, які потрібно виконати для досягнення мети

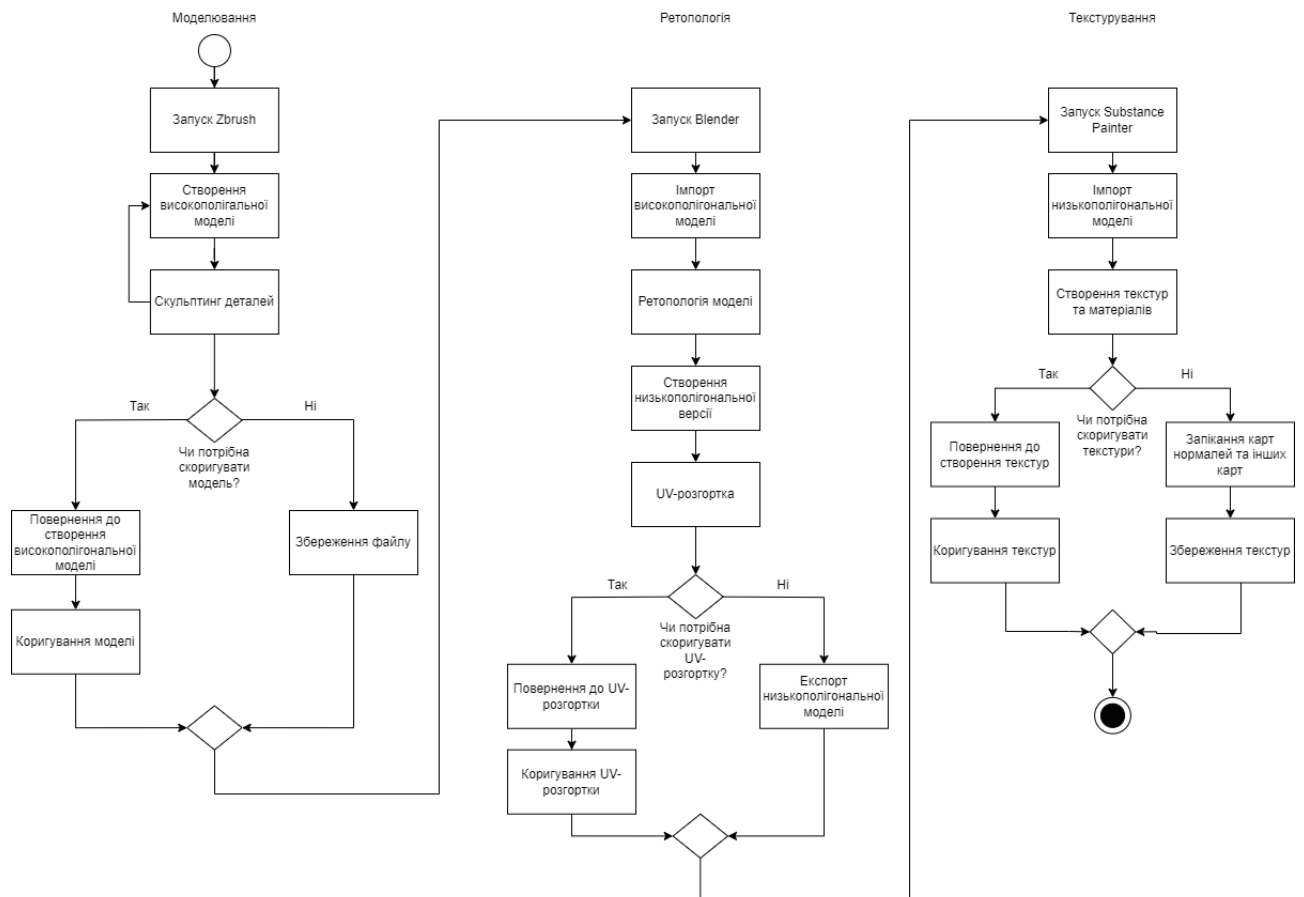


Рис.2.4 – Діаграма активності етапи розробки моделі

Таблиця 2.4 систематизує процес створення 3D-моделі, включаючи моделювання, ретопологію та текстурування, описуючи кожен крок та його мету.

Таблиця 2.4

Систематизація процесу створення 3D-моделі

Етап	Дія	Опис	Розгалуження
Моделювання	Запуск ZBrush	Починаємо процес моделювання, відкривши програму ZBrush.	
	Створення високополігональної моделі	Створення моделі з високою деталізацією.	
	Скульптинг деталей	Додавання дрібних деталей до моделі.	

Продовження Таблиці 2.4

	Розгалуження: Чи потрібно скоригувати модель?	Якщо модель потребує коригування, повертаємося до етапу створення високополігональної моделі і коригуємо її. Якщо коригування не потрібно, зберігаємо файл.	Так: Повернення до створення високополігональної моделі і коригування моделі. Ні: Збереження файлу.
Ретопологія	Запуск Blender	Починаємо процес ретопології, відкривши програму Blender.	
	Імпорт високополігональної моделі	Імпортуємо створену високополігональну модель в Blender.	
	Ретопологія моделі	Оптимізація моделі для зменшення кількості полігонів.	
	Створення низькополігональної версії	Створення версії моделі з меншою кількістю полігонів.	
	UV-розгортка	Створення UV-мапи для текстурування моделі.	
	Розгалуження: Чи потрібно скоригувати UV-розгортку?	Якщо UV-розгортка потребує коригування, повертаємося до етапу UV-розгортки і коригуємо її. Якщо коригування не потрібно, експортуємо низькополігональну модель.	Так: Повернення до UV-розгортки і коригування UV-розгортки. Ні: Експорт низькополігональної моделі.

Текстурування	Запуск Substance Painter	Починаємо процес текстурування, відкривши програму Substance Painter.	
	Імпорт низькополігональної моделі	Імпорт низькополігональної моделі, створеної на попередньому етапі, у Blender.	
	Створення текстур та матеріалів	Розробка та нанесення текстур і матеріалів на модель для додання їй реалістичного вигляду.	
	Розгалуження: Чи потрібно скоригувати текстури?	Якщо текстури потребують коригування, повертаємося до етапу створення текстур і коригуємо їх. Якщо коригування не потрібно, запікаємо карти нормалей та інші карти, а потім зберігаємо текстури.	Так: Повернення до створення текстур і коригування текстур. Ні: Запікання карт нормалей та інших карт, збереження текстур.
	Збереження текстур	Збереження готових текстур для використання у фінальній версії 3D-моделі.	

2.2 Розробка 3D моделей

Процес розробки 3D моделі включає кілька етапів, що дозволяють створити віртуальний об'єкт з тривимірною геометрією.

Загальний список пунктів для створення моделей на прикладі персонажа.

Перший етап - створення скульптури. Після того, як концепт-художник визначив основні деталі персонажа, скульптори переходять до створення 3D-моделі у спеціалізованому програмному забезпеченні, найкращою програмою для цього є Zbrush [6]. В ньому можна використовувати різні інструменти та техніки для створення різних деталей, таких як форми тіла, текстури, м'язи, а також відділення і структурування окремих частин персонажа на високому рівні.

Для початку створюється базова форма, а саме Blocking. Це початковий етап у процесі 3D-моделювання, під час якого художник створює основні форми та пропорції майбутнього об'єкта чи персонажа. Цей етап фокусується на загальних об'ємах і силуетах, без деталізації. Мета полягає в тому, щоб визначити правильні пропорції та основні форми перед тим, як перейти до детальної роботи. В даному випадку найкраще підійде інструмент Zspheres. Це інструмент в програмі ZBrush, який використовується для створення базових форм і каркасів майбутніх 3D-моделей. Він дозволяє швидко і легко створювати структуру об'єкта за допомогою сфер, які можна комбінувати, масштабувати і з'єднувати між собою. Використання ZSphere забезпечує високу гнучкість і швидкість при створенні базової топології персонажів або інших складних об'єктів (рис.2.5).



Рис.2.5. Zsphere из меню append

Додано першу сферу в центрі робочої області. Натискаючи на сферу витягується додаткових ще 2 сфери за умовою якщо включин режим тігго. Увімкнувши тігго (натискання клавіші X) для роботи над обома сторонами моделі одночасно (рис.2.6).

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				
Змн.	Арк.	№ докум.	Підпис	Дата		26

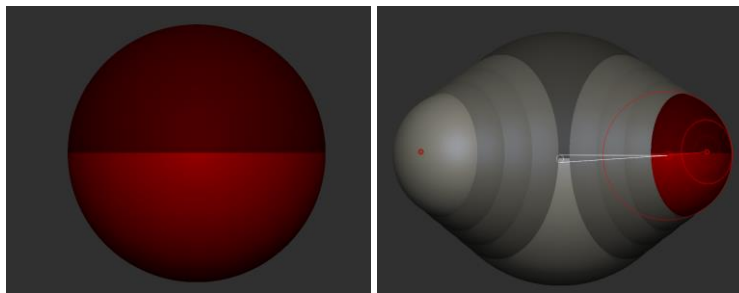


Рис.2.6. Додавання сфер

Налаштування позицій та розмірів.

Використовуючи інструменти переміщення (Move), обертання (Rotate) та масштабування (Scale), щоб налаштувати позиції і розміри сфер (рис.2.7).

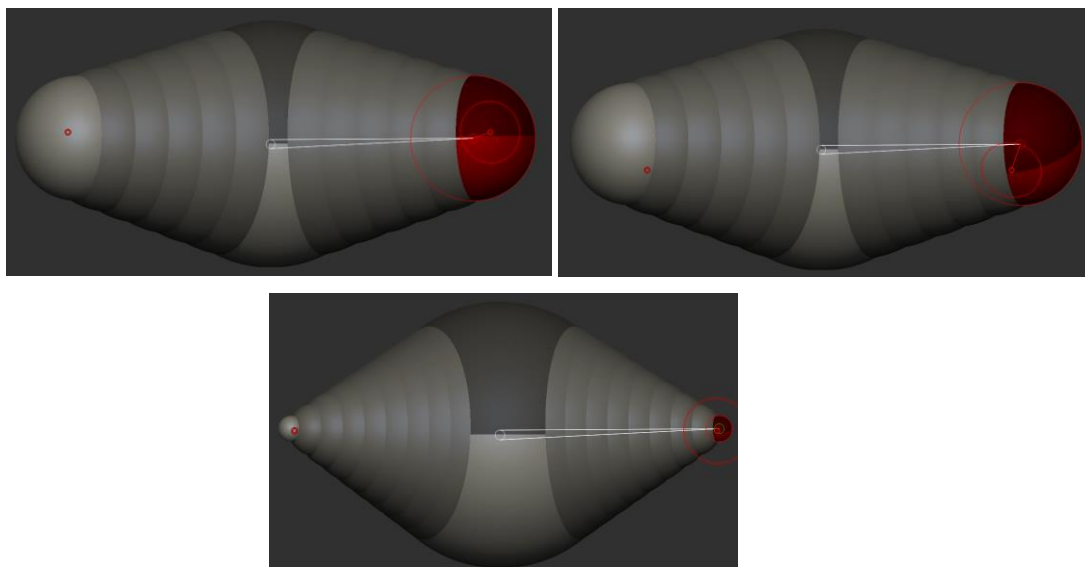


Рис.2.7. Використання Move, Rotate, Scale

Після завершення створення каркасу, він перетворюється в Adaptive Skin, перейшовши в меню Tool - Adaptive Skin і натиснувши кнопку Make Adaptive Skin (рис.2.8).

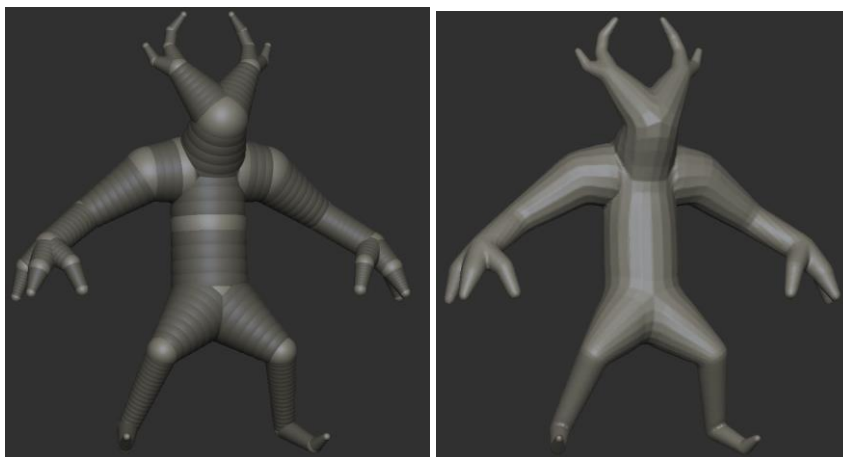


Рис.2.8. Blocking персонажа

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				27
Змн.	Арк.	№ докум.	Підпис	Дата		

Далі для продовження над роботою проведемо додаткові налаштування, а саме застосуємо динамічну топології DynaMesh. Це потужний інструмент для скульптингу, який дозволяє динамічно оновлювати топологію 3D-моделі під час роботи. Ця технологія є однією з ключових функцій ZBrush, що дозволяє художникам зосередитися на творчості, а не на технічних аспектах моделювання. Використовуємо DynaMesh для перетопологізації моделі при необхідності додання нових деталей або коригування форми, після його використання йдуть наступні етапи.

1. *Робота з об'ємами та формами.* Починається робота з великих форм і поступово переходиться до дрібних деталей. Використовується кисть Move (рис.2.9). Це інструмент для скульптингу який дозволяє переміщати великі частини сітки моделі. Його використовують для грубого моделювання форми або зміни пропорцій, та Inflate (рис.2.9). Це інструмент для скульптингу який дозволяє збільшувати об'єм поверхні моделі, наче надуваючи її для коригування загальних об'ємів і форм.

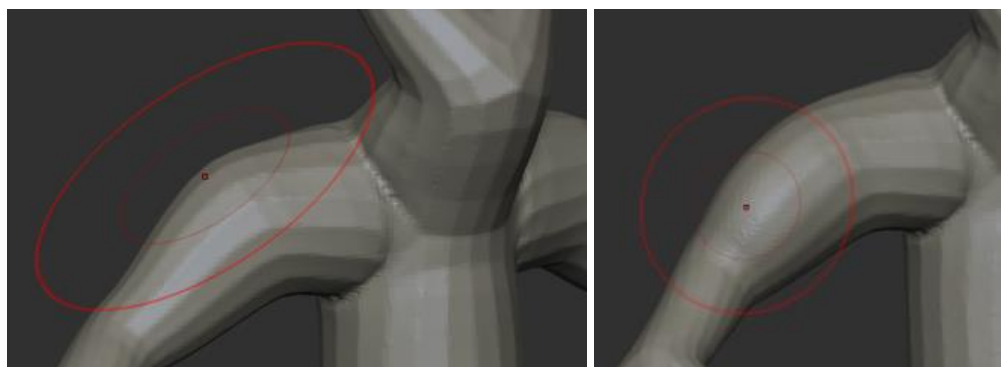


Рис.2.9. Використання Move та Inflate

2. *Додання основних деталей.* Використовується кисть ClayBuildup (рис.2.10) - це інструмент для скульптингу, який імітує накладання глини шарами на поверхню моделі. Він часто використовується для додавання великих деталей і грубої форми для створення м'язів, кісток і основних деталей анатомії. Згладжування Використається кисть Smooth(утримуючи Shift) (рис.2.10) - це інструмент для скульптингу який використовується для згладжування поверхні моделі. Він допомагає зробити поверхню більш

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				
Змн.	Арк.	№ докум.	Підпис	Дата		28

гладкою, зменшити різкість деталей і усунути небажані артефакти для згладжування нерівностей і переходів між різними частинами моделі.

