

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук

Кафедра інформатики

ДИПЛОМНА РОБОТА

Рівень вищої освіти бакалавр

на тему: Моделювання та розробка інтерактивного ігрового
середовища

Виконав студент 4 курсу групи К-42

Напрямок підготовки 6.050101

Комп'ютерні науки

Витичак Микола Іванович

Керівник к. т. н., доцент

Гнатовська Ганна Арнольдівна

Рецензент к. геогр. н., доцент

Кузніченко Світлана Дмитрівна

Одеса 2017

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ	5
ВСТУП	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1 Основне поняття комп'ютерних ігор	9
1.2 Класифікація комп'ютерних ігор	10
1.3 Комп'ютерні ігри і навчання	11
1.4 Тривимірна графіка в комп'ютерних іграх	13
1.5 Процес створення тривимірних моделей	14
1.6 Постановка задачі	16
2 ВИБІР АРХІТЕКТУРИ ТА ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ	18
2.1 Вибір способу тривимірного моделювання	18
2.2 Вибір програмних засобів реалізації моделювання	21
2.3 Методи рендерингу зображення	32
2.4 Вибір ігрового рушія для тестування тривимірних моделей	35
3 ПРОЕКТУВАННЯ ТРИВИМІРНИХ МОДЕЛЕЙ	41
3.1 Проектування ігрового простору	41
3.2 Проектування тривимірних моделей оточення	42
3.3 Проектування моделі ігрового персонажа	43
4 МОДЕЛЮВАННЯ ТРИВИМІРНОГО ПРОСТОРУ	45
4.1 Створення моделей ігрового простору	45
4.2 Створення моделі ігрового персонажа	56
ВИСНОВКИ	61
ПЕРЕЛІК ПОСИЛАНЬ	62

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ

Скорочення

TCP/IP – Transmission Control Protocol and Internet Protocol – Протокол керування передачею та Інтернет протокол.

Tcl – Tool Command Language – Язык Команд Інструментів.

NURBS – Non-uniform rational B-spline – неоднорідний раціональний B-сплайн.

MB – модуль візуалізації.

MEL – Maya Embedded Language – Вбудована Мова Майа.

IBM – International Business Machines Corporation – Міжнародна Корпорація Бізнес Машин.

API – Application Programming Interface – Програмний Інтерфейс Додатку.

POPs – Particle Operators – оператор частинок.

SHOPs – Shading Operators – оператор шейдингу.

COPs – Composite Operators – оператор композитингу.

DOPs – Dynamic Operators – оператор динаміки.

ROPs – Render Operators – оператор рендерингу.

SVN – Apache Subversion.

SSAO – Screen Space Ambient Occlusion – перегородження навколишнього світла в екранному просторі.

GLSL – OpenGL Shader Language – Мова OpenGL шейдерів.

Терміни

Вершина (Vertex) – абстрактна геометрична точка в просторі має координати X, Y, Z. Вершиною вона називається тому що являє крайньою точкою полігону.

Грань (з англ. Face) – площина, що з'єднує вершини в просторі, зазвичай нагадує трикутник, чотирикутник або інші прості опуклі багатокутники, так як це спрощує рендеринг.

Карта нормалей – це текстура, колірна інформація якої зчитується як інформація про розташування нормалі кожної точки того чи іншого об'єкта.

Полігон (з англ. Poly, Polygon) – основна функціональна складова тривимірної моделі. Є плоскою прямокутною фігурою в просторі, з безлічі

яких створена тривимірна модель, тобто набір, що лежить в одній площині граней.

Полігональна сітка (з англ. Mesh) – це сукупність вершин, ребер і граней, які визначають форму багатогранного об'єкта в тривимірній комп'ютерній графіці і об'ємному моделюванні.

Ребро (з англ. Edge) – відрізок прямої, з'єднує пару вершин.

Текстура – зображення, яке покриває поверхню тривимірної моделі, що використовується для додання фігурі матеріальної достовірності.

Тривимірна графіка – розділ комп'ютерної графіки, присвячений методам створення зображень або відео шляхом моделювання об'ємних об'єктів в тривимірному просторі.

RAW (англ. Raw – сирий, необроблений) – формат даних, що містить необроблені (або оброблені в мінімальному ступені) дані, що дозволяє уникнути втрат інформації, і не дає чіткої специфікації.

ВСТУП

В наш час комп'ютерні ігри дуже популярні, а дохід від них досить значний. Для одних ігри це хобі, для інших це робота (професійні гравці, розробники комп'ютерних ігор).

Комп'ютерні ігри – це програми, призначені для розваги людей на комп'ютері. Такі ігри отримали надзвичайно потужний розвиток і вилилися в величезний ринок за останні тридцять років. Багато людей годинами і цілодобово сидять за комп'ютером, граючи в різного роду шутери, MMORPG, RPG, стратегії, симулятори та інші жанри комп'ютерних ігор. Чи не менше людей готові платити за такого роду розваги. Це весело і цікаво, захоплює і викликає залежність. З розвитком комп'ютерної електроніки і поліпшенням програмного і апаратного забезпечення, ігри стають потужнішими і реалістичніше, все більше наближаючись за якістю візуалізації до реального світу [1].

Комп'ютерні ігри можуть створюватися на основі фільмів і книг; є і зворотні випадки. З 2011 року комп'ютерні ігри офіційно визнані в США окремим видом мистецтва.

В даний час в ряді випадків замість «комп'ютерна гра» може використовуватися термін «відеогра», тобто ці терміни можуть вживатися як синоніми і взаємозамінними.

Комп'ютерні ігри надали настільки істотний вплив на суспільство, що в інформаційних технологіях відзначена стійка тенденція до гейміфікації для неігрового прикладного програмного забезпечення.

Сучасні ігри включають в себе такі важливі компоненти як візуальна складова (змодельованої ігровий простір), звукові ефекти, привабливий геймплей і сюжет.

Сучасність надає різний вибір ігор на будь-який смак, тим самим дає можливість зрозуміти людині, який саме вид ігор їй симпатизує і підходить під вид її мислення, задовольняє її бажання і певні потреби. Барвистість, інтерфейс, сюжет деяких ігор досі вражає уяви користувачів комп'ютерів. Віртуальний світ може надовго поглинути гравця, змушуючи його забути про буденні проблеми. Саме тому з недавніх пір ігри почали приписувати до виду своєрідного відпочинку від зовнішнього світу. Людина може запросто вибрати те, що для нього буде найбільш легким, приємним і цікавим, таким чином, присвячуючи себе на час цьому хобі [2].

Метою дипломної роботи є здійснення моделювання та розробки інтерактивного ігрового середовища, за рахунок створення візуальної складової за допомогою різних редакторів тривимірної графіки на базі яких можна буде створити повноцінну ігрову програму. Для цього, насамперед, необхідно проаналізувати предметну область і вибрати відповідний редактор для поставлених цілей. Після чого спроектувати прототип моделі і виконати її програмну реалізацію.

Дипломна робота містить в собі 60 сторінок, 41 рисунок та 12 посилань.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Основне поняття комп'ютерних ігор

Комп'ютерна гра – комп'ютерна програма, що служить для організації ігрового процесу (геймплея), зв'язку з партнерами по грі, або, як правило, ігрова ситуація відтворюється на екрані дисплея або звичайного телевізора (в цьому випадку комп'ютерні ігри одночасно є і відеоіграми), але в той же час комп'ютерні ігри можуть бути звукові, телетайпові чи інші.

Основні компоненти гри – уявна ситуація, роль і реалізуючі її ігрові дії, а також ролі, узяті на себе граючими; ігрові дії як засіб реалізації цих ролей; ігрове використання предметів, тобто заміщення реальних предметів ігровими, умовними; реальні відносини між граючими.

Сучасні ігри не настільки прості з технічної сторони як може здатися. Більшість з них в наш час є складними і добре структурованими системами взаємодії гравця з навколишнім світом, що містять в собі десятки тисяч пов'язаних між собою тривимірних моделей або спрайтів, скриптів і ігрових об'єктів, що супроводжуються звуковим і візуальним супроводом. Велика частина ігор має тривимірне віртуальне середовище в якій гравець постійно приймає будь-які рішення дотримуючись плану розробників або рухаючись по сюжетній лінії.

Для розуміння того, як влаштовані тривимірні ігри з технічного боку необхідно розглянути такі поняття:

Вершина (з англ. Vertex) – абстрактна геометрична точка в просторі має координати X , Y , Z . Вершиною вона називається тому що являє крайньою точкою полігону.

Ребро (з англ. Edge) – відрізок прямої, з'єднує пару вершин.

Грань (з англ. Face) – площина, що з'єднує вершини в просторі, зазвичай нагадує трикутник, чотирикутник або інші прості опуклі багатокутники, так як це спрощує рендеринг.

Полігон (з англ. Poly, Polygon) – основна функціональна складова тривимірної моделі. Є плоскою прямокутною фігурою в просторі, з безлічі яких створена тривимірна модель, тобто набір, що лежить в одній площині граней.

Полігональна сітка (з англ. Mesh) – це сукупність вершин, ребер і граней, які визначають форму багатогранного об'єкта в тривимірній комп'ютерній графіці і об'ємному моделюванні.

Текстура – зображення, яке покриває поверхню тривимірної моделі, що використовується для додання фігурі матеріальної достовірності.

Нормаль і карта нормалей – це вектор, перпендикулярний поверхні в кожній її точці. Карта нормалей – це текстура, колірна інформація якої зчитується як інформація про розташування нормалі кожної точки того чи іншого об'єкта. За допомогою карти нормалей можна сформувати ілюзію більш складної поверхні (наприклад шороховатість на плоскій моделі).

1.2 Класифікація комп'ютерних ігор

Комп'ютерні ігри в основному класифікуються за жанрами, а також за кількістю гравців.

Внаслідок того, що критерії належності гри до того чи іншого жанру не визначені однозначно, класифікація комп'ютерних ігор недостатньо систематизована, і в різних джерелах дані про жанр конкретного проекту можуть розрізнятися. Проте, існує консенсус, до якого прийшли розробники ігор, і належність гри до одного з основних жанрів майже завжди можна визначити однозначно.

Існують гри з елементами кількох жанрів, які можуть належати кожному з них (наприклад, серія Grand Theft Auto, Космічні Рейнджери, Rome: Total War і багато інших). Такі проекти зараховують або до одного з жанрів, який в грі є основним, або відразу до всіх, присутніх в грі, якщо вони в рівній мірі становлять геймплей проекту [3].

Класифікація за жанрами:

- 1) Action (трансліт. – «екшен» або «стрілялки»; в перекладі з англ. – «дія») – жанр комп'ютерних ігор, в яких успіх гравця у великій мірі залежить від його швидкості реакції і здатності швидко приймати тактичні рішення. Дія таких ігор розвивається дуже динамічно і вимагає високу концентрацію уваги і швидкої реакції на події в грі події.
- 2) Симулятор – імітатор (зазвичай механічний або комп'ютерний), завдання якого полягає в імітації управління будь-яким процесом, апаратом або транспортним засобом. Найчастіше зараз слово «симулятор» використовується стосовно до комп'ютерних програм (зазвичай ігор). За допомогою комп'ютерно-механічних симуляторів, абсолютно точно відтворюють інтер'єр кабіни апарату, де тренуються пілоти, космонавти, машиністи високошвидкісних

поїздів. Симулятори – програмні і апаратні засоби, що створюють враження дійсності, відображаючи частина реальних явищ і властивостей у віртуальному середовищі.

- 3) Стратегії (англ. Strategy) – ігри, що вимагають планування і вироблення певної стратегії для досягнення якоїсь конкретної мети, наприклад, перемоги у військовій операції. Гравець управляє не одним персонажем, а цілим підрозділом, підприємством або навіть всесвіту. Розрізняють похідовою або покрокові стратегічні ігри (Turn-Based Strategy, TBS), де гравці по черзі роблять ходи, і кожному гравцеві відводиться необмежений або обмежений (в залежності від типу і складності гри) час на свій хід, і стратегічні ігри в реальному часі (Real Time Strategy, RTS), в яких всі гравці виконують свої дії одночасно, і хід часу не переривається.
- 4) Пригода (англ. Adventure) – гра-розповідь, в якій керований гравцем герой просувається по сюжету і взаємодіє з ігровим світом за допомогою застосування предметів, спілкування з іншими персонажами і рішення логічних задач.
- 5) Комп'ютерна рольова гра (англ. Computer Role-Playing Game, позначається аббревіатурою CRPG або RPG) – жанр комп'ютерних ігор, заснований на елементах ігрового процесу традиційних настільних рольових ігор. У рольовій грі гравець управляє одним або декількома персонажами, кожен з яких описаний набором чисельних характеристик, списком здібностей і вмінь; прикладами таких характеристик можуть бути хіт-пойнти (англ. hit points, HP), показники сили, спритності, захисту, ухилення, рівень розвитку того чи іншого навичку і т.п. В ході гри вони можуть змінюватися. Одним з характерних елементів ігрового процесу є підвищення можливостей персонажів за рахунок поліпшення їх параметрів і вивчення нових здібностей.
- 6) Традиційні та настільні – комп'ютерна реалізація настільних ігор (англ. Traditional; англ. Board), наприклад шахи, карти, шашки, «Монополія», серія ігор Warhammer.

1.3 Комп'ютерні ігри і навчання

Цифрові навчальні ігри – це предмет дослідження педагогіки комп'ютерної гри, яка в свою чергу є частиною медіапедагогіки (або

медіаосвіти). Педагогіка, що відноситься до навчальної гри комп'ютерних ігор розглядає три основні завдання:

- Вона досліджує і описує роль цифрових навчальних ігор у педагогічній практиці (В школах, під час соціальної роботи, в будинках для людей похилого віку і т. Д.). Комп'ютерні ігри поза педагогічного контексту (що відносяться до педагогіки формах, поширенні, діях) і тематизація цифрових навчальних ігор в педагогічному, але не науковому дискурсі (наприклад, в медіа, в професійної медійної літературі, проектах навчальних занять, а також в довідниках по дизайну цифрових навчальних ігор)
- Вона критично ставиться до комп'ютерних ігор в педагогічній практиці і педагогічні, ненаукові висловлювання про комп'ютерні ігри. При цьому вона спирається на описові результати педагогічного дослідження комп'ютерних ігор і на форми аргументації з педагогічної етики, наприклад, філософії освіти.
- Вона займається методологічним і технічним дослідженням (комп'ютерних ігор) з метою дати науково-обгрунтовані вказівки для створення цифрових навчальних ігор і їх виховно-педагогічного використання (розвитку і оцінки). Методологічне дослідження використовує результати обох названих дослідницьких завдань (наприклад, опис дидактично-методологічного дизайну розважальних комп'ютерних ігор). Наприклад, щоб розробити наукові гіпотези.

Ці три дослідницькі завдання хоч і пов'язані один з одним, але йдуть різною дослідницькою логікою і розрізняються з системної науково-теоретичної позиції. Важливими суміжними науками комп'ютерної педагогіки є комплекс наукових дисциплін про засоби комунікації та дослідження відеоігор.

Відеоігри можуть бути також дуже корисні в тому сенсі, що вони створюють ефект симуляції дії, але при цьому не несуть будь-якої очевидної небезпеки. У симуляторі гравець може зробити висновки зі своїх помилок і перезапустити програму. Цей процес веде в результаті до високого рівня майстерності в симуляторі, а далі і при управлінні справжнім літаком, на якому пілоти літатимуть в майбутньому. Військові також використовують гри франшизи Call of Duty і Rainbow Six: Siege. Ігри такого роду занурюють гравців в віртуальну реальність. Використовуючи тактичні навички, гравці намагаються досягти в грі будь-яких, поставлених перед ними цілей. Це

дозволяє військовим показати своїм солдатам, як справлятися з конкретними ситуаціями, не ризикуючи при цьому отримати поранення на полі бою [4].

Цифрові навчальні ігри відрізняються від традиційних навчальних ігор і не заснованого на іграх електронного навчання тим, що вони використовують методи мотивації розважальних ігор, щоб досягти своїх освітніх цілей [5]. Таким чином, вони досить часто використовують будь-яку історію і парасоціальні відносини між гравцем і неігрових персонажем (англ. Non-player character)[6], щоб запустити процес навчання. На основі численних ознак, за якими навчання в групах відрізняється від індивідуального, можна також провести принципову відмінність між цифровими іграми, спрямованими на окремого учня, і тими, які націлені на навчання в групах.

1.4 Тривимірна графіка в комп'ютерних іграх

На сьогоднішній день практично жодна сучасна гра не обходиться без тривимірної графіки. Тут тривимірна графіка знайшла найбільшу популярність серед всіх областей її застосування.

Тривимірна графіка – розділ комп'ютерної графіки, присвячений методам створення зображень або відео шляхом моделювання об'ємних об'єктів в тривимірному просторі [7].

Для створення тривимірної графіки використовується величезна кількість 3D-редакторів, від програмних пакетів з широким функціоналом до самописних утиліт призначених для виконання рутинних завдань. Прикладом таких програм є 3Ds Max, Maya, Blender, Cinema3D, Houdini і т.д. Всі пакети мають свої цілі, а також як і плюси, так і мінуси.

Процес створення тривимірної графіки з допомогою програмних пакетів називається 3D-моделюванням.

Завдання 3D-моделювання – розробити візуальний об'ємний образ бажаного об'єкта. При цьому модель може як відповідати об'єктам з реального світу (автомобілі, будівлі, ураган, астероїд), так і бути повністю абстрактною (проекція чотиривимірного фрактала) [8].

У процесі створення ігор використовується полігональне моделювання.

Полігональне моделювання (polygonal modeling) – це найперший різновид тривимірного моделювання, яка з'явилася в ті часи, коли для визначення точок в тривимірному просторі доводилося вводити вручну з клавіатури координати X, Y і Z. Як відомо, якщо три або більше точок

координат задані як вершини і з'єднані ребрами, то вони формують багатокутник (полігон), який може мати колір і текстуру. З'єднання групи таких полігонів дозволяє змоделювати практично будь-який об'єкт. Недолік полігонального моделювання полягає в тому, що всі об'єкти повинні складатися з крихітних плоских поверхонь, а полігони повинні мати дуже малий розмір, інакше краю об'єкта матимуть огранений вид. Це означає, що якщо для об'єкта на сцені передбачається збільшення, його необхідно моделювати з великою кількістю полігонів (щільністю) навіть, незважаючи на те, що більшість з них будуть зайвими при віддаленні від об'єкта.

У процесі створення тривимірних моделей для комп'ютерних ігор створюється високополігональна модель яка задовольняє вимоги розробки, після чого така модель проходить перевірку і проводиться її ретопологія. Завдяки чому забирається зайва частина полігонів і модель стає нізкополігональних (Low-poly).

Low-poly – (від англ. Low – низько і polygon – полігон) тривимірна модель з малою кількістю полігонів. Тобто це моделі, які складаються з мінімального числа полігонів. При цьому їх достатня кількість для візуального сприйняття одержуваного об'єкта. Широке поширення мають на мобільних платформах в ігровій індустрії в зв'язку з обмеженням продуктивності [9].

1.5 Процес створення тривимірних моделей

У процесі створення тривимірних моделей для комп'ютерних ігор можна виділити наступні етапи:

- моделювання;
- текстурування;
- риггінг (від англ. Rig – оснащення);
- анімація;
- рендеринг (3D візуалізація);
- композитінг.

Моделювання. В ігровій індустрії використовується полігональне моделювання так як полігональні моделі найпростіше візуалізувати (рендерити). Чим більше полігонів у моделі, тим детальніше вона буде виглядати в результаті. В іграх використовуються нізкополігональних моделі так як візуалізація відбувається прямо на ходу (real-time rendering). Також часто в комп'ютерних іграх зустрічається LOD-технологія (Level Of Detail –

від англ. Рівень деталізації). Її суть полягає в тому що моделі спрощуються шляхом підміни їх на більш примітивні, коли гравець віддаляється від них. Завдяки чому система розгружається і вона працює над повною обробкою деталей тільки в межах видимості.

Текстурування. Текстурування – це метод в комп'ютерній графіці для додання більш реалістичного і насиченого виду поверхні об'єктів. В ігровій індустрії цим процесом займаються фахівці, художники по текстурам. Насамперед моделер створює текстурну розгортку (UV-розгортку) – двовимірне зображення поверхні моделі. Така розгортка потрібна для того щоб текстура ідеально лягла на поверхню моделі. Після чого художник малює текстури і прив'язує їх до моделі завдяки UV-розгортці. Створюється цілий набір текстур: колірна текстура, карта нерівностей (bump map), карта нормалей (normal map), карта рельєфу (displacement map), карта відблисків (specular map) та карта прозорості (alpha map). Таким чином виходить готовий візуальний образ моделі.

Рігінг. Рігінг – створення «скелета», кісток моделі. Використовується це тільки в разі якщо змодельований об'єкт матиме анімації. У процесі створення рігінга зазвичай використовують ті ж програмні пакети, що і використовувалися на етапі моделювання. Створюються так звані віртуальні «кістки» моделі, до яких прив'язуються полігони сітки моделі. Після чого на базі «кісток» можуть створюватися анімації, так як при зміні положення цих «кісток», полігональна сітка повністю повторює їх рух.

Анімація. Найпростішим методом анімації тривимірних об'єктів є ключові кадри (Keyframes). Аніматор вказує початкове і кінцеве положення моделей, а проміжні стану обчислюються програмою. Хоч цей процес і простий, він є досить трудомістким так як для створення реалістичних і складних анімацій необхідна велика кількість часу а також великого досвіду і вміння аніматора.