3. *Деталізація.* Після завершення основних форм і об'ємів починається додавання дрібних деталі, такі як зморшки, складки шкіри, пори тощо. Використовується кисть *Dam_Standard* (рис.2.10)- це інструмент для скульптингу який використовується для створення чітких, різких ліній та канавок на поверхні моделі. Він дозволяє додавати тонкі, глибокі деталі, такі як зморшки або шви, частіше за все використовується для створення чітких ліній і різких переходів.

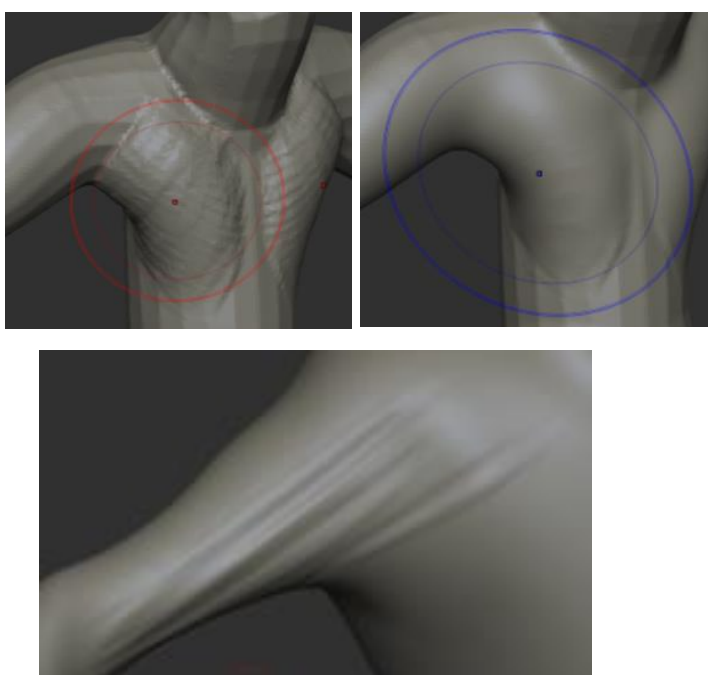


Рис.2.10. Використання ClayBuildup, Smooth та Dam_Standard

4. *Використання альфа-кісточок* (рис.2.11). Це інструмент, який використовуються для створення текстур і деталей на поверхні моделі шляхом застосування двовимірних зображень (альфа-каналів) до тривимірної поверхні. Вони дозволяють швидко і легко додавати складні текстури, візерунки та деталі, які було б важко створити вручну, використовуємо їх для додання текстури шкіри, тканини та інших дрібних деталей.

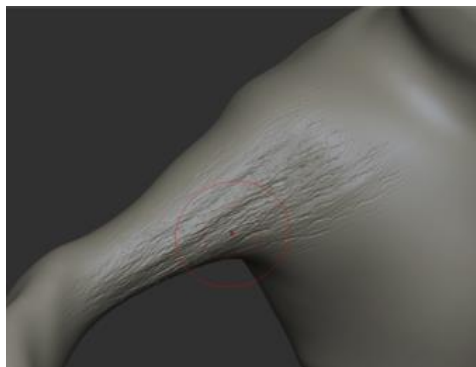


Рис.2.11. Використання Альфа-кісточок

Використовуючи всі продемонстровані інструменти можна створити готового до ретопології персонажа (рис.2.12).



Рис.2.12. Модель дерев'яного жука

Другий етап - створення ретопології. Після створення скульптури настає етап ретопології, коли модель оптимізується для оптимальної роботи в грі шляхом накладання поверх високодеталізованої моделі низькополігональну топологію, це допомагає забезпечити гладкість рухів персонажа та ефективну роботу гри без зайвого навантаження на обчислювальні ресурси.

1. Експорт моделі з Zbrush. Позавершенню скульптингу потрібно переконатися, що наша скульптура в завершена і всі деталі на місці.

2. Decimation Master. Якщо модель дуже високополігональна, можна використовувати Decimation Master. Це плагін, призначений для зменшення кількості полігонів у моделі, зберігаючи при цьому її деталізацію.

3. Ініціалізація Decimation Master. Відкрити плагін Decimation Master через меню Zplugin, натиснути кнопку Pre-process Current (для активного

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				30
Змн.	Арк.	№ докум.	Підпис	Дата		

субтула) або Pre-process All (для всіх субтулів) для передобчислення даних моделі, вказати бажаний відсоток зменшення полігонів у полі Percentage of Decimation або встановити конкретну кількість полігонів, натискаєм кнопку Decimate Current (для активного субтула) або Decimate All (для всіх субтулів) для зменшення кількості полігонів, після чого отримуємо готову тріангульовану модель (рис. 2.13).



Рис. 2.13. Результат виконання плагіна Decimation Master

4. Експорт. Відкрити плагін FBX ExportImport через меню Zplugin і вибрати тип відправки файлу, наприклад, Selected або Visible. Натискається Export для збереження файлу на комп'ютері.

5. Імпорт моделі в Blender. Для початку відкривається Blender [8] і створюється нова сцена, після чого натискається File - Import і оберається відповідний формат на прикладі FBX.

Організація сцени. Для високої якості текстури, модель повинна мати правильний масштаб та позицію відповідно до віртуальних одиниць виміру в Blender.

Наприклад, якщо іде робота з людським персонажем, висота моделі повинна бути в середньому приблизно 1.8 метра. Позиція модель повинна бути центром на сцені, з координатами (0, 0, 0) на осі X, Y і Z. Перевірка масштабу, виконується за допомогою інструментів трансформації (G для переміщення, S для масштабу) та налаштування координат у вкладці Item для коригування масштабу та позиції моделі. Використовуються одиниці виміру метри для точності. Якщо модель не відповідає правильному масштабу, виділяється модель та натискається Ctrl + A, в спливаючому меню оберається All Transforms, після чого модель виправляється на коректним масштаб (рис.2.14).



Рис. 2.14. Виправлення масштабу та позиції

6. Створення нової сітки. Створюється новий об'єкт-меш Shift + A та вибирається Plane як базу для ретопології.

7. Після виправлення масштабу та позиції моделі застосовується модифікатор Shrinkwrap для полегшення процесу ретопології, щоб нові полігони прилягали до поверхні скульпта. Основні налаштування модифікатора Shrinkwrap включають.

Target (Ціль). Об'єкт, до якого буде прив'язаний модифікований об'єкт. Це основний параметр, що визначає поверхню для прикріплення.

Wrap Method (Метод обгортання). Можливі методи включають "Nearest Surface Point" (найближча точка поверхні), "Projection" (проекція) та "Nearest Vertex" (найближча вершина). Вибір методу залежить від завдань модифікації.

Offset (Зміщення). Дозволяє встановити відстань між модифікованим об'єктом та поверхнею цілі. Це корисно для уникнення перетину об'єктів або для створення певного зазору між ними.

Shrinkwrap працює за алгоритмом The Iterative Closest Point (ICP) [21]. Використовується для вирівнювання двох 3D точкових хмар або поверхонь.

$$E(R, t) = \sum_i \| R p_i + t - q_i \|^2$$

де: R – обертання; t – трансляції; i - позначення поточної точки.

p_i - точка з першої хмари. q_i - найближча точка у другій хмарі.

$\| p_i - q_i \|$ - відстань між точками

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				
Змн.	Арк.	№ докум.	Підпис	Дата		32

Алгоритм роботи

Initialization (Ініціалізація). Встановити початкові наближення трансформації (R, t).

Point Matching (Відповідність точок). Для кожної точки з першої хмари (source point cloud). Це набір 3D точок, які потрібно вирівняти або співставити з іншою хмарою точок. Вона також відома як початкова або вихідна хмара. знайти найближчу точку в другій хмарі(target point cloud). Це набір 3D точок, до яких потрібно вирівняти першу хмару. Ця хмара служить опорою або орієнтиром для вирівнювання).

Weighting (Вага). Можливе призначення ваги кожній парі точок, щоб впливати на результат вирівнювання.

Outlier Rejection (Відкидання викидів). Виключити пари точок, що є викидами (не відповідають загальній структурі).

Error Minimization (Мінімізація помилки). Знайти оптимальну трансформацію (R, t), що мінімізує суму квадратів відстаней між відповідними точками.

Функція помилки

Iteration (Ітерація): Повторювати кроки 2-5 до досягнення збіжності (зменшення зміни в помилці нижче певного порогу).

Для швидкості та зручності також використовується модифікатор Mirror для відзеркалення моделі. Основні налаштування модифікатора Mirror включають.

Axes (Осі). Вибір осей, відносно яких буде здійснено дзеркальне відображення. Можливі варіанти. X, Y, Z. Можна вибрати одну або декілька осей.

Clipping (Обрізання). Ця опція забезпечує, що вершини, які знаходяться на дзеркальній осі, зливаються і залишаються на місці відображення. Це корисно для уникнення небажаних зазорів або перекриттів у центрі симетрії.

Merge (Злиття). Ця опція автоматично зливає вершини, що знаходяться на певній відстані одна від одної. Параметр Merge Limit визначає максимальну

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				
Змн.	Арк.	№ докум.	Підпис	Дата		33

відстань між вершинами, які будуть злиті. Це важливо для забезпечення цілісності та безперервності поверхні моделі (рис.2.15).

8. Активізація режиму редагування. Режим редагування (Edit Mode) в Blender [9, 10] є ключовим для роботи з геометрією об'єктів. Він надає широкі можливості для створення та модифікації тривимірних моделей. Основні аспекти режиму редагування включають.

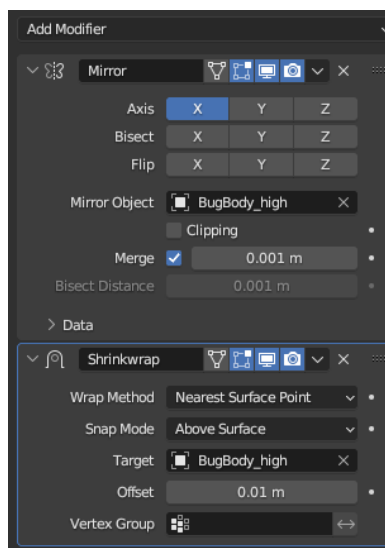


Рис. 2.15. Додавання модифікаторів Shrinkwrap та Mirror

Вибір компонентів. У режимі редагування користувач може вибирати та маніпулювати різними компонентами об'єкта.

1. Вершини (Vertices). Основні точки, з яких складається геометрія об'єкта. Вибір і редагування вершин дозволяє змінювати форму об'єкта на базовому рівні.

2. Ребра (Edges). Лінії, що з'єднують вершини. Вибір ребер дає можливість змінювати контури об'єкта, видаляти, з'єднувати або розрізати його частини.

3. Грани (Faces). Площини, утворені з кількох ребер. Вибір і редагування граней дозволяє змінювати поверхню об'єкта, створювати нові грані або видаляти існуючі.

Інструменти редагування

Blender надає широкий набір інструментів для редагування геометрії, які дозволяють виконувати різноманітні операції.

1. Трансформації.
 - Переміщення (Move). Зміщення компонентів у просторі.
 - Обертання (Rotate). Поворот компонентів навколо осей.
 - Масштабування (Scale). Зміна розмірів компонентів.
2. Екструзія (Extrude). Створення нових геометричних елементів шляхом витягування існуючих вершин, ребер або граней.
3. Вставка (Inset). Створення нових граней всередині вибраних, що дозволяє деталізувати модель.
4. Розрізання (Knife Tool). Створення нових геометричних розрізів на поверхні об'єкта.
5. З'єднання (Bridge Edge Loops). З'єднання кількох петель ребер, що дозволяє створювати складніші структури.

Модифікація топології

Режим редагування також дозволяє змінювати топологію об'єкта, що важливо для підготовки моделі до подальших етапів, таких як анімація або текстуркування. До основних операцій з топологією належать.

1. Злиття (Merge). Об'єднання кількох вершин в одну.
2. Розділення (Split). Відокремлення компонентів для створення окремих частин об'єкта.
3. Видалення (Delete). Видалення вибраних вершин, ребер або граней.

Переходим в режим редагування натиснувши клавішу Tab і починаєм створювати нову сітку поверх скульптури (рис.2.16)..

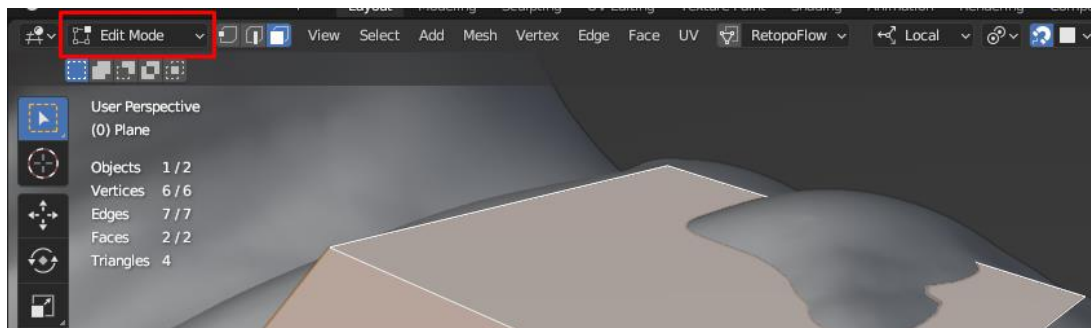


Рис. 2.16. Режим редагування

9. Інструменти ретопології. Використовується такі інструменти як Bsurfaces. Аддон який значно спрощує процес створення нових сіток і

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				
Змн.	Арк.	№ докум.	Підпис	Дата		35

ретонології. Він дозволяє створювати поверхні на основі кривих і нарисованих ліній, що робить процес створення топології швидшим та інтуїтивно зрозумілим. F2. Аддон який полегшує створення і редагування полігональних сіток. Він дозволяє швидко додавати нові грані до сітки, що робить процес ретонології більш ефективним. LoopTools. Аддон який надає набір інструментів для роботи з петлями (loops) і кривими (curves), що значно полегшує процес моделювання та ретонології.

Використовуючи всі продемонстровані інструменти створюється готовий до наложення uv швов модель персонажа [12] (рис.2.17).



Рис. 2.17. Завершенна ретонологія моделі

Третій етап - створення UV розгортки [22]

1. Перехід в режим редагування, вибарається режим виділення select edge(клавіша 2). Виділяється потрібні edge з зажатою клавішою shift, на місцях виділених edge будуть створенні шв які будуть визначати, як модель буде "розрізана" для UV розгортки. Найкраще обирати місця, які менш помітні для глядача. Натискається Ctrl+E і оберам Mark Seam. Обрані ребра, будуть позначені як шви.

1. Виділити всю модель. Коли всі шви наложенні, натискається клавіша A, щоб модель повністю виділити.

2. Розгортка UV. Натискаєм U і оберам Unwrap. Blender розгортатиме модель на основі позначених швів.