Також існує процедурна анімація, при якій використовуються спеціальні програмні пакети для управління персонажем. Така анімація генерується в режимі реального часу відповідно до встановлених правил, законів і обмежень. Така анімація найбільше підходить для поворотів голови або взаємодії моделі гравця із зовнішнім світом (підбір предметів, підйом по сходах, опускання ніг на нерівні поверхні і дотику руками до навколишніх предметів).

Візуалізація. Візуалізація є завершальним етапом створення тривимірних моделей. У комп'ютерних іграх використовується рендеринг в

реальному часі (real-time rendering). Реакція на дії гравця відбуваються ментально. Світло, колір і тіні формуються за допомогою раніше прорахованих карт і текстур. Щоб якість графіки не страждало при цьому використовуються різні 3D прискорювачі. З цього головним критерієм в грі є швидкість виконання прорахунку. У сучасних іграх рендерингом і компонуванням моделей займається ігровий рушій, наприклад Unity3D, Unreal Engine і інші, але при цьому розробник має повний контроль над цим процесом.

1.6 Постановка задачі

Метою дипломної роботи є створення тривимірних моделей, на базі яких можна реалізувати комп'ютерну гру.

Вимоги до програмного забезпечення:

- програмне забезпечення повинно бути простим в освоєнні і наочним у використанні;
- програмне забезпечення повинно бути безкоштовним на випадок комерційного використання кінцевого продукту;
- програмне забезпечення повинно підтримувати різні формати імпорту і експорту файлів для аналогічних програм і редакторів;
- програмне забезпечення повинно бути легковисним і легко переносимими на аналогічні платформи, відмінні від Windows;
- програмне забезпечення повинно бути продуктивним для забезпечення комфортної роботи на комп'ютері з низькими системними характеристиками;
- програмне забезпечення повинно мати можливість редагування, додавання або видалення текстур моделей;
- програмне забезпечення повинно бути оснащено функціями ригінга і анімації моделей;
- програмне забезпечення повинно мати способи підключення зовнішніх сторонніх модулів для спрощення рутинної роботи;

Вимоги до моделей:

- моделі повинні бути легковажними для можливості рендеринга на комп'ютерах з мінімальними характеристиками;
- моделі повинні представляти собою розташування в межах контурів відповідних векторних карт об'єктів;

- моделі повинні бути створені в нізкополігональних вигляді і покриті реалістичними текстурами або стандартними матеріалами соответствующего реальності;
- модель не повинна містити зайві площині, невидимі при нормальному режимі перегляду;
- всі текстури, що використовуються в моделі, повинні бути формату PNG, JPG або GIF. Всі текстури повинні бути контрастними і різкими. Колір текстур не повинен містити відтінків. На структурах не повинно бути швів, перепадів колірною балансу і різкості між сусідніми текстурами. Довжина і ширина текстур в пікселях повинна бути дорівнює ступеням двійки (16, 32, 64, і т.д.) при цьому достатніми для відображення якісного зображення;
- моделі повинні бути однозначно впізнаваними - максимальний збіг з прототипом;
- моделі повинні поєднуватися між собою і підходити як один одному так і тематиці того, що відбувається на сцені і гармонійно вписуватися в неї.

В результаті дипломної роботи буде створено набір моделей і текстур з необхідними анімаціями для постоення на їх базі повноцінної гри. Планується створення гри в жанрі «пісочниця з виживанням» (survival sandbox), де гравцеві доведеться пережити період засніженій холодної зими. Моделі такої гри повинні відповідати жанром і стилістикою, що відбувається, при цьому повинні сприяти передачі сюжетної лінії і напруженості геймплея в екстремальних ситуаціях гравцеві.

Для цього необхідно здійснити вибір архітектури побудови моделей і програмне забезпечення підходяще вимогам майбутньої реалізації гри.

2 ВИБІР АРХІТЕКТУРИ ТА ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ

2.1 Вибір способу тривимірного моделювання

Існує кілька способів тривимірного моделювання, які використовуються моделлерами в наш час: полігональне, сплайнова і NURBS моделювання. Вони можуть застосовуватися як окремо, так і комплексно.

Полігональне моделювання. Полігональне моделювання – це вид моделювання, який з'явився в ті часи, коли для визначення місцезнаходження точки необхідно було вручну вводити її координати X , Y , Z . Якщо три точки координат задати як вершини і з'єднати ребрами, вийде трикутник, який в тривимірному моделюванні називають полігоном.

Полігон з трьома вершинами називають треангульованим полігоном, з чотирма вершинами - квадрангульованим полігоном. Якщо звернути увагу на створені моделі за допомогою полігоном, можна помітити що в основному вони створюються за допомогою полігонів з чотирма і трьома вершинами. Кожен полігон має свою текстуру і колір, а з'єднавши такі полігони можна отримати тривимірну модель об'єкта. Сполучені між собою полігони утворюють полігональну сітку (Mesh) або полігональний об'єкт (рис. 2.1).

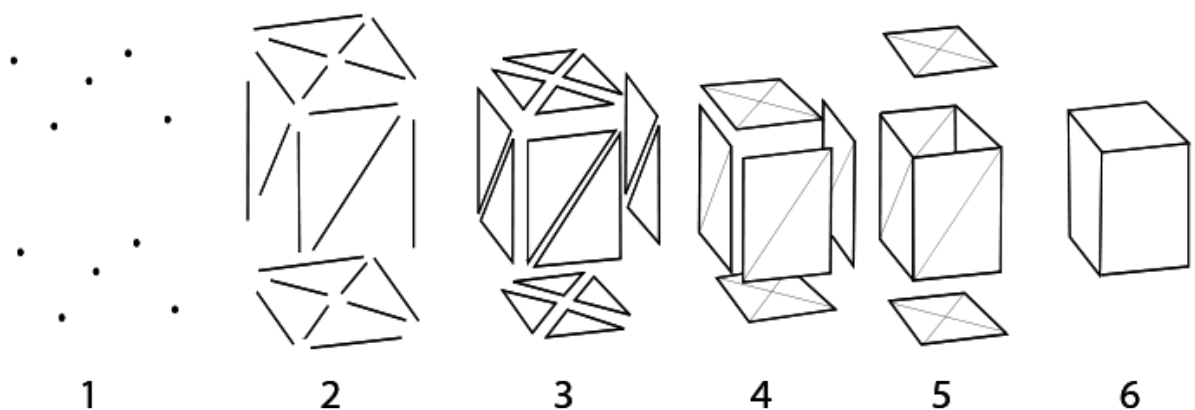


Рисунок 2.1 – Елементи моделювання сітки:

1 – вершини; 2 – ребра; 3 – межі; 4 – полігони; 5 – поверхні; 6 – полігональна сітка;

Об'єкти створені за допомогою полігональних сіток повинні зберігати різні елементи, такі як вершини, ребра, грані, полігони поверхні. У багатьох

випадках зберігаються лише вершини, ребра і або межі, або полігони. Рендерер може підтримувати лише тристоронні межі, так що полігони повинні бути відбудовані з їх безлічі (рис. 2.1). Однак багато рендерерів підтримують полігони з чотирма і більше сторонами, або вміють триангулювати полігони в трикутники на льоту, роблячи необов'язковим зберігання сітки в триангульованій формі.

Полігональна сітка може бути представлена різними способами зберігання вершин, ребер і граней:

- список граней: опис відбувається за допомогою покажчиків в список вершин;
- «крилате» представлення: кожна точка ребра вказується на дві вершини, дві грані і чотири (за годинниковою стрілкою) ребра, які її стосуються;
- полуреберні сітки: спосіб, схожий з «крилатим» представленням, за винятком того, що використовується інформація для обходу лише половини межі;
- чотириреберня сітка: зберігають ребра, полуребра і вершини без якої-небудь вказівки полігонів. Полігон прямо не виражені і можуть бути визначені повним обходом структури;
- таблиця кутів, котрі зберігають вершини в зумовленій таблиці, такий що обхід таблиці неявно задає полігони;
- вершинне уявлення: представлені лише вершини, що вказують на інші вершини. Інформація про гранях і ребрах виражена неявно. Однак простота уявлення дозволяє ефективно проводити над сіткою безліч операцій.

Для того щоб межі об'єкта не мали гранованого виду, необхідно щоб полігони були малого розміру, а поверхню об'єкта складалася з малих площин. При цьому якщо об'єкт буде видно з далекої відстані, досить буде малої кількості полігонів для його відображення.

Моделювання сплайна. Сплавное моделювання – вид моделювання, при якому модель створюється за допомогою сплайнів (рис. 2.2) (від англ. Spline – гнучке лекало, в 3D – тривимірна крива).

Усі сплайни зводяться до каркаса сплайна, на основі якого вже буде створюватися огинаюча тривимірна геометрична поверхня.

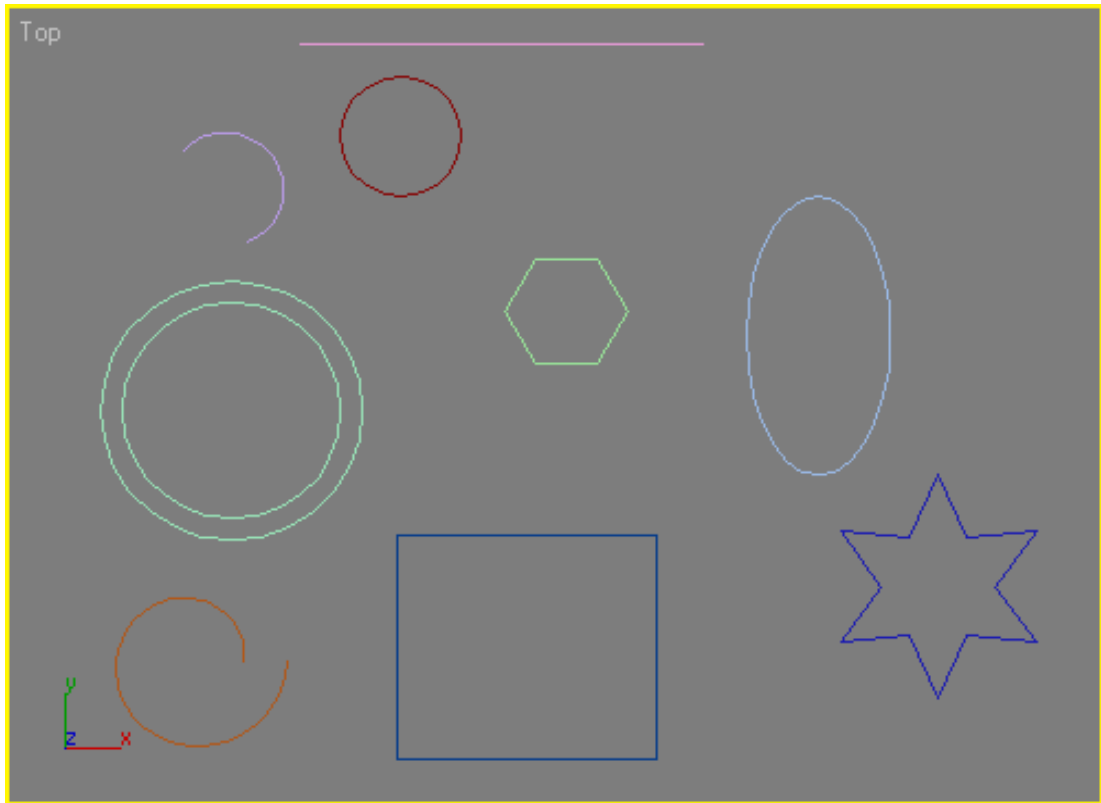


Рисунок 2.2 – Сплайнові форми

За замовчуванням примітиви сплайнів не відображаються на етапі візуалізації (рис. 2.2) і використовуються як допоміжні об'єкти для створення моделей зі складною геометрією. Однак будь-який сплайнові примітив може виступати в сцені як самостійний об'єкт (рис. 2.3).

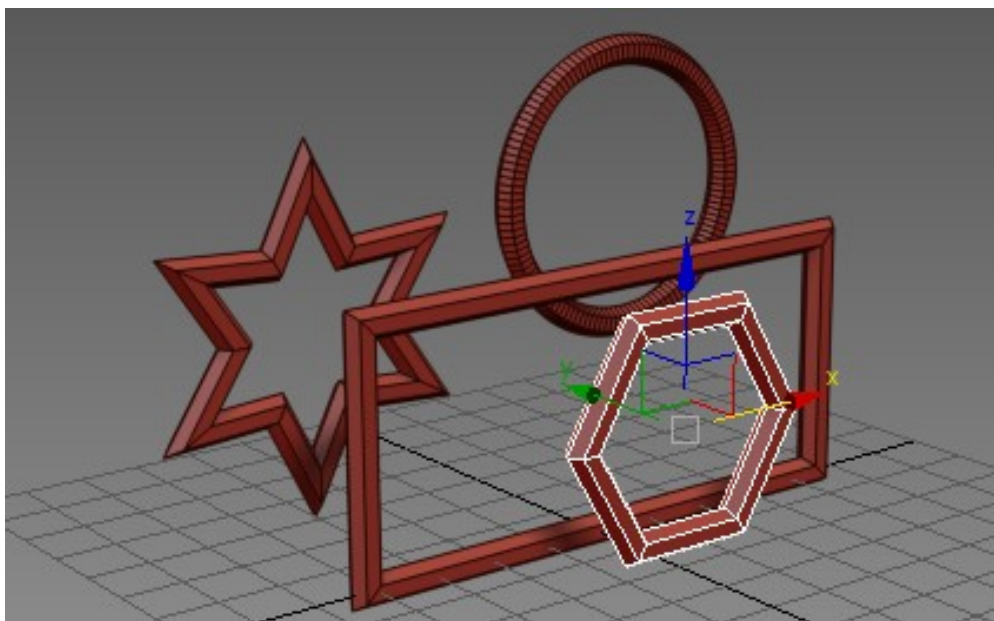


Рисунок 2.3 – Моделі, створені за допомогою 3ds Max

Крім того, в моделювання сплайна використовуються примітиви сплайнів (параметричні об'єкти, що використовуються для моделювання). Базовими сплайновими примітивами є: лінія, дуга, спіраль, окружність, кільце, еліпс, прямокутник, багатокутник, перетин, сплайна текст.

NURBS моделювання. NURBS моделювання або технологія Non-Uniform Rational B-Spline – технологія неоднорідних раціональних B-сплайнів, створення плавних форм і моделей, у яких немає гострих країв, як у полігональних моделей. Саме через цю відмінну рису технологію NURBS застосовують для побудови органічних моделей і об'єктів. NURBS криві, що застосовуються в даному виді моделювання, бувають двох видів: P (Point) криві, і CV (Control Vertex) криві. Point криві управляються вершинами, що знаходяться безпосередньо на самій лінії або об'єкті, а CV управляються точками, що лежать за межами лінії або об'єкта (рис. 2.4).

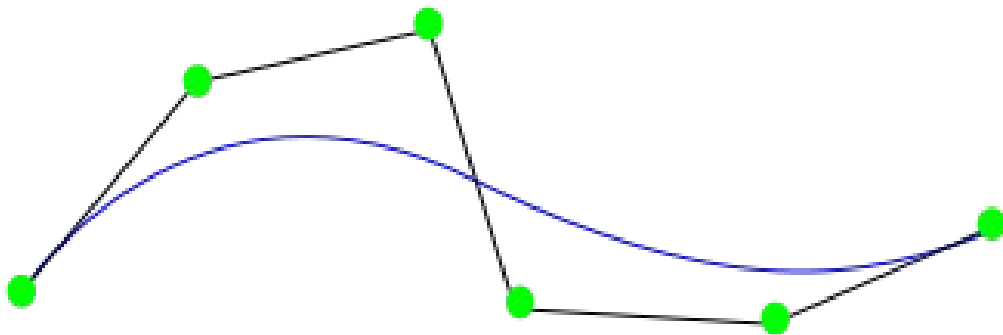


Рисунок 2.4 – NURBS-крива

Так як дипломний проект не вимагає високої деталізованості і має обмеженість в продуктивності комп'ютера для тестування кінцевого результату, ми будемо використовувати полігональне моделювання. Полігональне моделювання дає можливість швидко створювати якісні моделі як простих так і складних форм, при цьому є найпростішим і ефективним в плані продуктивності [10].

2.2 Вибір програмних засобів реалізації моделювання

В арсеналі 3D-моделлера є величезний вибір програмних пакетів для створення тривимірних моделей будь-якої складності. Всі пакети мають як переваги так і недоліки. Розглянемо основні з них, які використовуються в ігровій індустрії [11].

ZBrush. ZBrush – програма для 3D моделювання, створена компанією Pixologic. Відмінною особливістю даного програмного продукту є імітація процесу «ліплення» тривимірної скульптури (скульптинг), посиленого движком тривимірного рендеринга в реальному часі, що істотно спрощує процедуру створення необхідного тривимірного об'єкту. Кожна точка, яка називається піксоль, містить інформацію не тільки про свої координати і значеннях кольору, але і про глибину Z, орієнтацію і матеріал. Це означає що ви не тільки можете «ліпити» тривимірні об'єкти, але і «розфарбувати» його, малюючи штрихами з глибиною. Тобто вам не доведеться малювати тіні і відблиски, щоб вони виглядали натурально – ZBrush зробить все автоматично. Також швидко працює зі стандартними 3D об'єктами, використовуючи пензлі для модифікації геометрії матеріалів і текстур. Дозволяє досягти інтерактивності при великій кількості полігонів. Використовуючи спеціальні методи, можна підняти деталізацію до десятків (а то й сотень) полігонів. Також є безліч модулів. Всі функції ZBrush знаходяться всередині палітр. Кожна палітра містить групу пов'язаних функцій. В межах палітри ці функції далі розділені на групи, щоб полегшити доступ до певних налаштувань, які необхідні моделлеру.

Переваги пакета:

- величезний арсенал функцій і додаткових плагінів;
- зручність в роботі за допомогою сторонніх пристроїв введення;
- постійний розвиток;
- велика кількість навчальних матеріалів;
- використання підходу «скульптингу» в 3D моделюванні.

Недоліки:

- складний в освоєнні навіть досвідченим моделлерам;
- дорога вартість продукту і відсутність безкоштовної ознайомлювальної версії.

Blender. Blender – вільний, багатоплатформовий пакет для створення тривимірної комп'ютерної графіки, включаючи в себе засоби моделювання, анімації, текстурування, рендерингу, постобробки і монтажу відео зі звуком. В даний час користуються найбільшою популярністю серед безкоштовних 3D редакторів, в зв'язку з його швидким і стабільним розвитком, якому сприяє професійна команда розробників. Характерною особливістю цього пакету є кросс-платформовість, так як він здатний стабільно працювати як на Windows, так і на Linux. А також його малий розмір, в порівнянні з іншими пакетами і продуктивність, завдяки чому його можна запустити навіть на

старих моделях нетбуків. У базову поставку не входить розгорнута документація і болше кількість демонстраційних сцен, що є мінусом пакета.

Функції:

- підтримка різноманітних геометричних примітивів (сфера, куб, рівнина);
- універсальні вбудовані механізми рендеринга;
- інструменти анімації;
- мова програмування Python використовується як засіб створення інструментів і прототипів;
- базові функції нелінійного редагування і комбінування відео;
- Blender Game Engine – підпроект команди Blender, що представляє інтерактивні функції, такі як визначення колізій, рушій динаміки і програмована логіка;
- у Blender Об'єкт (який являє собою сутність, що взаємодіє з навколишнім світом) і його Дані (певна форма/функції об'єкта) розділені. Відношення Об'єкт-Дані представляється відношенням $m:n$ (термін, що відноситься до теорії баз даних, позначає можливість декількох об'єктів використовувати одні і ті ж дані) і динамічно зв'язані між собою, дозволяючи використовувати деякі процеси швидкого моделювання, унікальні для Blender.
- внутрішня файлова система, що дозволяє зберігати декілька сцен в єдиному файлі (так званому .blend файлі);
- усі «.blend» файли сумісні як із старішими, так і з новішими версіями Blender. Так само, всі вони переносимі з однієї платформи на іншу. І можуть використовуватися як засіб перенесення створеного раніше контенту;
- Blender робить резервні копії проектів під час всієї роботи програми, що дозволяє зберегти дані при непередбачених обставинах;
- усі сцени, об'єкти, матеріали, текстури (тільки власні, не імпортовані), звуки, зображення post-production ефекти можуть бути збережені у єдиний «.blend» файл;
- налаштування робочого середовища можуть бути збережені у «.blend» файл, завдяки чому, при завантаженні файлу Ви отримаєте саме те, що зберегли в нього. Файл можна зберегти як «користувацький за умовчанням», і щоразу при запуску Blender Ви

отримуватимете необхідний набір об'єктів та підготовлений до роботи інтерфейс.

Проте, внутрішній зміст «.blend» файлу менше схожий на структурований опис об'єктів та їхніх взаємин, і ближче до прямого дампу області пам'яті програми. Це робить практично неможливим перетворення «.blend» файлів в інші формати. При цьому слід відмітити вельми просунутий механізм експорту в різноманітні формати, такі як obj, dxf, stl, 3ds та інші (список поступово росте).