3. Перехід в UV редактор. Переходиться в UV редактор, обравши відповідне вікно в розділі роботи з UV, та знайти розгортку моделі(рис.2.18).

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				
Змн.	Арк.	№ докум.	Підпис	Дата		36

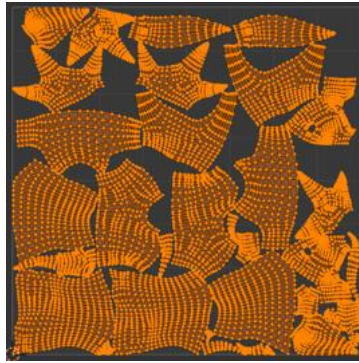


Рис. 2.18. UV Атлас

Збереження файлу. Після завершення ретопології, UV розгортки та інших коригувань, експортується модель для подальшого текстурювання.

2.3 Основні етапи створення текстур

1. Імпорт моделі в Substance Painter

Запуск Substance Painter [20, 7]. Відкривається Substance Painter і створюється новий проект, натиснувши File - New або використавши комбінацію клавіш Ctrl+N.

Імпорт моделі. У діалоговому вікні "New Project" натискається кнопка Select, щоб вибрати файл моделі, далі також потрібно налаштувати роздільну здатність текстур (2048 або 4096), перевіривши опції Template та Normal Map Format, а потім натиснувши ok (рис.2.19).

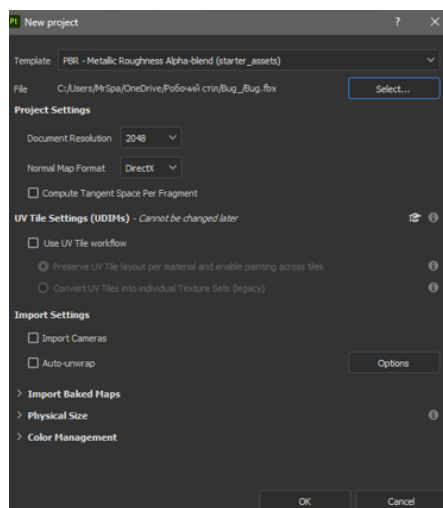


Рис. 2.19. Створення проекту в Substance Painter

2. Бейкінг допоміжних карт

Бейкінг карт. Після імпорту моделі, переходиться до вкладки Texture Set

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				37
Змн.	Арк.	№ докум.	Підпис	Дата		

Settings та натискається кнопка Bake Mesh Maps. Всі карти залишаються включеними. Normal. Карта яка додає дрібні деталі поверхні, створюючи ілюзію рельєфу без додавання додаткових полігонів.

World Space Normal. Карта яка визначає орієнтацію поверхні моделі в глобальній координатній системі світу.

ID. Карта яка застосовується для розділення різних частин моделі на кольорові сегменти.

Ambient Occlusion. Карта яка додає тіні в заглибленнях і на стиках об'єктів, що підвищує візуальну реалістичність.

Curvature. Карта яка відображає вигини та кривизну поверхні моделі.

Position. Карта яка зберігає інформацію про положення кожної точки моделі в 3D-просторі.

Thickness. Карта яка визначає товщину об'єкта в кожній точці поверхні.

Налаштування бейкінгу. У вікні налаштувань бейкінгу встановлюється роздільна здатність карт та потрібно переконатися, що всі параметри правильно налаштовані. В кінці натискається bake selected textures (рис.2.20).

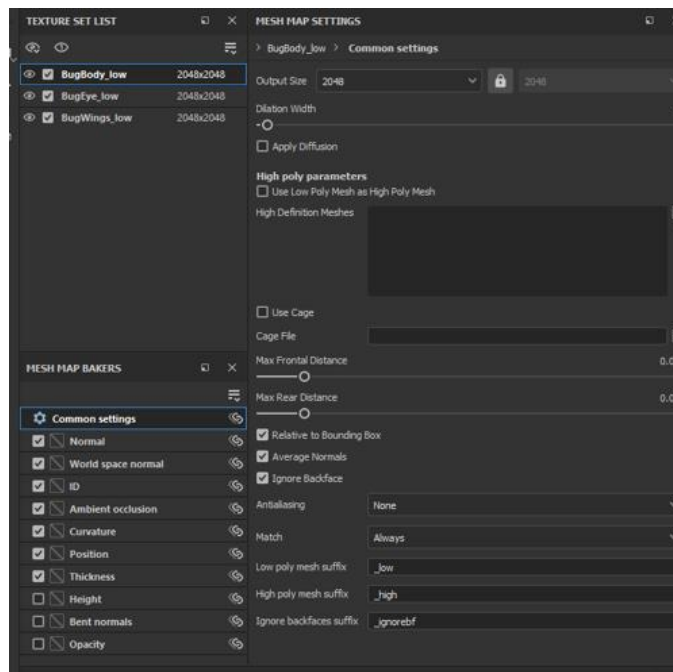


Рис. 2.20. Налаштування бейкінгу

3. Створення та нанесення матеріалів

Нанесення матеріалів. Перетягується обраний матеріал на модель у

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				38
Змн.	Арк.	№ докум.	Підпис	Дата		

вкладці Layers. Додається нові матеріали, деякі з яких мають специфічні властивості, такі як металевість або глянець. Після цього можна налаштувати параметри матеріалу, наприклад, змінити колір, відбивну здатність, прозорість та інші характеристики. У вікні властивостей матеріалу можна регулювати рівень глянцю, ступінь металевості та інші параметри, що дозволяє досягти бажаного вигляду моделі. Використовуючи різні шари та маски, можна комбінувати декілька матеріалів на одній моделі, створюючи складні та реалістичні текстури. (рис.2.21).

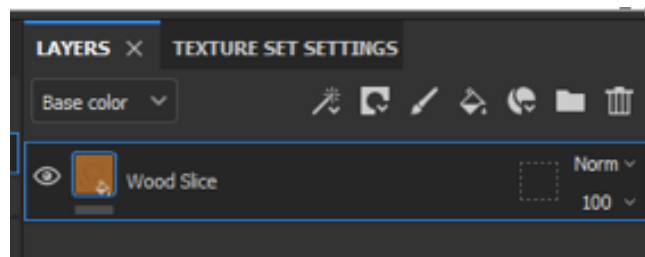


Рис. 2.21. Додавання

4. Маскування та малювання. Натискається правою кнопкою миші (ПКМ) на обраний шар матеріалу та обирається опція Add Black Mask. Чорна маска повністю приховує матеріал на моделі. Використовуючи пензель, можливо малювати білим кольором на масці в тих місцях, де хочемо, щоб матеріал став видимим. Це дозволяє точно контролювати, де саме матеріал має з'явитися на моделі (рис.2.22).

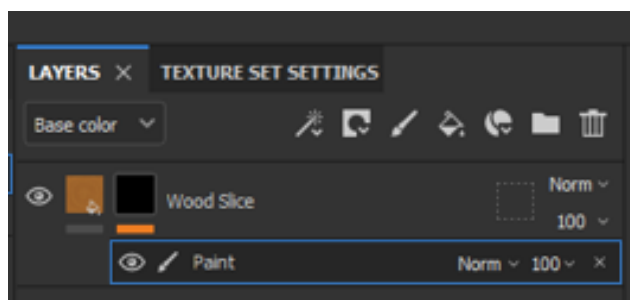


Рис. 2.22. Використання чорної маски

5. Налаштування та коригування текстур

Налаштування слоїв. Слої Wood Slice [11] дозволяє користувачеві детально налаштовувати вигляд текстури деревини, змінюючи її колір, насиченість, варіацію кольору, шорсткість, інтенсивність зернистості, густоту прожилок, їх зникання, викривлення та інтенсивність грубих зрізів (рис.2.23).

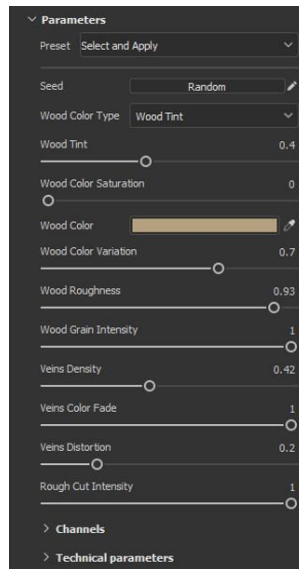


Рис. 2.23. Коригування слою

6. Додавання ефектів. Параметри генератора дозволяють користувачеві налаштовувати вигляд та властивості текстури бруду, включаючи рівень бруду, контраст, використання тріпланарного мапування, кількість та масштаб грубої текстури, а також маскування країв. Перемикачі дозволяють вмикати або вимикати функції, такі як інвертування та використання користувацької грубої текстури. Додається додаткові ефекти, такі як потертості, подряпини або бруд, використовуючи генератори та фільтри (рис.2.24).

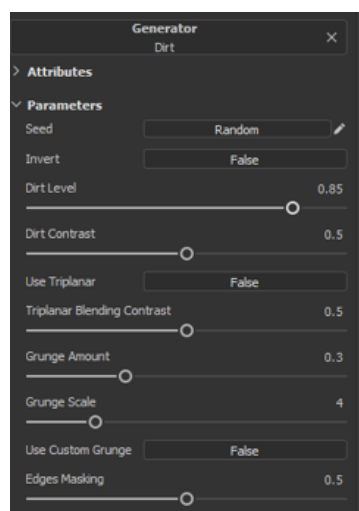


Рис. 2.24. Використання генератора Dirt

Використовуючи всі продемонстровані інструменти можна створити готові текстури для персонажа [14] (рис.2.25).

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				40
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис. 2.25. Модель з готовими текстурами

7. Експорт текстур. Переконавши що всі текстури нанесено коректно, та відповідають поставленим задачам і вибранному стилю, натискається File - Export Textures. У вікні експорту обирається потрібний пресет (наприклад, для Unreal Engine або Unity), вказавши папку для збереження текстур та налаштувавши параметри експорту (роздільну здатність, формати файлів тощо). Потім натискається Export (рис.2.26).

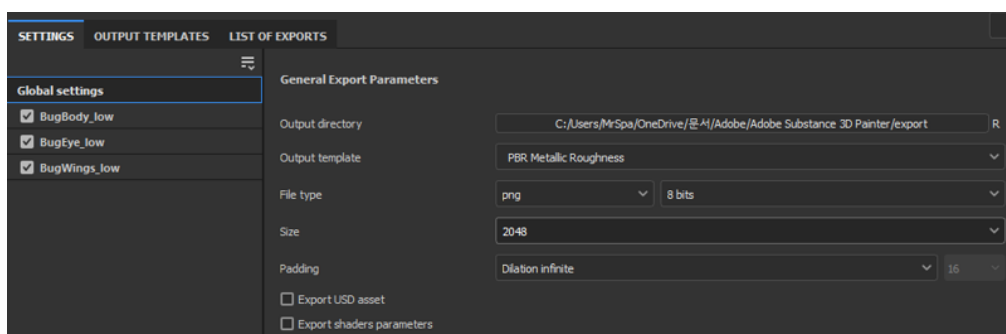


Рис. 2.26. Експорт текстур із Substance Painter

Після експорту текстур із Substance Painter важливо імпортувати їх у Blender для перевірки та налаштування перед відправкою в ігровий двигун.

Відкривається Blender і завантажується 3D-модель, переконавши що модель має правильний масштаб після імпорту, створюється новий матеріал для перевірки.

Далі потрібно за допомогою нодової системи імпортувати текстури до моделі. Нодова система в Blender використовується для створення і налаштування матеріалів, текстур, композиції та інших аспектів рендерингу шляхом з'єднання різних елементів (нодів) у вигляді графа. Ноди (вузли) - це окремі блоки, кожен з яких виконує певну функцію, і вони можуть бути

пов'язані між собою через входи і виходи для створення складних ефектів.

У рамках цього розділу було використано тільки 4 ноди.

1. Principled BSDF. Нода Principled BSDF є універсальним шейдером, який дозволяє створювати різні типи матеріалів, поєднуючи в собі функціонал багатьох інших шейдерів. Вона була розроблена для спрощення налаштувань матеріалів і надає широкий спектр параметрів для тонкого налаштування вигляду поверхні.

Нода має численні параметри, такі як базовий колір, металевість, шерсткість, тощо. Вона дозволяє користувачам створювати реалістичні матеріали з відблисками, відбиттями та іншими оптичними ефектами. При імпорті текстур, різні текстурні карти (наприклад, карта кольору, карта відблисків) можуть бути підключені до відповідних входів цієї ноди (рис.2.27).

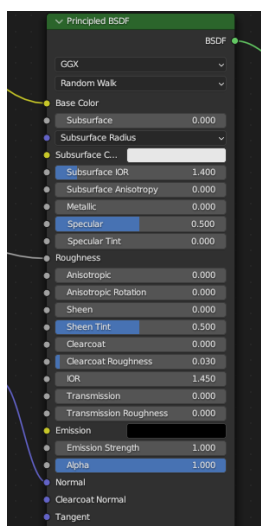


Рис. 2.27. Principled BSDF

2. Material Output. Нода Material Output є кінцевою точкою в графі матеріалу, яка визначає, як матеріал буде застосований до моделі під час рендерингу. Вона приймає дані від інших нод і передає їх до рендерера для візуалізації. Всі шейдери та текстури підключаються до цієї ноди. Вона має кілька входів, включаючи поверхню (Surface), об'єм (Volume) і дисплейсмент (Displacement), через які можна керувати різними аспектами матеріалу. Зазвичай нода Principled BSDF підключається до входу Surface ноди Material Output (рис.2.28).

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				
Змн.	Арк.	№ докум.	Підпис	Дата		42

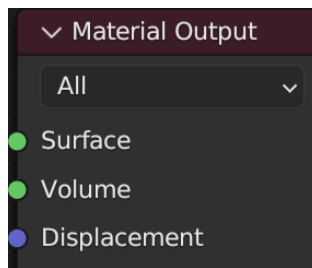


Рис. 2.28. Material Output

3. Image Texture. Нода Image Texture (Додаток Б) використовується для завантаження зображень, які потім можуть бути використані як текстури на моделі. Це можуть бути карти кольору, нормалей, рельєфу та інші типи текстур. Ця нода дозволяє завантажувати зовнішні зображення і підключати їх до відповідних входів інших нод. Наприклад, карта кольору (diffuse map) може бути підключена до базового кольору (Base Color) ноди Principled BSDF, а карта нормалей – до відповідного входу Normal Map. Нода Image Texture має параметри для вибору зображення, налаштування його повторення (tiling), а також інші налаштування текстурних координат (рис.2.29).