Blender мав репутацію програми складної для вивчення. Практично кожна функція має відповідне їй поєднання клавіш, і враховуючи кількість наданих можливостей у Blender, кожна клавіша включена в більш ніж одне поєднання (shortcut). З того часу як Blender став проектом з відкритим вихідним кодом, було додано повні контекстні меню до усіх функцій, а використання інструментів зроблене логічнішим та гнучкішим. Додамо сюди подальше поліпшення користувацького інтерфейсу з введенням колірних схем, прозорих плаваючих елементів, новою системою проглядання дерева об'єктів та різними дрібними змінами.

Користувацький інтерфейс Blender'a слідує наступним виразним концепціям:

- режими редагування. Два основні режими Об'єктний режим (Object mode) та Режим редагування (Edit mode), які перемикаються клавішею Tab. Об'єктний режим в основному використовується для маніпуляцій з індивідуальними об'єктами, тоді як режим редагування – для маніпуляцій з фактичними даними об'єкта. Наприклад, для полігональної моделі в об'єктному режимі ми можемо переміщати, змінювати розмір та обертати модель цілком, а режим редагування використовується для маніпуляції окремих вершин конкретної моделі. Також є деякі інші режими, такі як Vertex Paint та UV Face select;
- широке використання гарячих клавіш. Більшість команд виконуються з клавіатури. До появи 2.x і особливо 2.3x версії, це був єдиний шлях виконувати команди, і це було найбільшою причиною створення репутації Blender'у як складної для вивчення програми. Нова версія має повне графічне меню;
- управління робочим простором. Графічний інтерфейс Blender'a складається з одного або декількох екранів, кожен з яких може бути роздільний на секції та підсекції, які можуть бути будь-якою

частиною інтерфейсу Blender'a. Графічні елементи кожної секції можуть контролюватися тими ж інструментами, що і для маніпуляції в 3d просторі, для прикладу можна зменшувати та збільшувати кнопки інструментів тим же шляхом, що і у 3d перегляді. Користувач повністю контролює розташування і організацію графічного інтерфейсу, це робить можливою налаштування інтерфейсу під конкретні завдання, такі як редагування відео, UV mapping та текстуровання, і захвати елементи інтерфейсу які не потрібні для даного завдання. Цей стиль графічного інтерфейсу дуже схожий на стиль, використовуваний у редакторі UnrealEd мап для гри Unreal Tournament.

Для рендерингу Blender використовує рушій трасування променів Cycles Render.

Переваги пакета:

- безкоштовність;
- відкритий код;
- постійний розвиток;
- кросс-платформовість.

Недоліки:

- відсутність документації в базовій поставці (її можна знайти на сайті програми або з інших джерел).

Autodesk Mudbox. Autodesk Mudbox – професійна графічна програма, призначена для моделювання високополігональних цифрових скульптур і текстурного фарбування 3D моделей. Надає фахівцям з моделювання і художникам по текстурам всі можливості для створення цифрових 3D об'єктів і 2D скетчів, як якщо б вони працювали з глиною і фарбами. Так само Mudbox використовується для створення карт нерівностей, карт нормалей, карт заміщення та ін. На відміну від інших програм подібного роду, Mudbox володіє зручним інтуїтивним інтерфейсом, що дозволяє повністю освоїти його за кілька днів.

Autodesk Mudbox це передове програмне забезпечення, в арсеналі якого 3D пензлі високого дозволу, що дозволяють моделювати скульптури, які складаються з десятків мільйонів полігонів. Ця програма була розроблена для задоволення професійних потреб художників і 3D модельєрів, які працюють у кіноіндустрії, розробників комп'ютерних ігор, дизайнерів і промислових проектувальників. З самого початку Mudbox мав великий вплив у сфері кіноіндустрії, дизайну та промисловості. Його інноваційні функції

включають в себе: цифрова скульптура – легкі в поводженні скульптурні пензлі дозволяють художникам з великою точністю моделювати 3D геометрію будь-якої складності. 3D шари: 3D еквівалент багат шарових 2D зображень в таких графічних редакторах, як Adobe Photoshop, Corel Draw та ін, що дозволяє одночасно працювати з декількома шарами різної деталізації. Це дає можливість зберігати і повертатися до деталізації будь-якого рівня, комбінувати, змішувати, видозмінювати і маскувати шари. Дружній користувальницький інтерфейс і зручне робоче простір значно скорочують час, необхідний для освоєння програми і однаково підходять як для дизайнерів, так і для митців у галузі мультимедіа. Продуктивність: інноваційні технології дозволяють 3D художнику працювати швидко і ефективно навіть з самими складними за своєю будовою об'єктами складаються з мільйонів полігонів. Мульти-дозвіл: технологія ієрархічного поділу, що дозволяє художникам отримувати високополігональні моделі з великою кількістю полігонів. Запікання текстур: перенесення елементів зовнішнього вигляду об'єкта (дифузний колір, пряме освітлення, GI тощо) в текстури і їх автоматичний маппінг на поверхню об'єкта. Має широке поширення в ігровій індустрії, дозволяючи запікати не тільки дифузний колір об'єкту, але і карти зсуву (добавляючі рельєфність об'єкту). Mudbox дозволяє експортувати текстури високого дозволу (до 8000 пікселів) і зберігати 8, 16, і 32-бітові карти. Селективна симетрія надає можливість зберігати індивідуальну симетрію для кожного шару. Маскування дає можливість редагувати шари не порушуючи геометрії замаскованого шару. Діє подібно ластіку з можливістю відновлення віддалених ділянок шару. Менеджер шарів дозволяє дублювати шари, переміщати шаром вище або нижче, групувати і об'єднувати в один шар всі виділені, змінювати силу впливу шару тощо. 3D пензлі: Mudbox своєму розпорядженні великий набір скульптурних пензлів за допомогою яких можливий моделювання гладких опуклих і вдавлених поверхонь, зморшок, плоских поверхонь, згладжування, загострення тощо. За допомогою кривих Falloff можливо міняти гостроту пензля, роблячи її більш круглої або загостреною. Пензлі Stamps дозволяють залишати опуклі або вдавнені відбитки будь-якій поверхні, імітуючи шкіру рептилій, кам'янисту поверхню, штукатурку і багато іншого. За допомогою пензлів Stencils можна з легкістю моделювати нерівності з будь-якого завантаженого зображення, інтерактивно змінювати його масштаб, обертати і переміщати по робочій області. Криві лінії: діючи як напрямні, криві лінії надають повний контроль художника над створюваним мазком.

Локальна інтеграція полігонів дозволяє збільшувати або зменшувати кількість полігонів у виділеній області, з можливістю переходити з одного рівня на інший. Режим згладжування нормалей дозволяє художнику працювати як з грубою сіткою (в цьому режимі чітко видно грані), так і з згладженої, що приховує всі нерівності. Режим Draft візуалізації (Draft Render) різко підвищує продуктивність інтерактивної навігації для сцен з високою щільністю сітки. Режим швидкого огляду завдяки оперативній роботі рушія тривимірної візуалізації, забезпечує високоякісне відображення 3D об'єктів в реальному часі. Підтримка трикутних і багатокутних граней. На відміну від інших програм, Mudbox дозволяє імпортувати і модифікувати 3D об'єкти, які з трьох-, чотирьох- і багатокутних полігонів, з можливістю записування текстур, збереженням структури сітки і ID вершин. Камери Full 3D: Mudbox надає можливість художникам користуватися кількома 3D-камерами в одній сцені, плюс він підтримує імпорт і експорт існуючих камер між сценами. Камери можуть бути заблоковані або перетворені за допомогою введення з клавіатури. За допомогою 3D-камер можливе переміщення не тільки навколо об'єкта, але і всередині нього. Такі камери дозволяють виробляти будь-які дії над об'єктом, незалежно від напрямку відстані. Підтримка мультисцен: 3D художники мають можливість створювати або імпортувати мультисцени, що складаються з персонажів, камер, джерел світла, текстур і матеріалів. Управляти кожним з них окремо, виділяючи, приховуючи, блокуючи, множачи кількість полігонів тощо. Вбудований графічний браузер (Image Browser) з повною підтримкою 16 і 32-бітових зображень – дозволяє користувачам переглядати малюнки, використовувати їх як stamps, stencils або як зображення для 3D камер.

Переваги пакета:

- висока продуктивність;
- зручний інтуїтивний інтерфейс;
- простий в освоєнні;
- широкий функціонал для створення як моделей так і карт нормалей для них.

Недоліки:

- дорогий продукт, не має безкоштовної ознайомлювальної версії.

Autodesk 3ds Max. Autodesk 3ds Max (раніше 3D Studio Max) – повнофункціональна професійна система для створення і редагування тривимірної графіки. Містить в собі найсучасніші засоби і алгоритми для

художників і фахівців в області тривимірного моделювання. Працює тільки на операційних системах Windows і Windows NT.

Autodesk 3ds Max доступний в двох ліцензійних версіях: студентська – безкоштовна (вимагає реєстрації), яка надає повну версію пакету, однак її не можна використовувати в комерційних цілях, і повна (комерційна) версія.

3ds Max володіє величезним функціоналом для створення різноманітних за формою і складності тривимірних моделей, реальних чи фантастичних об'єктів навколишнього світу з використанням технік і механізмів, що включають:

- полігональне моделювання, в яке входять Editable mesh (редагована сітка) і Editable poly (редагований полігон);
- моделювання на основі неоднорідних раціональних B-сплайнів (NURBS);
- моделювання на основі порцій поверхонь Безьє (Editable patch) – підходить для моделювання тіл обертання;
- моделювання з використанням вбудованих бібліотек стандартних параметричних об'єктів (примітивів) і модифікаторів.

Методи моделювання можуть поєднуватися один з одним. Моделювання на основі стандартних об'єктів, як правило, є основним методом моделювання і є початковою точкою для створення об'єктів складної структури, що пов'язано з використанням примітивів у поєднанні один з одним як елементарних частин складових об'єктів. Стандартний об'єкт «Чайник» входить до цього набору в силу історичних причин: він використовується для тестів матеріалів та освітлення в сцені, і, крім того, давно став своєрідним символом тривимірної графіки. Візуалізація (або вимальовування) є заключним етапом роботи над модельованою сценою. Тільки після візуалізації можна побачити усі властивості матеріалів об'єктів, ефекти зовнішнього середовища, які застосовані в складі сцени. Для виведення кінцевого зображення на екран вибирають необхідний модуль візуалізації (MB). Більшість MB є окремими програмами вбудованими як доповнення в 3ds Max.

Переваги пакета:

- величезний функціонал;
- постійний розвиток;
- велика кількість плагінів;
- документованість;
- простий в ознайомленні завдяки розвиненій спільноті.

Недоліки:

- дорога вартість для комерційного використання;
- потребує великий об'єм пам'яті для пакету з усіма його функціями.