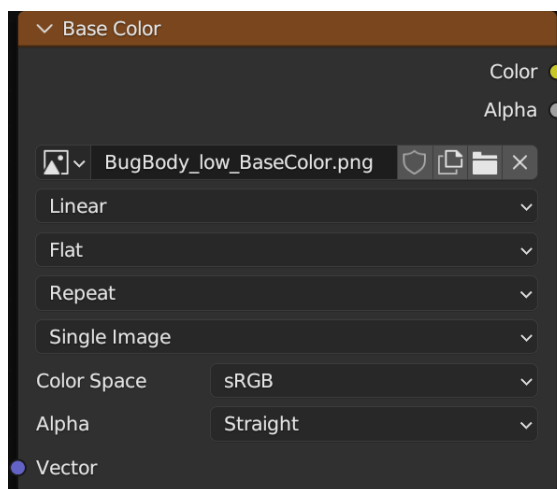


Рис. 2.29. Image Texture

4. Нода Normal Map використовується для інтерпретації карти нормалей, яка визначає дрібні деталі та нерівності на поверхні моделі без зміни її геометрії. Карта нормалей (normal map) завантажувється за допомогою ноди Image Texture і підключається до входу ноди Normal Map. Вона переводить інформацію з карти нормалей у формат, який може бути використаний іншими шейдерами. Для коректної роботи важливо переконатися, що тип колірного простору в ноді Image Texture встановлений на "Non-Color Data", оскільки

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				43
Змн.	Арк.	№ докум.	Підпис	Дата		

карта нормалей не є кольоровим зображенням, а представляє вектори напрямку поверхневих нормалей. Вихід цієї ноди підключається до входу Normal ноди Principled BSDF, що додає деталі до поверхні матеріалу, роблячи її більш реалістичною. Також, для більш тонкої настройки, можна використовувати ноду Vector Bump, яка дозволяє регулювати інтенсивність впливу карти нормалей на матеріал. Це забезпечує додатковий контроль над виглядом дрібних деталей на поверхні моделі (рис.2.30).

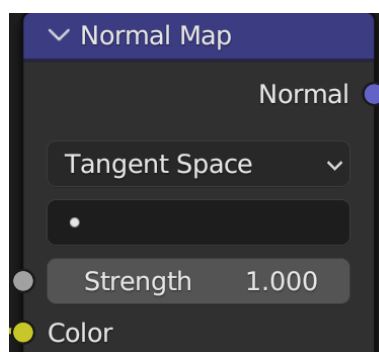


Рис. 2.30. Normal Map

Image Texture (Color Map) завантажує та передає кольорову текстуру до Principled BSDF.

Image Texture (Normal Map) завантажує та передає карту нормалей до Normal Map.

Normal Map обробляє карту нормалей та передає її до Principled BSDF.

Principled BSDF об'єднує всю інформацію про матеріал і передає її до Material Output, який застосовує матеріал до моделі під час рендерингу.

Після додавання всіх текстур за допомогою нодової системи, модель виглядає коректно і всі текстури відображаються належним чином (рис.2.31).



Рис. 2.31. Готовий результат

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				
Змн.	Арк.	№ докум.	Підпис	Дата		44

2.4 Реалізація програмного кода

Easy Tool — це плагін для Blender, розроблений для полегшення роботи з об'єктами в 3D сценах. Він надає набір корисних інструментів, які значно прискорюють процес створення та редагування моделей. Плагін додає нову панель у вікні 3D View, де зібрані всі його функції.

Функції плагіна.

1. Додавання та видалення матеріалів.

Create Material. Створює новий матеріал для вибраних об'єктів, використовуючи назву об'єкта як назву матеріалу. Це допомагає швидко застосовувати матеріали до об'єктів.

Remove Material. Видаляє всі матеріали з вибраних об'єктів і очищає непотрібні матеріали з бібліотеки даних.

2. Робота з Face Orientation.

Face Orientation. Перемикає відображення орієнтації обличчя в 3D вікні, що дозволяє легко виявляти проблеми з нормальними об'єктів.

3. Додавання та видалення суфіксів до назв об'єктів.

Add _low Suffix: Додає суфікс "_low" до назв вибраних об'єктів.

Add _high Suffix: Додає суфікс "_high" до назв вибраних об'єктів.

Delete Suffix: Видаляє всі суфікси "_low" і "_high" з назв вибраних об'єктів.

4. Робота з користувацьким текстом.

Add Text to Selected. Додає користувацький текст до назв вибраних об'єктів. Користувач може ввести текст у спеціальному полі на панелі.

Remove Text from Selected. Видаляє користувацький текст з назв вибраних об'єктів.

Clear Custom Text. Очищає поле користувацького тексту.

Основний клас плагіну __init__.py.

```
import bpy
```

```
from .panels import my_panel
```

```
from .preferences import addon_preferences
```

```
from .operators.add_low_suffix import OBJECT_OT_add_low_suffix
```

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				45
Змн.	Арк.	№ докум.	Підпис	Дата		

```

from .operators.add_high_suffix import OBJECT_OT_add_high_suffix
from .operators.create_and_apply_material import OBJECT_OT_create_and_apply_material
from .operators.remove_material import OBJECT_OT_remove_material
from .operators.toggle_face_orientation import OBJECT_OT_toggle_face_orientation
from .operators.delete_all_suffix import OBJECT_OT_delete_all_suffix
from .operators.add_text_to_selected import OBJECT_OT_add_text_to_selected
from .operators.remove_text_from_selected import OBJECT_OT_remove_text_from_selected
from .operators.clear_custom_text import OBJECT_OT_clear_custom_text

```

Імпорт необхідних модулів та класів, що визначають різні частини аддону.

- bpy - головний модуль Blender Python API;
- my_panel - модуль, який містить клас панелі інтерфейсу користувача;

- addon_preferences - модуль, який містить клас налаштувань аддону;
- OBJECT_OT_* - модулі, які містять оператори, що виконують різні дії в Blender.

```

def register():
    bpy.utils.register_class(addon_preferences.MyAddonPreferences)
    bpy.utils.register_class(my_panel.MyPanel)
    bpy.utils.register_class(OBJECT_OT_add_low_suffix)
    bpy.utils.register_class(OBJECT_OT_add_high_suffix)
    bpy.utils.register_class(OBJECT_OT_create_and_apply_material)
    bpy.utils.register_class(OBJECT_OT_remove_material)
    bpy.utils.register_class(OBJECT_OT_toggle_face_orientation)
    bpy.utils.register_class(OBJECT_OT_delete_all_suffix)
    bpy.utils.register_class(OBJECT_OT_add_text_to_selected)
    bpy.utils.register_class(OBJECT_OT_remove_text_from_selected)
    bpy.utils.register_class(OBJECT_OT_clear_custom_text)
    bpy.types.Scene.custom_text = bpy.props.StringProperty(name="Custom Text")

```

Функція що реєструє класи аддону та додає користувацьку властивість до сцени.

1. register_class - реєструє кожен клас аддону, щоб він був доступний у Blender;
 - addon_preferences.MyAddonPreferences - клас для налаштувань аддону, що дозволяє користувачам налаштовувати аддон через інтерфейс Blender;
 - my_panel.MyPanel - клас панелі інтерфейсу, який додається до вікна Properties або 3D View;

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				
Змн.	Арк.	№ докум.	Підпис	Дата		46

- OBJECT_OT_* - оператори, які виконують конкретні функції в Blender, такі як додавання суфіксів до об'єктів, створення та застосування матеріалів, видалення матеріалів, перемикання орієнтації граней, та маніпуляція з текстом;

2. додавання користувацької властивості.

- bpy.types.Scene.custom_text - додає нову властивість типу StringProperty до сцени, яка зберігає користувацький текст. Ця властивість буде доступна в будь-якій сцені в Blender;

def unregister():

```
bpy.utils.unregister_class(addon_preferences.MyAddonPreferences)
bpy.utils.unregister_class(my_panel.MyPanel)
bpy.utils.unregister_class(OBJECT_OT_add_low_suffix)
bpy.utils.unregister_class(OBJECT_OT_add_high_suffix)
bpy.utils.unregister_class(OBJECT_OT_create_and_apply_material)
bpy.utils.unregister_class(OBJECT_OT_remove_material)
bpy.utils.unregister_class(OBJECT_OT_toggle_face_orientation)
bpy.utils.unregister_class(OBJECT_OT_delete_all_suffix)
bpy.utils.unregister_class(OBJECT_OT_add_text_to_selected)
bpy.utils.unregister_class(OBJECT_OT_remove_text_from_selected)
bpy.utils.unregister_class(OBJECT_OT_clear_custom_text)
```

del bpy.types.Scene.custom_text

Функція знімає реєстрацію класів аддону та видаляє користувацьку властивість зі сцени:

1. unregister_class - відреєструє кожен клас аддону, роблячи його недоступним у Blender;

- addon_preferences.MyAddonPreferences - відреєструє клас налаштувань аддону;

- my_panel.MyPanel - відреєструє клас панелі інтерфейсу;

- OBJECT_OT_*. - відреєструє оператори, які виконували різні функції в Blender;

2. Видалення користувацької властивості.

- del bpy.types.Scene.custom_text - видаляє властивість custom_text з типу Scene, що робить її недоступною в будь-якій сцені в Blender;

```
if __name__ == "__main__":
    register()
```

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				
Змн.	Арк.	№ докум.	Підпис	Дата		47

register() викликає функцію, якщо скрипт виконується безпосередньо.

- if __name__ == "__main__" - перевіряє чи виконується скрипт безпосередньо (не імпортується як модуль);

- register() - викликає функцію реєстрації, щоб зареєструвати всі класи аддону та додати користувацьку властивість до сцени;

Переваги плагіна.

- Зручний інтерфейс, що інтегрується у вікно 3D View. Додає нову панель безпосередньо в інтерфейс Blender, що розташовується у вікні 3D View. Це дозволяє користувачам миттєво отримати доступ до функцій плагіна без переходу до інших меню чи вікон.

- Швидкий доступ до часто використовуваних функцій. Панель Easy Tool містить набір кнопок, які виконують часто використовувані завдання, такі як створення та видалення матеріалів, додавання суфіксів до назв об'єктів і робота з користувацьким текстом. Це дозволяє швидко та ефективно виконувати рутинні операції.

- Можливість кастомізації за допомогою користувацьких іконок. Плагін дозволяє користувачам завантажити власні іконки для кожної кнопки на панелі. Це дозволяє персоналізувати вигляд інтерфейсу, роблячи його більш зрозумілим та зручним для користувача.

- Прискорює робочий процес, зменшуючи кількість рутинних операцій. Easy Tool допомагає оптимізувати робочий процес шляхом автоматизації ряду завдань, таких як додавання суфіксів до назв об'єктів або видалення матеріалів. Це дозволяє користувачам швидше та ефективніше виконувати свої завдання, зменшуючи кількість часу, витраченого на рутинні операції, і зосереджуючи увагу на творчому процесі.

Діаграма компонентів(рис.2.32) відображає архітектуру додатку, де панель (MyPanel) відповідає за взаємодію з користувачем і викликає різні оператори для виконання відповідних завдань. MyAddonPreferences зберігає налаштування додатку і також має метод для відображення контенту.

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				
Змн.	Арк.	№ докум.	Підпис	Дата		48

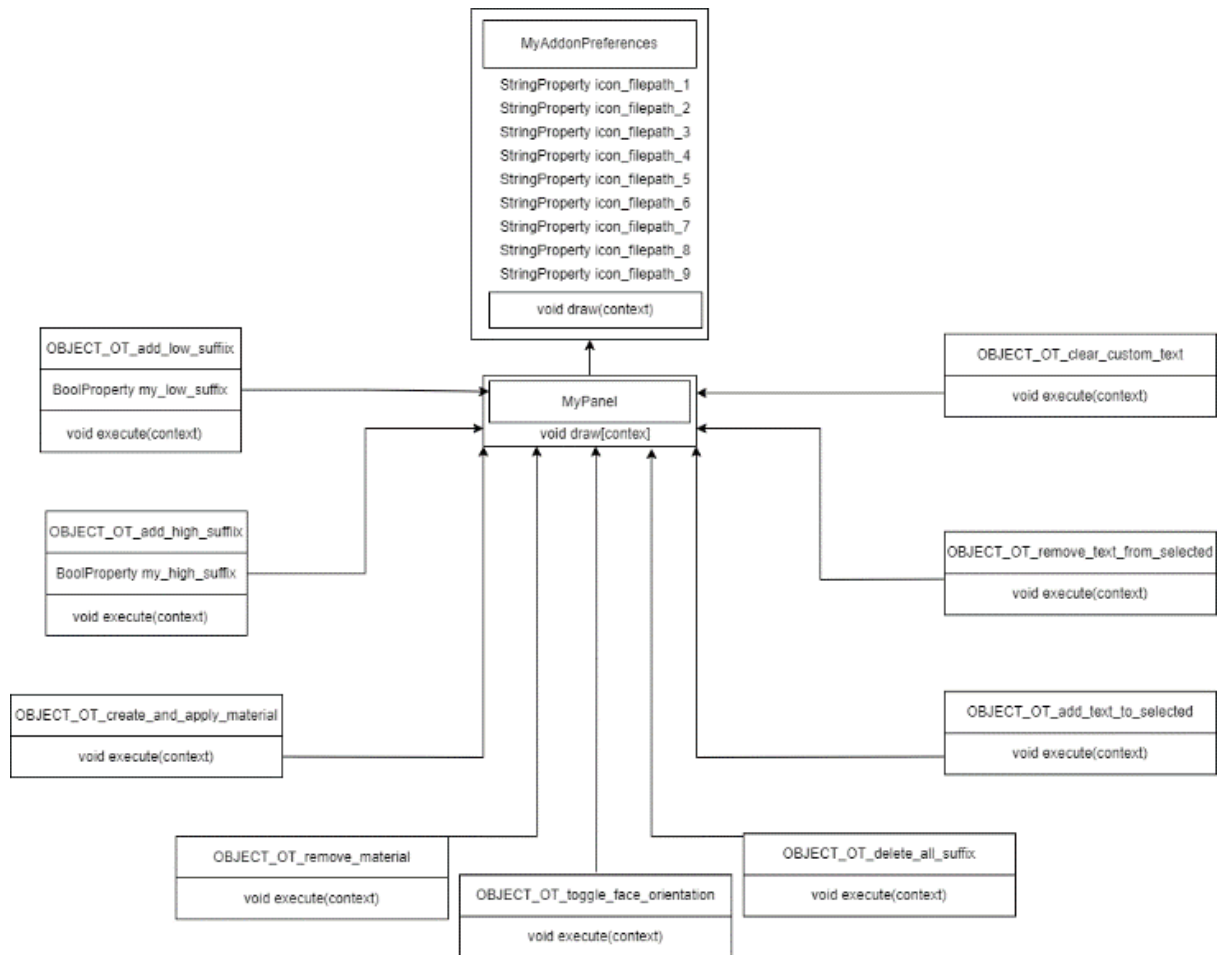


Рис. 2.32. Діаграма Компонентів

Діаграма класів(рис.2.33) представляє архітектуру додатку для Blender, показуючи різні оператори, панелі та налаштування, які взаємодіють між собою для забезпечення функціональності. Ця структура дозволяє легко розширювати функціональність додатку, додаючи нові оператори, панелі або налаштування. Кожен елемент має чітко визначені функції та взаємодії, що робить код організованим і зрозумілим.