Autodesk Maya. Autodesk Maya – інструментальна система тривимірної графіки та комп'ютерної анімації, що володіє потужними інструментами для створення і редагування тривимірних моделей. Спочатку розроблена для ОС Irix, але незабаром була перенесена на Linux, Windows і Mac OS X. В Maya вбудований потужна інтерпретуєма платформеннонезалежна мова MEL (Maya Embedded Language), дуже схожий на Tcl і C. Мова є потужним інструментом для доопрацювання самої Maya, так як внутрішній інструментал пакета був написаний на MEL, зокрема користувач може написати свої дії як скрипт MEL, з якого незабаром можна зробити зручний макрос. Для написання зовнішніх розширень на C ++ існує зручний документований C++ API, а також є можливість написання доповнень на Python.

Переваги пакета:

- широкий функціонал як для моделлера так і для аніматора або розробника;
- постійний розвиток;
- кросплатформеність;
- внутрішня скриптова мова MEL.

Недоліки:

- дорогий продукт для комерційного використання, не має безкоштовних ліцензій;
- високий поріг входження при навчанні.

Houdini. Houdini – професійний програмний пакет для роботи з тривимірною графікою, розроблений компанією Side Effects Software (Торонто, Канада). На сьогодні є стандартом у CG-індустрії. Головна перевага даного пакета в тому, що він є середовищем візуального програмування для моделювання:

- безліч геометричних примітивів (куб, сфера, т.і.);
- полігональне моделювання;
- воксельне моделювання;
- сплайні (включаючи B-сплайні (NURBS), Криві Без'є);
- метасфера (англ. Metaball);
- система Ліндермаєра (англ. L-system);

- ноди для зміни властивостей об'єктів об'єднаних в групу SOPs (surface operators);
- групування геометрії;
- частинки – ноди для роботи з частинками об'єднаних в групу – POPs (particle operators).

Для анімації:

- заснована на ключових кадрах;
- можливість процедурного управління параметрами каналів сцени за допомогою HScript або за допомогою операторів групи CHOPs (channel operators);
- редактор ключових кадрів – з можливістю графічного представлення параметрів сцени, зміни типу інтерполяції значення та розташування в часі ключового кадру;
- анімація персонажів – інструменти для персонажної анімації: скелети, клунь, волосся і шерсть (з прорахунком фізичної взаємодії), Crowd control т.і.

Для фізичного моделювання (розрахунок фізичної взаємодії):

- твердих (Rigid Body) і м'яких тіл (Soft Body);
- «канатних» тіл (Wire);
- тканин (Cloth);
- волосся і вовни (Fur);
- розрахунок фізичної поведінки і візуалізації газів і рідин (моделювання поведінки вогню, диму, більшості атмосферних явищ, розчинення однієї рідини в іншій, карт намочання (Wet Map) т.і.

Для освітлення:

- великий набір джерел світла;
- шейдери освітлення (Light Shader);
- прорахунок більшості фізичних властивостей світла: тіней, глобального освітлення, акустики тощо.

Для матеріалів і шейдерів:

- бібліотека матеріалів;
- можливість створення користувальницького матеріалу шляхом редагування існуючого або програмування шейдера матеріалу – SHOPs (shading operator).

Для об'ємного звуку:

- Houdini володіє набором інструментів, які дозволяють працювати з об'ємним звуком. Розташувавши віртуальні джерела звуку в сцені, за допомогою спеціального об'єкта виконує роль мікрофона (Microphone object), можна отримати звуковий сигнал в конкретній точці сцени, при цьому, враховуючи акустичні властивості об'єктів сцени.

Для рендерингу:

- зручна нодова структура рендерингу (ROPs – render operators) з можливістю черги прорахунку, ієрархічного взаємодії нод прорахунку між собою. Ноди рендеринга «пам'ятають» свої параметри, що дає можливість швидкого перерахунку потрібних елементів сцени;
- підтримка різних засобів візуалізації: Mantra, Renderman, mental ray, та інших;
- підтримка основних графічних форматів для експорту сцени з підтримкою багатопланових зображень;
- мережевий рендеринг;
- рендеринг для окремо взятих джерел світла або об'єктів належать до певної групи геометрії.

Для композитингу:

- вбудований інструмент для композитингу (COPs – composite operators) з повноцінною підтримкою пікселів з глибиною кольору 32 біта з плаваючою точкою;
- інформація з COPs може бути безпосередньо використана в сцені проекту.

Всі операції в Houdini доступні для виконання і за допомогою програмного коду. Оператор, при цьому, представляється у вигляді класу, а параметри оператора у вигляді членів цього класу. Кожен оператор має свою адресу в проекті схожий за структурою з Unix-подібними операційними системами. Змінні і параметри функцій оператора можуть обробляти велику кількість типів даних:

- геометричні дані – точки, примітиви, нормалі, проекції, криві і т.д.;
- графічні дані – растровий масив даних із графічного файлу або безпосередньо з вбудованого композеру COPs;
- частинки – структури даних для POPs;
- фізичні дані – DOPs;
- рядкові дані – текстові параметри операторів;

- числові дані – параметри операторів, атрибути, вектори, більшість даних для програмування шейдер т.і.;
- інформація з пристроїв введення – миші, клавіатури, графічного планшета, MIDI-порту, мікрофона;
- RAW дані з файлу;
- дані з TCP/IP-з'єднання;
- аудіо-потік з файлу або віртуального мікрофона розташованого в просторі сцени.

Переваги пакета:

- величезний функціонал;
- кросс-платформовість;
- постійний розвиток;
- документованість;
- можливість програмування дій за допомогою програмного коду.

Недоліки:

- дорога вартість для комерційного використання;
- велика вага пакета з усіма його функціями.

Виходячи з розглянутих пакетів і проаналізувавши їх переваги та недоліки був обраний програмний пакет Blender, так як він єдиний являє собою безкоштовним в комерційному використанні, а також являє собою кроссплатформним і простим в освоєнні і повністю задовольняє вимогам дипломного проекту в плані функціональності.

2.3 Методи рендерингу зображення

Важливим етапом створення тривимірних моделей є ренеринг вихідного зображення на екран користувача, завдяки чому наприклад гравець може побачити тривимірну модель ігрового світу.

Візуалізація (візуалізація) – термін в комп'ютерній графіці, що позначає процес отримання зображення по моделі за допомогою комп'ютерних програм.

Моделлю є опис будь-яких об'єктів або явищ на певній мові або у вигляді структури даних. Такий опис може містити геометричні дані, положення точки спостереження, інформації про висвітлення, ступеня наявності певної речовини, напруженість фізичного поля і т.п.

Часто в комп'ютерній графіці під рендерингом розуміють створення плоскої картини за розробленою 3D сценою, синонімом чогось являється «візуалізація».

На даний момент розроблено безліч алгоритмів візуалізації. Існує програмне забезпечення яке може використовувати різні алгоритми для створення кінцевого зображення.

Трасування кожного променя світла на сцені непрактично і займає неприпустимо довгий час.

Навіть трасування малої кількості променів, достатнього, щоб отримати зображення, займає надто багато часу, якщо не застосовується апроксимація.

Внаслідок цього було розроблено чотири групи методів, більш ефективних, ніж моделювання всіх променів світла, що освячують сцену:

- рестерізація спільно з методом сканування рядків. Візуалізація проводиться проектуванням об'єктів сцени на екран без розгляду ефекту перспективи щодо спостерігача;
- ray casting (рейкастинг). Сцена розглядається, як спостережувана з певної точки. З точки спостереження на об'єкти сцени прямують промені, за допомогою яких визначається колір пікселя на двовимірному екрані. При цьому промені припиняють своє поширення, коли досягають будь-якого об'єкта сцени або її фону. Можливо використання будь-яких дуже простих способів додавання оптичних ефектів. Ефект перспективи виходить природним чином в разі, коли кидаючі промені запускаються під кутом, що залежать від положення пікселя на екрані і максимального кута огляду камери;
- трасування променів схожа на метод кидання променів. З точки спостереження на об'єкти сцени прямують промені, за допомогою яких визначається колір пікселя на двовимірному екрані. Але при цьому промінь не припиняє своє поширення, а розділяється на три промені-компонента, кожен з яких вносить свій внесок в колір пікселя на двовимірному екрані: відбитий, тінювий і заломлений. Кількість таких компонентів визначає глибину трасування і впливає на якість і фотореалізм зображення (рис. 2.5). Завдяки своїм концептуальним особливостям, метод дозволяє отримати дуже фотореалістичні зображення, проте через велику ресурсоемності процес візуалізації займає чимало часу.

- Трасування шляху (path tracing) використовує схожий принцип трасування поширення променів, однак метод являється найбільш наближеним до фізичних законів поширення світла. Також являється ресурсомістким.

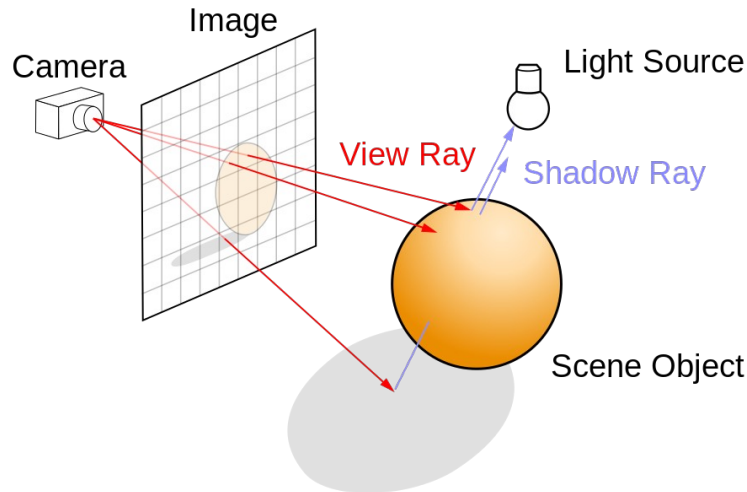


Рисунок 2.5 – Трасування променів

Передове програмне забезпечення зазвичай поєднує в собі кілька технік, щоб отримати достатньо якісне і фотореалістичне зображення за прийнятні витрати обчислювальних ресурсів.

Реалізація механізму рендеринга завжди ґрунтується на фізичній моделі. Винонуванні обчислення відносяться до тієї чи іншої фізичної або абстрактної моделі. Основні ідеї прості для розуміння, але складні для застосування. Як правило, кінцеве рішення або алгоритм більш складні і містять в собі комбінацію різних технік.