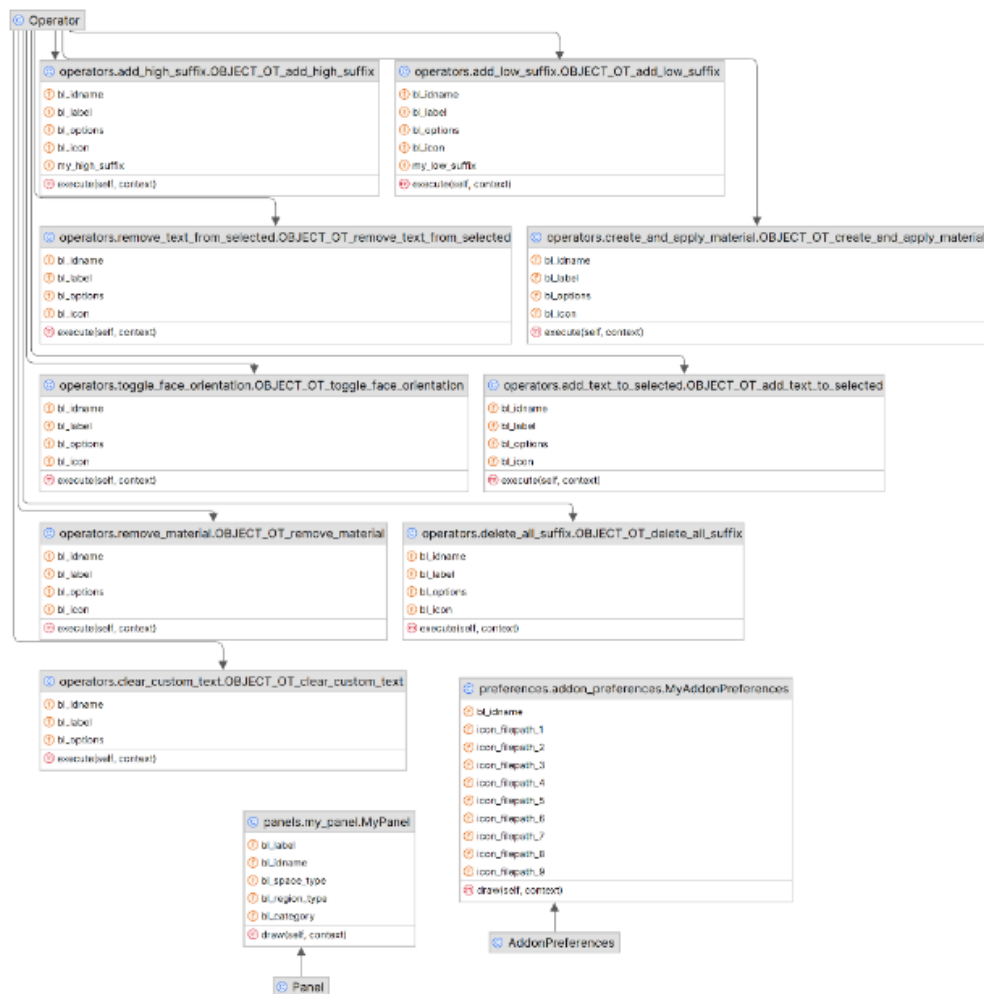


Рис. 2.33. Діаграма класів

Висновки до другого розділу

Було продемонстровано повний процес створення ігрового персонажа, який включав кілька ключових етапів, таких як скульптинг, ретопологія, UV та текстурування. Огляд процесу супроводжувався чотирма діаграмами та таблицями, що ілюструють кожен етап роботи. Крім того, детально було розглянуто техніки використання програмного забезпечення для кожного з цих етапів, а також наведено приклади налаштувань та параметрів, які забезпечують найкращі результати. Учасники мали змогу ознайомитися з практичними рекомендаціями щодо уникнення поширених помилок, а також отримали корисні поради щодо оптимізації роботи для досягнення високої якості фінального продукту.

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				50
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. ОЦІНКА ФУНКЦІОНАЛЬНОСТІ ТА НАДІЙНОСТІ МОДЕЛЕЙ

3.1 Тестування Моделі

Розробка 3D моделей для відеоігор вимагає ретельної перевірки на всіх етапах створення. Від точності топології до правильності текстуровання, кожен аспект моделі повинен бути бездоганим, щоб забезпечити її функціональність та естетичну привабливість у грі. Розділ описує детальний процес перевірки моделей, який включає різні методи і техніки з використанням плагінів та внутрішніх програмних інструментів для виявлення та виправлення помилок.

Використання внутрішніх програмних інструментів Blender для перевірки

1. Режим перегляду Wireframe. Переходим до режиму Wireframe (Клавіша Z на клавіатурі), це дозволяє легко бачити внутрішню структуру моделі та виявляти проблемні області, якщо їх не видно на поверхності моделі (рис.3.1).

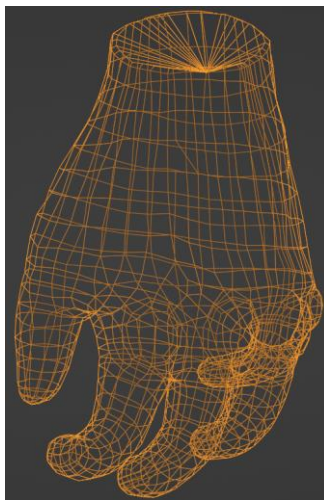


Рис. 3.1. Режим перегляду Wireframe

2. Перевірка нормалей.

Нормалі – це вектори, які вказують напрямом, перпендикулярний до поверхні кожного полігону або вершини. Вони відіграють ключову роль у відображенні світла і тіней на 3D-моделі, визначаючи, як світло буде взаємодіяти з поверхнею.

Неправильні нормалі можуть призвести до артефактів освітлення та несподіваних візуальних ефектів.

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				
Змн.	Арк.	№ докум.	Підпис	Дата		51

Для перевірки правильності увімкнено відображення нормалей у меню Overlay (Для коректної роботи потрібно перейти в режим редагування). Правильна нормальна повинна відображатися синім кольором, якщо було побачанно області з червоним, це означає що на полігону вивернуті нормалі що призводить до артефактів таких як почорніння полігонів моделі(рис.3.3), якщо їх не виправити це може призвести до виникнення артефактів освітлення.

Виправлення. Команда Alt+N (Recalculate Normals) після чого вона стане коректного синього кольору(рис.3.4).

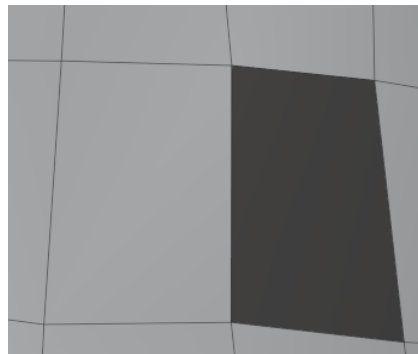


Рис. 3.3. Артефакт

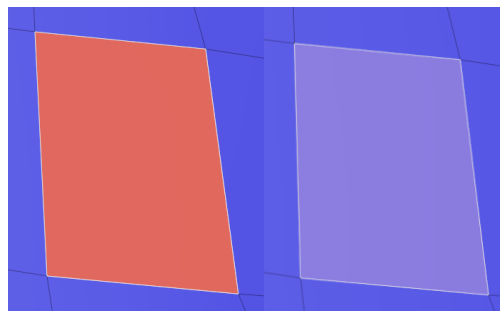


Рис. 3.4. Виправлення нормалі

Тестування гладкості поверхонь за допомогою Shading.

Режим відображення в матеріалах Matcap (Material Capture) дозволяє швидко переглянути модель з використанням готових матеріалів і освітлення, що допомагає виявити артефакти та нерівності поверхні. При перевірці моделі в режимі Matcap (рис.3.5), всі поверхні виглядають гладкими і коректними, що свідечує про те, що модель готова до подальшої інтеграції в ігровий двіжок.



Рис. 3.5. Matcap

Використання плагіна.

Встановлення та активація. Завантажуємо плагін "Mesh Check BGL edition" з офіційного джерела магазину аддонів Blender [13].

Плагін "Mesh Check BGL edition" для Blender — це інструмент, призначений для перевірки та виправлення топології 3D-моделей. Він допомагає виявити та виправити різні проблеми в сітці, що можуть виникнути під час моделювання. Основні функції та можливості плагіну включають:

1. Перевірка геометрії сітки. Визначення некоректних граней. Плагін може виявляти проблемні елементи, такі як нгеги (ngons — багатокутники з більш ніж чотири вершинами), самоперетинаючі грані та невірно орієнтовані нормалі. Пошук дублікатів. Виявлення дублікатів вершин, ребер і граней, які можуть створювати зайву геометрію.

2. Візуалізація проблем. Плагін надає візуальні підказки, використовуючи кольорове кодування для виділення проблемних областей на сітці, що дозволяє швидко знаходити та виправляти помилки.

3. Інтерактивні інструменти виправлення. Злиття вершин: Можливість злиття дублікатів вершин. Виправлення нормалей: Автоматичне виправлення нормалей для коректного відображення поверхонь. Переробка граней: Інструменти для зміни топології сітки, наприклад, для поділу багатокутників на чотирикутники або трикутники.

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				
Змн.	Арк.	№ докум.	Підпис	Дата		53

4. Підтримка різних форматів. Плагін може працювати з моделями, імпортованими з різних форматів файлів, забезпечуючи універсальність його використання в різних робочих процесах.

5. Інтеграція з іншими інструментами Blender. Легко інтегрується з іншими інструментами та аддонами Blender, дозволяючи використовувати його як частину комплексного процесу моделювання.

Процес перевірки моделі.

Відкривається Blender і переходиться до меню "Edit" - "Preferences" - "Add-ons".

Натискається "Install" і вибирається файл плагіна. Після встановлення знаходиться плагін у списку аддонів і ставиться галочка біля нього для активації (рис.3.6).

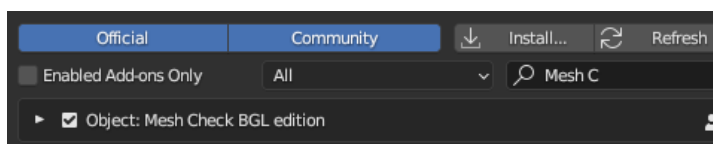


Рис. 3.6. Установка плагіна Mesh Check BGL edition

Використання плагіна для перевірки. Запускається Mesh Check. Переходиться до режиму редагування (Edit Mode), натиснувши клавішу Tab або вибравши відповідний режим у нижньому меню (рис.3.7).

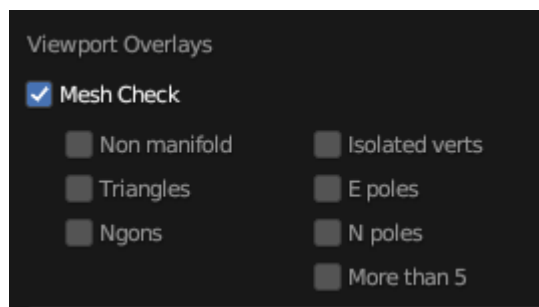


Рис. 3.7. Плагін Mesh Check BGL edition

Перевірка геометрії. В панелі інструментів (розташована зліва) або у вкладці "N" панелі праворуч знаходиться інструменти плагіна "Mesh Check BGL edition". Виберається опція "Check Mesh" або подібну для початку перевірки. Плагін проведе аналіз сітки і виділить проблемні області.

Візуалізація проблеми. Проблемні ділянки будуть підсвічені різними кольорами, що означає типи помилок, наприклад, червоний для нгегов (ngons),

синій для дубльованих вершин, зелений для самоперетинаючих граней тощо (рис.3.8).

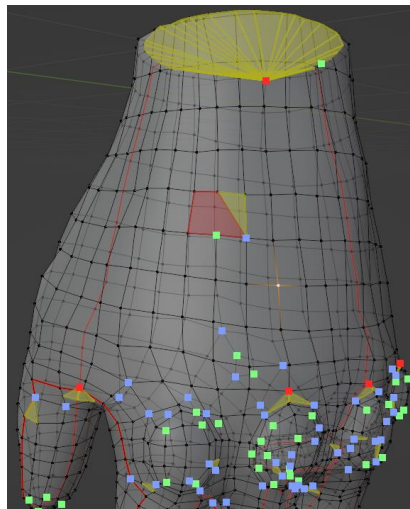


Рис. 3.8. Перевірка моделі

Виправлення геометрії.

Для виправлення вибирається проблемна область і використовується інтерактивні інструменти плагіна для виправлення. Наприклад, злиття вершин, видалення дублюючих граней, виправлення нормалей.

Для злиття вершин виділяється модель та натискається "Merge Vertices". Для виправлення нормалей вибирається "Recalculate Normals".

Завершення перевірки

Підсумкова перевірка. Після виправлення всіх помилок запускаєм перевірку ще раз, щоб переконатися, що всі проблеми усунуті (рис.3.9).

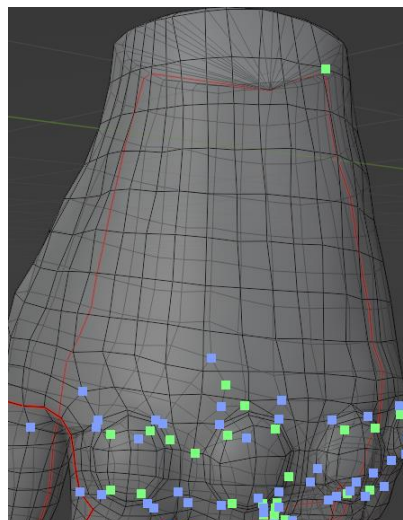


Рис. 3.9. Повторна перевірка моделі

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				
Змн.	Арк.	№ докум.	Підпис	Дата		55

Після виконання виправлень та повторної перевірки з допомогою плагіна модель виглядає коректно. Усі виявлені раніше помилки були успішно виправлені, що підтверджується відсутністю будь-яких помилок чи попереджень під час повторної перевірки. Геометрія моделі тепер є чистою, без зайвої вершини, дубльованих граней або невідтримуваних елементів. Завдяки цьому модель готова до подальшого використання чи інтеграції в проект без ризику виникнення проблем під час рендерингу чи анімації.

3.2 Тестування UV розгортки

Одним з важливих етапів підготовки 3D-моделі до текстурування є перевірка UV розгортки на наявність перекриттів або спотворень, оскільки від цього напряму залежить якість текстури моделі. Для цього можна використати Checker Texture - текстуру у вигляді шахової дошки(рис.3.3), яка дозволяє візуально оцінити рівномірність та правильність розгортки.

Процес перевірки.

1. Створення Checker Texture. В режимі редагування потрібно обрати UV розгортку моделі. Після чого створюється новий матеріал та додається до нього Checker Texture. Для цього потрібно з допомогою Shader Editor додати ноду Checker Texture та під'єднати до Base Color в ноді Principled BSDF.

2. Налаштування відображення текстури. Режим відображення текстур (Texture Mode) у вікні 3D-огляду. Це дозволить бачити накладену текстуру безпосередньо на вашій моделі.

3. Аналіз UV розгортки. Оглядається модель, звертаючи увагу на шахову текстуру. Якщо квадрати текстури рівномірні й мають однаковий розмір по всій поверхні моделі, це свідчить про правильну UV розгортку.

Перекося, розтягнення або перекриття квадратів текстури вказують на проблеми з UV розгорткою. Ці проблеми можуть спричиняти дефекти при накладенні фінальних текстур.

Корекція помилок.

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				
Змн.	Арк.	№ докум.	Підпис	Дата		56

1. Виправлення перекриттів. Використовується інструменти в UV Editor для корекції позицій UV вершин, щоб усунути перекриття. Це можна зробити вручну, переміщуючи вершини та ребра.

2. Виправлення. Розтягнення текстури можна зменшити шляхом використання функції Unwrap з різними параметрами, таких як Smart UV Project, Cube Projection або Cylinder Projection (рис.3.10).

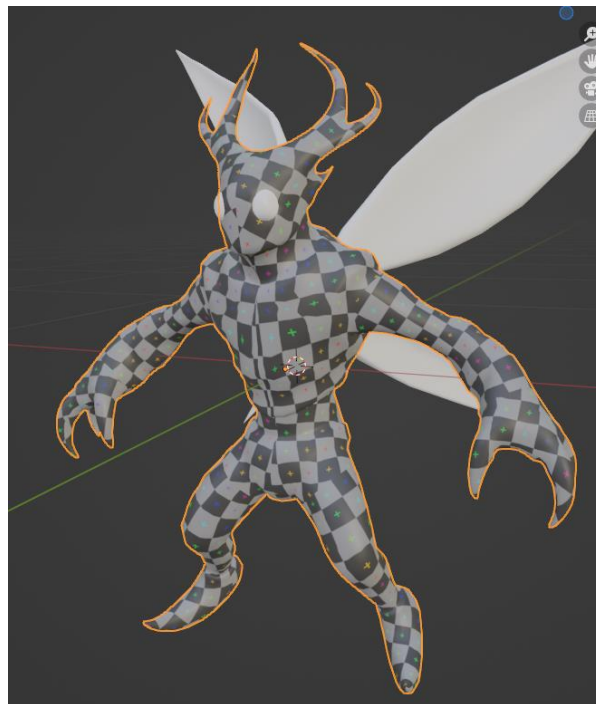


Рис. 3.10. Checker Texture

Після перевірки UV розгортки за допомогою Checker Texture, помилок або порушень у моделі не знайдено. UV-шви правильно розміщені, розгортка не містить перекриттів, спотворень чи інших артефактів. Всі полігони рівномірно заповнюють UV-простір, що забезпечує якісне та однорідне нанесення текстур на модель.

Висновки до третього розділу

У третьому розділі було детально розглянуто процес тестування моделі та UV-карти для виявлення помилок або спотворень. Перевірка моделей є ключовим етапом у створенні 3D контенту для відеоігор. Вона гарантує візуальну якість, оптимізацію продуктивності, точність анімацій, правильність текстур та матеріалів, а також сумісність з ігровим двигуном. Важливо

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				
Змн.	Арк.	№ докум.	Підпис	Дата		57

виявляти та виправляти помилки на ранніх стадіях, щоб уникнути проблем у майбутньому. Ці заходи забезпечують відповідність моделей високим стандартам якості та сприяють створенню захоплюючого та привабливого ігрового досвіду для гравців.

		<i>М. О. ТИХОНОВ</i>			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		<i>Г.В.МАРЧУК</i>				
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		58

ВИСНОВКИ

В ході виконання кваліфікаційної роботи було розроблено 3D моделі для ігрового додатку "Forbidden Forest" у жанрі Rogue-lite.

У першому розділі визначено основну мету розробки та поставлено завдання. Здійснено огляд трьох популярних ігор у жанрі Rogue-lite, виділено їхні сильні та слабкі сторони. Проведено аналіз графіки в обраному жанрі та визначено ключові особливості для її відтворення у грі. Обґрунтовано вибір інструментів (Zbrush, Blender і Substance Painter), враховуючи їхні переваги, які стануть корисними для проекту.

У другому розділі детально розглянуто процес проектування та розробки 3D моделей для ігрового додатку "Forbidden Forest". Ключові етапи роботи включали проектування за допомогою діаграм UML, створення моделей, текстурування та реалізацію програмного коду.

У третьому розділі докладно описано процес тестування моделей та їх UV-розгортку для виявлення можливих помилок або спотворень. Підкреслено важливість перевірки моделей як ключового етапу у створенні 3D контенту для відеоігор. Це забезпечує високу візуальну якість та оптимізацію затрат ресурсів гри. Виявлення та виправлення помилок на ранніх етапах дозволяє уникнути проблем у майбутньому. Такі заходи гарантують відповідність моделей високим стандартам якості та сприяють створенню захоплюючого ігрового досвіду для гравців.

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				
Змн.	Арк.	№ докум.	Підпис	Дата		59

СПИСОК ВИКОРАСТАНИХ ДЖЕРЕЛ

1. Borderlands 3 [Електронний ресурс] // Steam. – 2020. – Режим доступу до ресурсу: https://store.steampowered.com/app/397540/Borderlands_3/.
2. Rage 2 [Електронний ресурс] // Steam. – 2019. – Режим доступу до ресурсу: https://store.steampowered.com/app/548570/RAGE_2/.
3. The Outer Worlds [Електронний ресурс] // Steam. – 2020. – Режим доступу до ресурсу: https://store.steampowered.com/app/578650/The_Outer_Worlds/.
4. The Role of 3D Technology in Video Games [Електронний ресурс] // Stefano Triberti. – 2018. – Режим доступу до ресурсу: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0200724>
5. The Evolution of 3D Graphics in Video Games: The Hunt for Photorealism [Електронний ресурс] // Gameopedia. – 2020. – Режим доступу до ресурсу: <https://www.gameopedia.com/the-evolution-of-3d-graphics-in-video-games-the-hunt-for-photorealism/>.
6. ZBrush 2024 Features Official Documentation [Електронний ресурс] // Pixologic. – 2024. – Режим доступу до ресурсу: <https://help.maxon.net/zbr/en-us/#html/features/features.html>.
7. Substance Painter Official Documentation [Електронний ресурс] // Adobe. – 2023. – Режим доступу до ресурсу: <https://helpx.adobe.com/substance-3d-painter/home.html>.
8. Blender 4.1 Reference Manual [Електронний ресурс] // Blender Foundation. – 2024. – Режим доступу до ресурсу: <https://docs.blender.org/manual/en/latest/>.
9. Mastering 3D Modeling in Blender: From Novice to Pro [Електронний ресурс] // ResearchGate. – 2023. – Режим доступу до ресурсу: https://www.researchgate.net/publication/375025409_Mastering_3D_Modeling_in_Blender_From_Novice_to_Pro.

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				60
Змн.	Арк.	№ докум.	Підпис	Дата		

10. Blender Introduction to Modeling [Електронний ресурс] // Blender Foundation. – 2024. – Режим доступу до ресурсу: <https://docs.blender.org/manual/en/latest/modeling/meshes/introduction.html>.
11. Substance Share Library [Електронний ресурс] // Adobe. – 2024. – Режим доступу до ресурсу: <https://substance3d.adobe.com/community-assets>.
12. How To Retopologize Your Model The Fast Way? [Електронний ресурс] // Blender Base Camp – 2023. – Режим доступу до ресурсу: <https://blenderbasecamp.com/how-to-retopologze-your-model-the-fast-way/>.
13. Mesh Check GPU Edition [Електронний ресурс] // Gumroad. – 2023. – Режим доступу до ресурсу: https://pistiwique.gumroad.com/l/mesh_check_BGL_edition.
14. Substance Painter for Game Texturing [Електронний ресурс] // Allegorithmic. – 2020. – Режим доступу до ресурсу: <https://www.adobe.com/products/substance3d/apps/painter.html>.
15. Player Experience and Enjoyment [Електронний ресурс] // Sean Eshuis. – 2023. – Режим доступу до ресурсу: <https://journals.sagepub.com/doi/full/10.1177/10468781231158818>.
16. Global visual attention SPAN in different video game genres [Електронний ресурс] // Marc Argilés. – 2023. – Режим доступу до ресурсу: <https://www.nature.com/articles/s41598-023-49434-1>.
17. The role of game genres and gamers [Електронний ресурс] // Chang Won Jung. – 2020. – Режим доступу до ресурсу: <https://www.nature.com/articles/s41599-020-0439-y>.
18. An Operational Study of Video Games' Genres [Електронний ресурс] // University of Jeddah. – 2020. – Режим доступу до ресурсу: <https://online-journals.org/index.php/i-jim/article/view/16691>.
19. About Blender [Електронний ресурс] // Blender Foundation. – 2024. – Режим доступу до ресурсу: https://docs.blender.org/manual/en/latest/getting_started/about/index.html.

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				61
Змн.	Арк.	№ докум.	Підпис	Дата		

20. Getting Started with Substance 3D Painter [Електронний ресурс] // Adobe. – 2021. – Режим доступу до ресурсу: <https://creativecloud.adobe.com/learn/substance-3d-painter/web/getting-started-with-substance-3d-painter-2021?locale=en>.

21. The Iterative Closest Point Algorithm: Principles and Applications [Електронний ресурс] // Jana Procházková. – 2018. – Режим доступу до ресурсу: https://www.researchgate.net/publication/324500004_Notes_on_Iterative_Closest_Point_Algorithm.

22. Stylized Character Workflow - UV Mappin [Електронний ресурс] // Blender Studio. – 2020. – Режим доступу до ресурсу: <https://studio.blender.org/training/stylized-character-workflow/live-uv-mapping-finale/>.

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				62
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				63
Змн.	Арк.	№ докум.	Підпис	Дата		

preferences\addon_preferences.py

```
import bpy
import os

class MyAddonPreferences(bpy.types.AddonPreferences):
    bl_idname = "my_simple_addon"

    icon_filepath_1: bpy.props.StringProperty(
        name="Create Material",
        subtype='FILE_PATH',
        default=os.path.join(os.path.dirname(__file__), "..", "icons", "bake_obj_cage.png"),
    )

    icon_filepath_2: bpy.props.StringProperty(
        name="Remove Material",
        subtype='FILE_PATH',
        default=os.path.join(os.path.dirname(__file__), "..", "icons", "op_uv_fill.png"),
    )

    icon_filepath_3: bpy.props.StringProperty(
        name="Face Orientation",
        subtype='FILE_PATH',
        default=os.path.join(os.path.dirname(__file__), "..", "icons", "confirm_icon.png"),
    )

    icon_filepath_4: bpy.props.StringProperty(
        name="Button_low",
        subtype='FILE_PATH',
        default=os.path.join(os.path.dirname(__file__), "..", "icons", "op_randomize.png"),
    )

    icon_filepath_5: bpy.props.StringProperty(
        name="Button_high",
        subtype='FILE_PATH',
        default=os.path.join(os.path.dirname(__file__), "..", "icons", "op_align_center.png"),
    )

    icon_filepath_6: bpy.props.StringProperty(
        name="Delete Suffix",
        subtype='FILE_PATH',
        default=os.path.join(os.path.dirname(__file__), "..", "icons", "op_align_vertical.png"),
    )

    icon_filepath_7: bpy.props.StringProperty(
        name="Remove Text Button",
        subtype='FILE_PATH',
        default=os.path.join(os.path.dirname(__file__), "..", "icons", "op_align_bottomright.png"),
    )
```

```

icon_filepath_8: bpy.props.StringProperty(
    name="Add Text",
    subtype='FILE_PATH',
    default=os.path.join(os.path.dirname(__file__), "..", "icons", "op_island_align_world.png"),
)

icon_filepath_9: bpy.props.StringProperty(
    name="Remove Text",
    subtype='FILE_PATH',
    default=os.path.join(os.path.dirname(__file__), "..", "icons", "op_meshtex_trim.png"),
)

def draw(self, context):
    layout = self.layout
    layout.prop(self, "icon_filepath_1")
    layout.prop(self, "icon_filepath_2")
    layout.prop(self, "icon_filepath_3")
    layout.prop(self, "icon_filepath_4")
    layout.prop(self, "icon_filepath_5")
    layout.prop(self, "icon_filepath_6")
    layout.prop(self, "icon_filepath_7")
    layout.prop(self, "icon_filepath_8")
    layout.prop(self, "icon_filepath_9")

```

panels\my_paned.py

```

import bpy
import os

bpy.types.Scene.my_custom_text = bpy.props.StringProperty(name="Custom Text", default="")

from ..operators.create_and_apply_material import OBJECT_OT_create_and_apply_material
from ..operators.remove_material import OBJECT_OT_remove_material
from ..operators.toggle_face_orientation import OBJECT_OT_toggle_face_orientation
from ..operators.add_low_suffix import OBJECT_OT_add_low_suffix
from ..operators.add_high_suffix import OBJECT_OT_add_high_suffix
from ..operators.delete_all_suffix import OBJECT_OT_delete_all_suffix
from ..operators.add_text_to_selected import OBJECT_OT_add_text_to_selected
from ..operators.remove_text_from_selected import OBJECT_OT_remove_text_from_selected
from ..operators.clear_custom_text import OBJECT_OT_clear_custom_text

loaded_icons = {}

class MyPanel(bpy.types.Panel):
    bl_label = "Easy Tool"
    bl_idname = "VIEW3D_PT_my_panel"
    bl_space_type = 'VIEW_3D'
    bl_region_type = 'UI'

```

		М. О. ТИХОХОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.Б.МАРЧУК				65
Змн.	Арк.	№ докум.	Підпис	Дата		

```
bl_category = 'Easy Tool'
```

```
def draw(self, context):  
    layout = self.layout
```

```
addon_prefs = bpy.context.preferences.addons.get("my_simple_addon")  
if addon_prefs:  
    addon_prefs = addon_prefs.preferences  
else:  
    print("Addon 'my_simple_addon' not found in preferences.")
```

```
icon_filepath_1 = addon_prefs.icon_filepath_1  
icon_filepath_2 = addon_prefs.icon_filepath_2  
icon_filepath_3 = addon_prefs.icon_filepath_3  
icon_filepath_4 = addon_prefs.icon_filepath_4  
icon_filepath_5 = addon_prefs.icon_filepath_5  
icon_filepath_6 = addon_prefs.icon_filepath_6  
icon_filepath_7 = addon_prefs.icon_filepath_7  
icon_filepath_8 = addon_prefs.icon_filepath_8  
icon_filepath_9 = addon_prefs.icon_filepath_9
```

```
row = layout.row()  
row.alignment = 'CENTER'  
row.label(text="Material")
```

```
box = layout.box()
```

```
if icon_filepath_1 and os.path.exists(icon_filepath_1):  
    if icon_filepath_1 not in loaded_icons:  
        loaded_icons[icon_filepath_1] = bpy.utils.previews.new()  
        loaded_icons[icon_filepath_1].load("icon_1", icon_filepath_1, 'IMAGE')  
    row = box.row(align=True)  
    row.operator(OBJECT_OT_create_and_apply_material.bl_idname, text="Create Material",  
icon_value=loaded_icons[icon_filepath_1]["icon_1"].icon_id)
```

```
if icon_filepath_2 and os.path.exists(icon_filepath_2):  
    if icon_filepath_2 not in loaded_icons:  
        loaded_icons[icon_filepath_2] = bpy.utils.previews.new()  
        loaded_icons[icon_filepath_2].load("icon_2", icon_filepath_2, 'IMAGE')  
    row = box.row(align=True)  
    row.operator(OBJECT_OT_remove_material.bl_idname, text="Remove Material",  
icon_value=loaded_icons[icon_filepath_2]["icon_2"].icon_id)
```

```
if icon_filepath_3 and os.path.exists(icon_filepath_3):  
    if icon_filepath_3 not in loaded_icons:  
        loaded_icons[icon_filepath_3] = bpy.utils.previews.new()
```