Ключем до теоритичного обґрунтування моделей рендеринга служить рівняння рендеринга. Воно є найбільш повним формальним описом частини рендеринга, що не відноситься до сприйняття кінцевого зображення. Всі моделі являють собою якісь наближені рішення цього рівняння:

$$L_o(x, \vec{w}) = L_e(x, \vec{w}) + \int_{\Omega} f_r(x, \vec{w}, \vec{w}') L_i(x, \vec{w}') (\vec{w} \cdot \vec{n}) d\vec{w}'$$

Кількість світлового випромінювання (L_o), що виходить з певної точки в певному напрямку є власне випромінювання і відбите випромінювання. Відбите випромінювання є добуток суми по всіх напрямках випромінювання,

що надходить (L_i), та коефіцієнту відбиття з даного кута. Об'єднуючи в одному рівнянні світло, яке надходить, з тим, що випромінюється, в одній точці, це рівняння формує опис усього світлового потоку в заданій системі [12].

2.4 Вибір ігрового рушія для тестування тривимірних моделей

Завершаючим етапом у створенні тривимірних моделей є тестування на коректність відображення обраним ігровим рушієм. У наш час з'вилось велика кількість різних професійних програмних пакетів для створення комп'ютерних ігор, завдяки яким було створено як тисячі ігор від професійних компаній, до мільйонів невеликих ігор від малих команд розробників.

Гральний рушій (англ. Game engine) – програмний рушій, центральна програмна частина будь-якої відеогри, яка відповідає за всю її технічну сторону, дозволяє полегшити розробку гри за рахунок уніфікації і систематизації її внутрішньої структури. Важливим значенням рушія є можливість створення багатоплатформових ігор (сьогодні найчастіше одночасно для ПК, PS4 та Xbox One).

Основну функціональність гри зазвичай забезпечує рушій гри, що включає рушій рендеринга («візуалізатор»), фізичний рушій, звук, систему скриптів, анімацію, ігровий штучний інтелект, мережевий код, керування пам'яттю, багатонитевість і граф сцени. Часто на процесі розробки можна заощадити за рахунок повторного використання одного рушія гри для створення декількох різних ігор.

Unity3D. Unity – багатоплатформовий інструмент для розробки дво- та тривимірних додатків та ігор, що працює на операційних системах Windows і OS X. Створені за допомогою Unity застосування працюють під системами Windows, OS X, Android, Apple iOS, Linux, а також на гральних консолях Wii, PlayStation 3 та Xbox 360. Також є можливість створювання інтернет-додатків за допомогою спеціального під'єднуваного модуля для браузера Unity а також за допомогою експериментальної реалізації в межах модуля Adobe Flash Player. Застосування створені за допомогою Unity підтримують DirectX та OpenGL.

Редактор Unity має простий Drag & Drop інтерфейс, який легко налаштовувати, що складається з різних вікон, завдяки чому можна проводити налагодження гри прямо в редакторі. Рушій підтримує три сценарних мови: C #, JavaScript (модифікація). Проект в Unity ділиться на

сцени (рівні) – окремі файли, що містять свої ігрові світи зі своїм набором об'єктів, сценаріїв, і налаштувань. Сцени можуть містити в собі як, об'єкти (моделі), так і порожні ігрові об'єкти – тобто ті які не мають моделі. Об'єкти, в свою чергу містять набори компонентів, з якими і взаємодіють скрипти. Також у них є назва (в Unity допускається наявність двох і більше об'єктів з однаковими назвами), може бути тег (мітка) і шар, на якому він повинен відображатися.

Так, у будь-якого предмета на сцені обов'язково присутній компонент Transform – він зберігає в собі координати місця розташування, повороту і розмірів по всіх трьох осях. У об'єктів з видимою геометрією також за замовчуванням присутній компонент Mesh Renderer, що робить модель видимою.

Також Unity підтримує фізику твердих тіл і тканини, фізику типу Ragdoll (ганчіркова лялька). У редакторі є система успадкування об'єктів; дочірні об'єкти будуть повторювати всі зміни позиції, повороту і масштабу батьківського об'єкта. Скрипти в редакторі прикріплюються до об'єктів у вигляді окремих компонентів.

При імпорті текстури в рушій можна згенерувати alpha-канал, тір-рівні, normal-map, light-map, карту відображень, проте безпосередньо на модель текстуру прикріпити не можна – буде створено матеріал, з яким буде призначений шейдер, і потім матеріал прикріпиться до моделі.

Редактор Unity підтримує написання і редагування шейдерів. Крім того він містить компонент для створення анімації, яку також можна створити попередньо в 3D-редакторі та імпортувати разом з моделлю, а потім розбити на файли.

- технічна характеристика;
- сценарії на C#, JavaScript та Boo;
- ігровий рушій повністю пов'язаний із середовищем розробки. Це дозволяє випробовувати гру прямо в редакторі;
- робота з ресурсами можлива через звичайний Drag&Drop;
- система успадкування об'єктів;
- підтримка імпортування великої кількості форматів файлів;
- вбудований генератор ландшафтів;
- вбудована підтримка мережі;
- існує рішення для спільної розробки – Asset Server. Також можна використовувати зручний для користувача спосіб контролю версій. Наприклад, SVN або Source Gear.

Графічний рушій використовує DirectX (Windows), OpenGL (Mac, Windows, Linux), OpenGL ES (Android, iOS), та спеціальне власне API для Wii.

Також підтримуються bump mapping, reflection mapping, parallax mapping, screen space ambient occlusion (SSAO), динамічні тіні з використанням shadow maps, render-to-texture та повноекранні ефекти post-processing.

Unity підтримує файли 3ds Max, Maya, Softimage, Blender, modo, ZBrush, Cinema 4D, Cheetah3D, Adobe Photoshop, Adobe Fireworks та Allegorithmic Substance.

В ігровий проект Unity можна імпортувати об'єкти цих програм та робити налаштування за допомогою графічного інтерфейсу.

Для написання шейдерів використовується ShaderLab, що підтримує шейдерні програми написані на GLSL або Cg. Шейдер може включати декілька варіантів реалізації, що дозволяє Unity визначати найкращий варіант для конкретної відеокarti. Unity також має вбудовану підтримку фізичного рушія Nvidia PhysX (колишнього Ageia), підтримку симуляції одягу в системі реального часу на довільній та прив'язаній полігональній сітці (починаючи з Unity 3.0), підтримку системи ray casts та шарів зіткнення.

Переваги пакета:

- умовно-безкоштовний для комерційного використання;
- відкритий код;
- низький поріг входження;
- підтримка усіх сучасних форматів експорту тривимірних моделей;
- постійний розвиток;
- кросс-платформовість.

Недоліки:

- низька якість оптимізації вихідного продукту для комп'ютерів з низькими системними потребуваннями.

CryEngine. CryEngine 3 – ігровий рушій, розроблений німецькою компанією Crytek і є наступником рушія CryEngine 2. Він із самого початку розроблявся як багатоплатформовий і дозволяє створювати ігри для наступних платформ: ігрових консолей Microsoft Xbox 360, PlayStation 3 і IBM PC-сумісних комп'ютерів, а також наступних їхніх версій. CryEngine 3 написано мовами C++ і Lua. CryEngine 3 є кросплатформним рушієм, він підтримує IBM PC-сумісні комп'ютери та ігрові консолі Microsoft Xbox 360 і Sony PlayStation 3.

Шейдери в CryEngine 3 пишуться одноразово на мові програмування високого рівня, а потім автоматично компілюються під кожну платформу.

Вихід шейдерів оптимізується через налаштування художника і під ту тривимірну середу, в якій даний шейдер буде використовуватися. Завдяки цьому стає можливим створення таких ефектів, як: «ефект невидимості», мокрі, брудні і заморожені поверхні, які можуть бути «нашаровані» одна на іншу і об'єднані з іншими шейдерами, що симулюють такі ефекти, як метал, скло та інші. CryEngine 3 підтримує попіксельне освітлення реального часу, відображення, заломлення, ефекти об'ємного спека і анімовані текстури для симуляції вікон, кульових отворів, поверхонь з сонячними відблисками і багато інших ефектів. Шейдери CryEngine 3 використовують уніфіковану шейдерну архітектуру, яка стала доступна починаючи з Direct3D 10.

«Übershader» являє собою одну шейдерну програму з багатьма особливостями: від одного до чотирьох джерел світла, типи джерел світла, кубічні карти віддзеркалень, туман, деталізовані текстури, карти нормалей, дзеркальні текстури та ін. Можуть бути згенеровані мільйони комбінацій Übershader. У Übershader використовується динамічне розгалуження, поділ на безліч проходів, зменшення комбінацій і прийняття варіантів з меншою функціональністю і меншою необхідною продуктивністю. Використовується асинхронна компіляція шейдерів і розподілена система завдань (англ. Distributed Job System) для компіляції шейдерного кеша.

Переваги пакета:

- має безкоштовну версію для ознайомлення;
- підтримка усіх сучасних форматів експорту тривимірних моделей;
- висока якість рендерингу моделей та ефектів;
- іноваційна технологія створення шейдерів.

Недоліки:

- відсутність безкоштовної версії для комерційного використання;
- високі вимоги до характеристик комп'ютера розробника;
- закритий програмний код.

Unreal Engine. Unreal Engine – ігровий рушій, розроблюваний і підтримуваний компанією Epic Games. Написаний мовою C++, рушій дозволяє створювати ігри для більшості операційних систем і платформ: Microsoft Windows, Linux, Mac OS і Mac OS X, консолей Xbox, Xbox 360, PlayStation 2, PlayStation Portable, PlayStation 3, Wii, Dreamcast і Nintendo GameCube. Для спрощення портування рушій використовує модульну систему залежних компонентів: підтримує різні системи рендерингу (Direct3D, OpenGL, Pixomatic; раніше підтримувалися Glide API, S3 Metal, PowerVR SGL), відтворення звуку (EAX, OpenAL, DirectSound3D; раніше

підтримувалися A3D), засоби голосового відтворення тексту, розпізнавання мовлення (тільки для Xbox360, PlayStation 3, Nintendo Wii і Microsoft Windows, також планувалося для Linux і Mac), модулі для роботи з мережею й підтримка різних пристроїв вводу. Для гри у мережі підтримуються технології Windows Live, Xbox Live, і GameSpy, включаючи до 64 гравців (клієнтів) одночасно. Попри те, що офіційно засоби розробки не містять у собі підтримки великої кількості клієнтів на одному сервері, рушій використовувався для створення MMORPG-ігор. Один з найвідоміших представників жанру, Lineage II, використовує рушій Unreal Engine.

Усі елементи ігрового рушія представлені у вигляді об'єктів, що мають набір характеристик, і клас, який визначає доступні характеристики. У свою чергу будь-який клас є «дочірнім» класом об'єкт. Серед основних класів і об'єктів можна виділити наступні:

Актор (actor) – базовий клас, що містить усі об'єкти, які мають відношення до ігрового процесу й мають просторові координати.