		М. О. ТИХОХОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				66
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        loaded_icons[icon_filepath_3].load("icon_3", icon_filepath_3, 'IMAGE')
        row = box.row(align=True)
        row.operator(OBJECT_OT_toggle_face_orientation.bl_idname, text="Face Orientation",
icon_value=loaded_icons[icon_filepath_3]["icon_3"].icon_id)

        row = layout.row(align=True)
        row.alignment = 'CENTER'
        row.label(text="Suffix")

        box = layout.box()

        if icon_filepath_4 and os.path.exists(icon_filepath_4):
            if icon_filepath_4 not in loaded_icons:
                loaded_icons[icon_filepath_4] = bpy.utils.previews.new()
                loaded_icons[icon_filepath_4].load("icon_4", icon_filepath_4, 'IMAGE')
            row = box.row(align=True)
            op = row.operator(OBJECT_OT_add_low_suffix.bl_idname, text="_low",
icon_value=loaded_icons[icon_filepath_4]["icon_4"].icon_id)
            op.my_low_suffix = True

        if icon_filepath_5 and os.path.exists(icon_filepath_5):
            if icon_filepath_5 not in loaded_icons:
                loaded_icons[icon_filepath_5] = bpy.utils.previews.new()
                loaded_icons[icon_filepath_5].load("icon_5", icon_filepath_5, 'IMAGE')

            op = row.operator(OBJECT_OT_add_high_suffix.bl_idname, text="_high",
icon_value=loaded_icons[icon_filepath_5]["icon_5"].icon_id)
            op.my_high_suffix = True

        if icon_filepath_6 and os.path.exists(icon_filepath_6):
            if icon_filepath_6 not in loaded_icons:
                loaded_icons[icon_filepath_6] = bpy.utils.previews.new()
                loaded_icons[icon_filepath_6].load("icon_6", icon_filepath_6, 'IMAGE')
            row = box.row(align=True)
            row.operator(OBJECT_OT_delete_all_suffix.bl_idname, text="Delete Suffix",
icon_value=loaded_icons[icon_filepath_6]["icon_6"].icon_id)

            row = box.row(align=True)
            row.alignment = 'CENTER'
            row.label(text="Add Custom Text")

            row = box.row(align=True)
            row.label(text="Add Text:")
            row.prop(context.scene, "my_custom_text", text="")

```

		М. О. ТИХОХОВ		
		Г.В.МАРЧУК		
Змн.	Арк.	№ докум.	Підпис	Дата

```

if icon_filepath_7 and os.path.exists(icon_filepath_7):
    if icon_filepath_7 not in loaded_icons:
        loaded_icons[icon_filepath_7] = bpy.utils.previews.new()
        loaded_icons[icon_filepath_7].load("icon_7", icon_filepath_7, 'IMAGE')
    row.operator(OBJECT_OT_clear_custom_text.bl_idname, text="",
icon_value=loaded_icons[icon_filepath_7]["icon_7"].icon_id)

if icon_filepath_8 and os.path.exists(icon_filepath_8):
    if icon_filepath_8 not in loaded_icons:
        loaded_icons[icon_filepath_8] = bpy.utils.previews.new()
        loaded_icons[icon_filepath_8].load("icon_8", icon_filepath_8, 'IMAGE')
    row = box.row(align=True)
    row.operator(OBJECT_OT_add_text_to_selected.bl_idname, text="Add Text",
icon_value=loaded_icons[icon_filepath_8]["icon_8"].icon_id)

if icon_filepath_9 and os.path.exists(icon_filepath_9):
    if icon_filepath_9 not in loaded_icons:
        loaded_icons[icon_filepath_9] = bpy.utils.previews.new()
        loaded_icons[icon_filepath_9].load("icon_9", icon_filepath_9, 'IMAGE')
    row.operator(OBJECT_OT_remove_text_from_selected.bl_idname, text="Remove Text",
icon_value=loaded_icons[icon_filepath_9]["icon_9"].icon_id)

```

operators\add_high_suffix.py

```

import bpy

class OBJECT_OT_add_high_suffix(bpy.types.Operator):
    bl_idname = "object.add_high_suffix"
    bl_label = "Add_high Suffix"
    bl_options = {'REGISTER', 'UNDO'}
    bl_icon = 'MESH_MONKEY'

    my_high_suffix: bpy.props.BoolProperty()

    def execute(self, context):
        selected_objects = bpy.context.selected_objects

        for obj in selected_objects:

            if self.my_high_suffix:
                obj.name += "_high"

            self.report({'INFO'}, "Added '_high' suffix to the selected objects' names.")
            return {'FINISHED'}

    def register():
        bpy.utils.register_class(OBJECT_OT_add_high_suffix)

    def unregister():
        bpy.utils.unregister_class(OBJECT_OT_add_high_suffix)

```

operators\add_low_suffix.py

		М. О. ТИХОХОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.Б.МАРЧУК				68
Змн.	Арк.	№ докум.	Підпис	Дата		

```

import bpy

class OBJECT_OT_add_low_suffix(bpy.types.Operator):
    bl_idname = "object.add_low_suffix"
    bl_label = "Add _low Suffix"
    bl_options = {'REGISTER', 'UNDO'}
    bl_icon = 'MESH_ICOSPHERE'

    my_low_suffix: bpy.props.BoolProperty()

    def execute(self, context):
        selected_objects = bpy.context.selected_objects

        for obj in selected_objects:

            if self.my_low_suffix:
                obj.name += "_low"

        self.report({'INFO'}, "Added '_low' suffix to the selected objects' names.")
        return {'FINISHED'}

    def register():
        bpy.utils.register_class(OBJECT_OT_add_low_suffix)

    def unregister():
        bpy.utils.unregister_class(OBJECT_OT_add_low_suffix)
operators\add_text_to_selected.py

import bpy

class OBJECT_OT_add_text_to_selected(bpy.types.Operator):
    bl_idname = "object.add_text_to_selected"
    bl_label = "Add Text to Selected"
    bl_options = {'REGISTER', 'UNDO'}
    bl_icon = 'TEXT'

    def execute(self, context):
        text_to_add = context.scene.my_custom_text
        selected_objects = context.selected_objects

        for obj in selected_objects:

            if text_to_add.endswith('_'):
                obj.name += text_to_add
            else:
                obj.name += text_to_add

        self.report({'INFO'}, f"Added '{text_to_add}' to the selected objects' names.")
        return {'FINISHED'}

    def register():
        bpy.utils.register_class(OBJECT_OT_add_text_to_selected)

```

		М. О. ТИХОХОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.Б.МАРЧУК				69
Змн.	Арк.	№ докум.	Підпис	Дата		

```

def unregister():
    bpy.utils.unregister_class(OBJECT_OT_add_text_to_selected)
                                operators\clear_custom_text.py

import bpy

class OBJECT_OT_clear_custom_text(bpy.types.Operator):
    bl_idname = "object.clear_custom_text"
    bl_label = "Clear Custom Text"
    bl_options = {'REGISTER'}

    def execute(self, context):
        context.scene.my_custom_text = ""
        return {'FINISHED'}

def register():
    bpy.utils.register_class(OBJECT_OT_clear_custom_text)

def unregister():
    bpy.utils.unregister_class(OBJECT_OT_clear_custom_text)
                                operators\create_and_apply_material.py

import bpy

class OBJECT_OT_create_and_apply_material(bpy.types.Operator):
    bl_idname = "object.create_and_apply_material"
    bl_label = "Create and Apply Material"
    bl_options = {'REGISTER', 'UNDO'}
    bl_icon = 'MATERIAL'

    def execute(self, context):
        selected_objects = bpy.context.selected_objects

        for obj in selected_objects:

            mat = bpy.data.materials.new(name=obj.name)
            mat.use_nodes = True
            obj.data.materials.append(mat)

            self.report({'INFO'}, "Created and applied materials to the selected objects based on their names.")
            return {'FINISHED'}

def register():
    bpy.utils.register_class(OBJECT_OT_create_and_apply_material)

def unregister():
    bpy.utils.unregister_class(OBJECT_OT_create_and_apply_material)

if __name__ == "__main__":
    register()
                                operators\delete_all_suffix.py

```

		М. О. ТИХОХОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.Б.МАРЧУК				70
Змн.	Арк.	№ докум.	Підпис	Дата		

```

import bpy

class OBJECT_OT_delete_all_suffix(bpy.types.Operator):
    bl_idname = "object.delete_all_suffix"
    bl_label = "Delete all Suffix"
    bl_options = {'REGISTER', 'UNDO'}
    bl_icon = 'X'

    def execute(self, context):
        selected_objects = bpy.context.selected_objects

        for obj in selected_objects:

            obj.name = obj.name.split("_low")[0].split("_high")[0]

        self.report({'INFO'}, "Deleted all suffixes from the selected objects' names.")
        return {'FINISHED'}

    def register():
        bpy.utils.register_class(OBJECT_OT_delete_all_suffix)

    def unregister():
        bpy.utils.unregister_class(OBJECT_OT_delete_all_suffix)
operators\remove_material.py

import bpy

class OBJECT_OT_remove_material(bpy.types.Operator):
    bl_idname = "object.remove_material"
    bl_label = "Remove Material"
    bl_options = {'REGISTER', 'UNDO'}
    bl_icon = 'CANCEL'

    def execute(self, context):
        selected_objects = bpy.context.selected_objects

        for obj in selected_objects:

            obj.active_material_index = 0
            bpy.ops.object.material_slot_remove({'object': obj})

        for material in bpy.data.materials:
            if not material.users:
                bpy.data.materials.remove(material)

        self.report({'INFO'}, "Removed material from the selected objects and deleted unused materials.")
        return {'FINISHED'}

    def register():
        bpy.utils.register_class(OBJECT_OT_remove_material)

```

```

def unregister():
    bpy.utils.unregister_class(OBJECT_OT_remove_material)
                                operators\remove_text_from_selected.py

import bpy

class OBJECT_OT_remove_text_from_selected(bpy.types.Operator):
    bl_idname = "object.remove_text_from_selected"
    bl_label = "Remove Text from Selected"
    bl_options = {'REGISTER', 'UNDO'}
    bl_icon = 'TEXT'

    def execute(self, context):
        text_to_remove = context.scene.my_custom_text
        selected_objects = context.selected_objects

        for obj in selected_objects:
            obj.name = obj.name.replace(text_to_remove, "")

        self.report({'INFO'}, f"Removed '{text_to_remove}' from the selected objects' names.")
        return {'FINISHED'}

def register():
    bpy.utils.register_class(OBJECT_OT_remove_text_from_selected)

def unregister():
    bpy.utils.unregister_class(OBJECT_OT_remove_text_from_selected)
                                operators\toggle_face_orientation.py

import bpy

class OBJECT_OT_toggle_face_orientation(bpy.types.Operator):
    bl_idname = "object.toggle_face_orientation"
    bl_label = "Face Orientation"
    bl_options = {'REGISTER', 'UNDO'}
    bl_icon = 'FACESEL'

    def execute(self, context):
        for area in context.screen.areas:
            if area.type == 'VIEW_3D':
                for space in area.spaces:
                    if space.type == 'VIEW_3D':
                        space.overlay.show_face_orientation = not space.overlay.show_face_orientation
        return {'FINISHED'}

def register():
    bpy.utils.register_class(OBJECT_OT_toggle_face_orientation)

def unregister():
    bpy.utils.unregister_class(OBJECT_OT_toggle_face_orientation)

#include "node_shader_util.hh"

```

Додаток Б

		М. О. ТИХОХОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				72
Змн.	Арк.	№ докум.	Підпис	Дата		

```

#include "node_util.hh"

#include "BKE_image.h"
#include "BKE_node_runtime.hh"
#include "BKE_texture.h"

#include "IMB_colormanagement.hh"

#include "DEG_depsgraph_query.hh"

namespace blender::nodes::node_shader_tex_image_cc {

static void sh_node_tex_image_declare(NodeDeclarationBuilder &b)
{
    b.is_function_node();
    b.add_input<decl::Vector>("Vector").implicit_field(implicit_field_inputs::position);
    b.add_output<decl::Color>("Color").no_muted_links();
    b.add_output<decl::Float>("Alpha").no_muted_links();
}

static void node_shader_init_tex_image(bNodeTree * /*ntree*/, bNode *node)
{
    NodeTexImage *tex = MEM_cnew<NodeTexImage>(__func__);
    BKE_texture_mapping_default(&tex->base.tex_mapping, TEXMAP_TYPE_POINT);
    BKE_texture_colormapping_default(&tex->base.color_mapping);
    BKE_imageuser_default(&tex->iuser);

    node->storage = tex;
}

static int node_shader_gpu_tex_image(GPUMaterial *mat,
                                     bNode *node,
                                     bNodeExecData *,
                                     GPUNodeStack *in,
                                     GPUNodeStack *out)
{
    Image *ima = (Image *)node->id;
    NodeTexImage *tex = (NodeTexImage *)node->storage;

    bNode *node_original = node->runtime->original ? node->runtime->original : node;
    NodeTexImage *tex_original = (NodeTexImage *)node_original->storage;
    ImageUser *iuser = &tex_original->iuser;

    if (!ima) {
        return GPU_stack_link(mat, node, "node_tex_image_empty", in, out);
    }

    GPUNodeLink **texco = &in[0].link;
    if (!*texco) {
        *texco = GPU_attribute(mat, CD_AUTO_FROM_NAME, "");
        node_shader_gpu_bump_tex_coord(mat, node, texco);
    }
}

```

		М. О. ТИХОХОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.Б.МАРЧУК				73
Змн.	Арк.	№ докум.	Підпис	Дата		

```

}

node_shader_gpu_tex_mapping(mat, node, in, out);

GPUSamplerState sampler_state = GPUSamplerState::default_sampler();

switch (tex->extension) {
case SHD_IMAGE_EXTENSION_EXTEND:
    sampler_state.extend_x = GPU_SAMPLER_EXTEND_MODE_EXTEND;
    sampler_state.extend_yz = GPU_SAMPLER_EXTEND_MODE_EXTEND;
    break;
case SHD_IMAGE_EXTENSION_REPEAT:
    sampler_state.extend_x = GPU_SAMPLER_EXTEND_MODE_REPEAT;
    sampler_state.extend_yz = GPU_SAMPLER_EXTEND_MODE_REPEAT;
    break;
case SHD_IMAGE_EXTENSION_CLIP:
    sampler_state.extend_x = GPU_SAMPLER_EXTEND_MODE_CLAMP_TO_BORDER;
    sampler_state.extend_yz = GPU_SAMPLER_EXTEND_MODE_CLAMP_TO_BORDER;
    break;
case SHD_IMAGE_EXTENSION_MIRROR:
    sampler_state.extend_x = GPU_SAMPLER_EXTEND_MODE_MIRRORED_REPEAT;
    sampler_state.extend_yz = GPU_SAMPLER_EXTEND_MODE_MIRRORED_REPEAT;
    break;
default:
    break;
}

if (tex->interpolation != SHD_INTERP_CLOSEST) {
    sampler_state.filtering = GPU_SAMPLER_FILTERING_ANISOTROPIC |
GPU_SAMPLER_FILTERING_LINEAR |
        GPU_SAMPLER_FILTERING_MIPMAP;
}

const bool use_cubic = ELEM(tex->interpolation, SHD_INTERP_CUBIC, SHD_INTERP_SMART);

if (ima->source == IMA_SRC_TILED) {
    const char *gpu_node_name = use_cubic ? "node_tex_tile_cubic" : "node_tex_tile_linear";
    GPUNodeLink *gpu_image, *gpu_image_tile_mapping;
    GPU_image_tiled(mat, ima, iuser, sampler_state, &gpu_image, &gpu_image_tile_mapping);
    GPU_stack_link(mat, node, gpu_node_name, in, out, gpu_image, gpu_image_tile_mapping);
}
else {
    const char *gpu_node_name = use_cubic ? "node_tex_image_cubic" : "node_tex_image_linear";

    switch (tex->projection) {
    case SHD_PROJ_FLAT: {
        GPUNodeLink *gpu_image = GPU_image(mat, ima, iuser, sampler_state);
        GPU_stack_link(mat, node, gpu_node_name, in, out, gpu_image);
        break;
    }
    case SHD_PROJ_BOX: {
        gpu_node_name = use_cubic ? "tex_box_sample_cubic" : "tex_box_sample_linear";
    }
    }
}

```

		М. О. ТИХОХОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.Б.МАРЧУК				74
Змн.	Арк.	№ докум.	Підпис	Дата		

```

GPUNodeLink *vnor, *wnor, *col1, *col2, *col3;
GPUNodeLink *blend = GPU_uniform(&tex->projection_blend);
GPUNodeLink *gpu_image = GPU_image(mat, ima, iuser, sampler_state);
GPU_link(mat, "world_normals_get", &vnor);
GPU_link(mat, "normal_transform_world_to_object", vnor, &wnor);
GPU_link(mat, gpu_node_name, in[0].link, wnor, gpu_image, &col1, &col2, &col3);
GPU_link(mat, "tex_box_blend", wnor, col1, col2, col3, blend, &out[0].link, &out[1].link);
break;
}

case SHD_PROJ_SPHERE: {
    sampler_state.disable_filtering_flag(GPU_SAMPLER_FILTERING_MIPMAP);
    GPUNodeLink *gpu_image = GPU_image(mat, ima, iuser, sampler_state);
    GPU_link(mat, "point_texco_remap_square", *texco, texco);
    GPU_link(mat, "point_map_to_sphere", *texco, texco);
    GPU_stack_link(mat, node, gpu_node_name, in, out, gpu_image);
    break;
}

case SHD_PROJ_TUBE: {
    sampler_state.disable_filtering_flag(GPU_SAMPLER_FILTERING_MIPMAP);
    GPUNodeLink *gpu_image = GPU_image(mat, ima, iuser, sampler_state);
    GPU_link(mat, "point_texco_remap_square", *texco, texco);
    GPU_link(mat, "point_map_to_tube", *texco, texco);
    GPU_stack_link(mat, node, gpu_node_name, in, out, gpu_image);
    break;
}
}
}

if (out[0].hasoutput) {
    if (ELEM(ima->alpha_mode, IMA_ALPHA_IGNORE, IMA_ALPHA_CHANNEL_PACKED) ||
        IMB_colormangement_space_name_is_data(ima->colorspace_settings.name))
    {
        GPU_link(mat, "color_alpha_clear", out[0].link, &out[0].link);
    }
    else {
        if (ima->alpha_mode == IMA_ALPHA_PREMUL) {
            if (out[1].hasoutput) {
                GPU_link(mat, "color_alpha_unpremultiply", out[0].link, &out[0].link);
            }
            else {
                GPU_link(mat, "color_alpha_clear", out[0].link, &out[0].link);
            }
        }
        else {
            if (out[1].hasoutput) {
                GPU_link(mat, "color_alpha_clear", out[0].link, &out[0].link);
            }
            else {
                GPU_link(mat, "color_alpha_premultiply", out[0].link, &out[0].link);
            }
        }
    }
}

```

		М. О. ТИХОХОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.Б.МАРЧУК				75
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
}

return true;
}

NODE_SHADER_MATERIALX_BEGIN
#ifdef WITH_MATERIALX
{
    std::string image_node_name = node_name(false) + "_Color";

    NodeItem res = empty();
    res.node = graph_>getNode(image_node_name);
    if (!res.node) {
        res = val(MaterialX::Color4(1.0f, 0.0f, 1.0f, 1.0f));

        Image *image = (Image *)node_>id;
        if (image) {
            NodeTexImage *tex_image = static_cast<NodeTexImage*>(node_>storage);

            std::string image_path = image->id.name;
            if (export_image_fn_) {
                Scene *scene = DEG_get_input_scene(depsgraph_);
                Main *bmain = DEG_get_bmain(depsgraph_);
                image_path = export_image_fn_(bmain, scene, image, &tex_image->iuser);
            }

            NodeItem vector = get_input_link("Vector", NodeItem::Type::Vector2);
            if (!vector) {
                vector = texcoord_node();
            }

            std::string filtertype;
            switch (tex_image->interpolation) {
                case SHD_INTERP_LINEAR:
                    filtertype = "linear";
                    break;
                case SHD_INTERP_CLOSEST:
                    filtertype = "closest";
                    break;
                case SHD_INTERP_CUBIC:
                case SHD_INTERP_SMART:
                    filtertype = "cubic";
                    break;
                default:
                    BLI_assert_unreachable();
            }
            std::string addressmode;
            switch (tex_image->extension) {
                case SHD_IMAGE_EXTENSION_REPEAT:
                    addressmode = "periodic";

```

		М. О. ТИХОБОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.Б.МАРЧУК				76
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        break;
    case SHD_IMAGE_EXTENSION_EXTEND:
        addressmode = "clamp";
        break;
    case SHD_IMAGE_EXTENSION_CLIP:
        addressmode = "constant";
        break;
    case SHD_IMAGE_EXTENSION_MIRROR:
        addressmode = "mirror";
        break;
    default:
        BLI_assert_unreachable();
}

res = create_node("image",
    NodeItem::Type::Color4,
    {"texcoord", vector},
    {"filtertype", val(filtertype)},
    {"uaddressmode", val(addressmode)},
    {"vaddressmode", val(addressmode)});
res.set_input("file", image_path, NodeItem::Type::Filename);
res.node->setName(image_node_name);
}
}

if (STREQ(socket_out->name, "Alpha")) {
    res = res[3];
}
return res;
}
#endif
NODE_SHADER_MATERIALX_END
}
void register_node_type_sh_tex_image()
{
    namespace file_ns = blender::nodes::node_shader_tex_image_cc;

    static blender::bke::bNodeType ntype;
    sh_node_type_base(&ntype, SH_NODE_TEX_IMAGE, "Image Texture", NODE_CLASS_TEXTURE);
    ntype.declare = file_ns::sh_node_tex_image_declare;
    ntype.initfunc = file_ns::node_shader_init_tex_image;
    blender::bke::node_type_storage(
        &ntype, "NodeTexImage", node_free_standard_storage, node_copy_standard_storage);
    ntype.gpu_fn = file_ns::node_shader_gpu_tex_image;
    ntype.labelfunc = node_image_label;
    blender::bke::node_type_size_preset(&ntype, blender::bke::eNodeSizePreset::Large);
    ntype.materialx_fn = file_ns::node_shader_materialx;

    blender::bke::nodeRegisterType(&ntype);
}

```

3D Модель дерев'яного жука без текстур



3D Модель дерев'яного жука з текстурою



3D Модель лісового охоронця без текстур



3D Модель лісового охоронця з текстурами



3D воїна ріпи з маскою без текстур



3D воїна ріпи з маскою з текстурами



3D Модель лісового тролля без текстур



3D Модель лісового тролля з текстурами



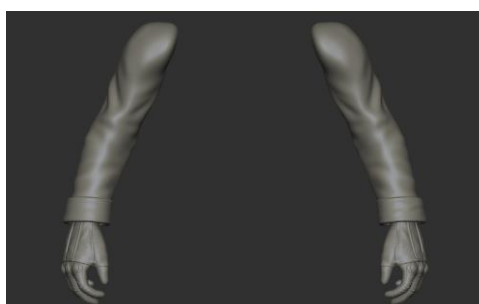
3D Модель лісового короля без текстур



3D Модель лісового короля з текстурами



3D Модель рук гравця без текстур

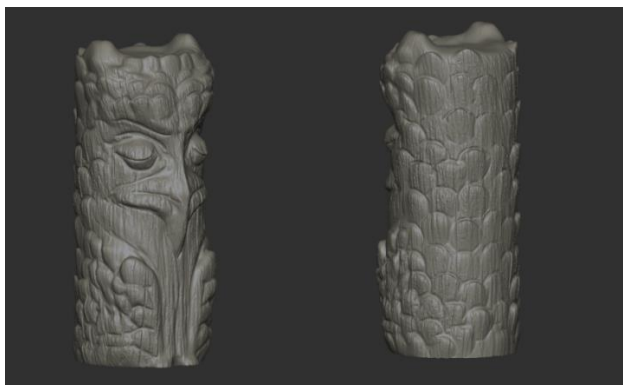


		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				80
Змн.	Арк.	№ докум.	Підпис	Дата		

3D Модель рук гравця з текстурами



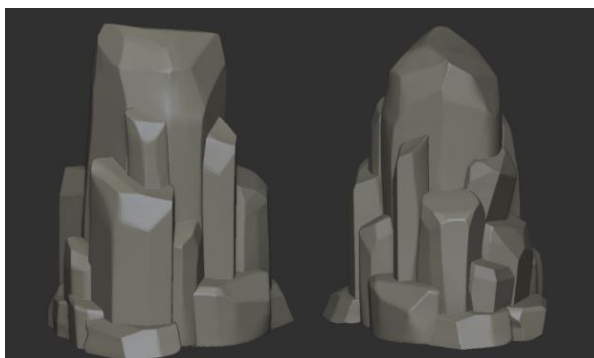
3D Модель Тотема Соби без текстур



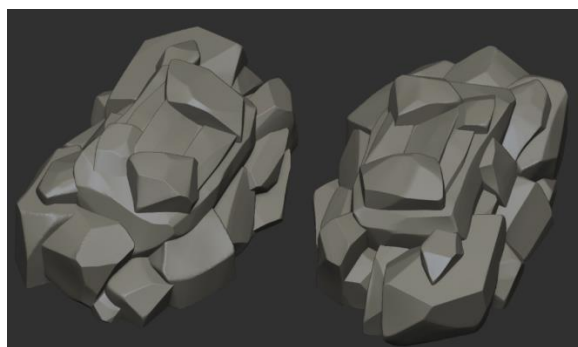
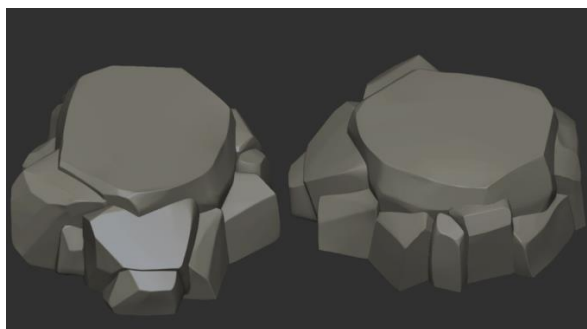
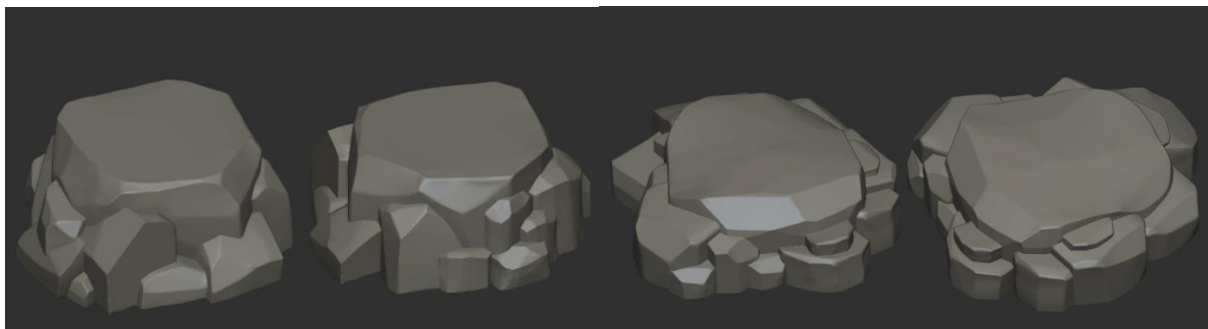
3D Модель Тотема Соби з текстурами



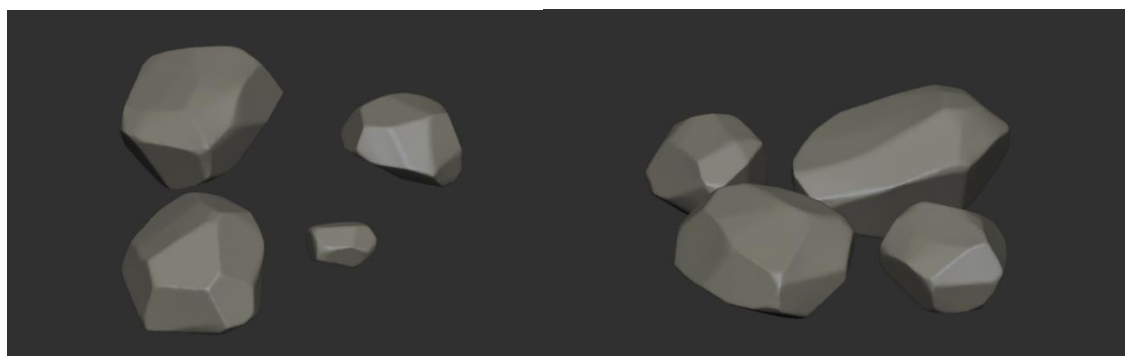
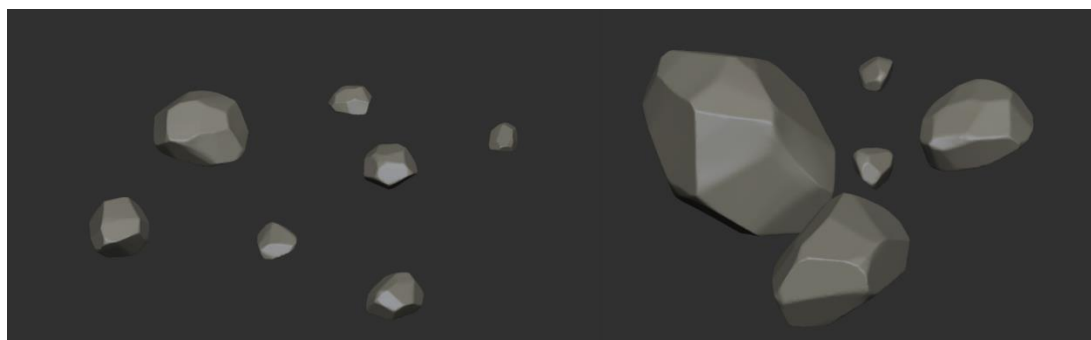
3D Моделі великих каменів без текстур



		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				
Змн.	Арк.	№ докум.	Підпис	Дата		81



3D Моделі маленьких каменів без текстур



		М. О. ТИХОНОВ			КН.КР.Б – 122 – 24 – ПЗ	Арк.
		Г.В.МАРЧУК				82
Змн.	Арк.	№ докум.	Підпис	Дата		