Павн, пішак (pawn) – фізична модель гравця або об'єкта, керованого штучним інтелектом. Назва походить від англ. Pawn – той, ким маніпулюють (pawn можна перевести також як пішак, тому такий об'єкт без якої-небудь моделі виглядає як пішак). Метод керування описаний спеціальним об'єктом, такий об'єкт називається контролером. Контролер штучного інтелекту описує лише загальну поведінку пішака під час ігрового процесу, а такі параметри як «здоров'я» (кількість пошкоджень, після яких пішак перестає функціонувати) або, наприклад, відстань, на якій пішак звертає увагу на звуки, задаються для кожного об'єкта окремо.

Світ, рівень (world, game level) – об'єкт, що характеризує загальні властивості «простору», наприклад, силу тяжіння й туман, у якому розташовуються всі актори. Також може містити в собі параметри ігрового процесу, як, наприклад, ігровий режим, для якого призначений рівень.

Для роботи із простими й, як правило, нерухомими елементами ігрового простору (наприклад, стіни) використовується бінарна розбивка простору – увесь простір ділиться на «заповнене» і «порожнє». В «порожній» частині простору розташовуються всі об'єкти а також тільки в ній може перебувати «точка спостереження» при відмальовці сцени. Можливість повного або часткового поміщення об'єктів в «заповнену» частину простору не виключається, однак може привести до неправильної обробки таких об'єктів (наприклад, розрахунки фізичної взаємодії) або неправильної відмальовки у випадку поміщення туди «точки спостереження» (наприклад,

ефект «залу дзеркал»). Усі пішаки, що потрапляють в «заповнену» частину простору, відразу «гинуть».

Переваги пакета:

- умовно-безкоштовна версія для комерційного використання;
- підтримка усіх сучасних форматів експорту тривимірних моделей;
- висока якість рендерингу моделей та ефектів.

Недоліки:

- середні вимоги до характеристик комп'ютера розробника;
- закритий програмний код.

Детально розглянувши перераховані ігрові рушії був обраний програмний пакет Unity. Завдяки своїй ієрархічності, дружньому користницьму інтерфейсу, підтримці усіх сучасних форматів експорту тривимірних моделей та простоті Unity дозволяє швидко отримати необхідний результат та освоїти ігровий рушії за короткий відрізок часу. Також Unity має чудово розвинуте спільнство та добра організовану документацію та API, де кожен користувач може отримати відповідь на свої запитання.

3 ПРОЕКТУВАННЯ ТРИВИМІРНИХ МОДЕЛЕЙ

3.1 Проектування ігрового простору

Насамперед на етапі проектування необхідно проаналізувати концепцію майбутньої гри. У нашому випадку це симулятор виживання в холодних зимових умовах. В процесі проектування було створено кілька концепт-артів (рис. 3.1 – 3.2):



Рисунок 3.1 – Концепт-арт комп'ютерної гри



Рисунок 3.2 – Концепт-арт комп'ютерної гри

Завдяки концепт-артам можна було отримати перше уявлення про створення моделей для такої гри. Для цього необхідно точно виділити жанри і опис проекту. Жанри: RPG, Action, Adventure, Indie, Survival, Sandbox. Тип: одиночна гра з видом від першої особи.

Короткий опис: Гравцеві належить пережити зиму в безлюдному куточку світу і здобути навички виживання в екстремальних погодних умовах.

Після опису і створення концептів для комп'ютерної гри, була створена карта місцевості для правильного позиціонування майбутніх об'єктів на сцені, а також для точного уявлення про кількість комп'ютерних моделей.

Завдяки інформації отриманої на етапі проектування гри можна приступати до проектування моделей.

3.2 Проектування тривимірних моделей оточення

На етапі проектування моделей необхідно виділити три основні типи моделей: дерева, споруди, внутрішньоігрові предмети. Завдяки розумінню того, для чого буде використовуватися майбутня модель, можна досягти необхідного ступеня деталізації при створенні моделі. Наприклад, якщо це частина оточення, то гравцеві необов'язково представляти його деталізованим, так як гравець може знаходитися в будь-якому місці карти і виконувати різні дії не звертаючи уваги на деталі. У випадку з внутрішньоігрові предметами, їх необхідно зробити досить деталізованими, так як гравець буде постійно взаємодіяти з ними, а також постійно звертати уваги на деякі з них. В процесі проектування тривимірних моделей оточення були створені такі концепт-арти (рис. 3.3 та рис 3.4):

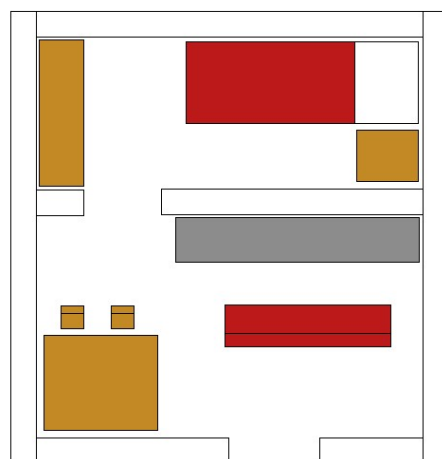


Рисунок 3.3 – Концепт хатини

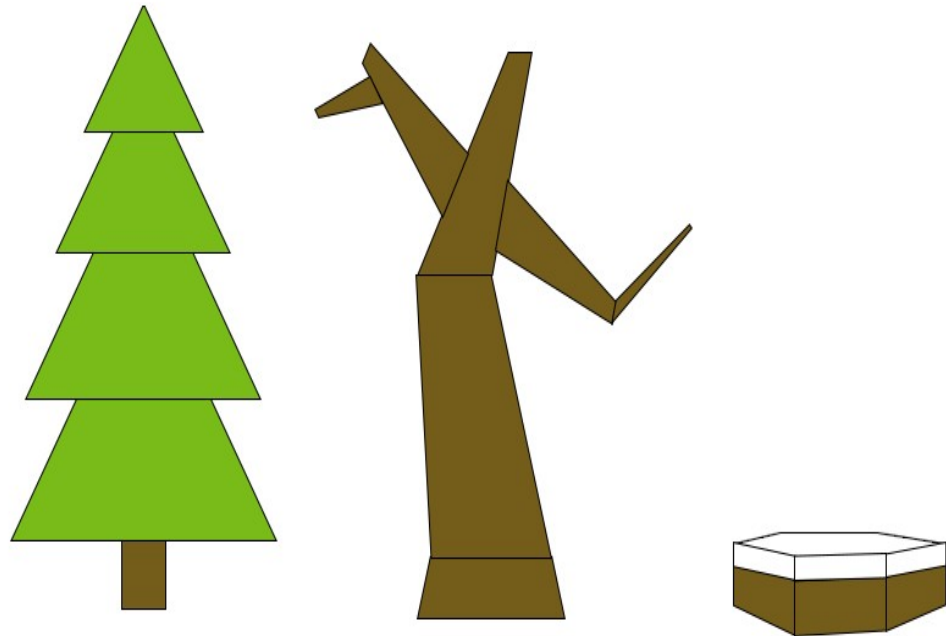


Рисунок 3.4 – Концепт різновиду лісної рослинності

Наступним етапом є представлення моделі ігрового персонажа.

3.3 Проектування моделі ігрового персонажа

На цьому етапі завданням є представити майбутню модель ігрового персонажа з можливістю анімації руху і взаємодії з навколишнім середовищем. У професійній індустрії такі моделі створюються за допомогою референсів.

Референс – це вторинне зображення; малюнок або фотографія, які художник або дизайнер вивчає перед роботою, щоб точніше передати деталі, отримати додаткову інформацію, ідеї.

Референсом можуть бути фотографії людини в різних позах, фотографії тільки рук або інших частин тіла, необхідних для вивчення і відтворення анатомії. Також референс використовують в анімації для відтворення великої кількості кадрів.

Зазвичай в них входить:

- повне зображення персонажа;
- деякі характерні особливості;
- опис розмірів;
- текстові коментарі деяких об'єктів.

Для даного проекту також був обраний референс живої людини знятий в професійній студії (рис. 3.5):

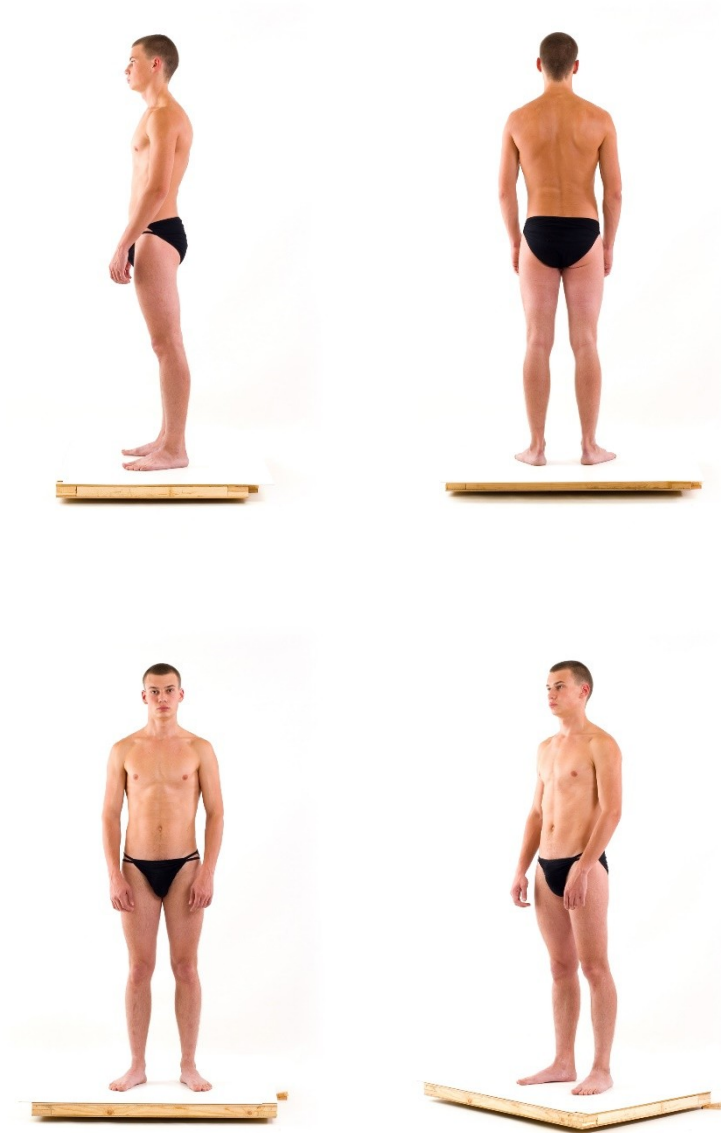


Рисунок 3.5 – Референс ігрового персонажу

Так як ігровий персонаж не повинен бути деталізованим або оснащеним спеціальними предметами в первісному вигляді, даного референсу цілком достатньо для відображення коректної моделі і подальшої анімації її рухів.

4 МОДЕЛЮВАННЯ ТРИВИМІРНОГО ПРОСТОРУ

4.1 Створення моделей ігрового простору

Розібравшись з концептуальної частиною необхідно перейти до етапу моделювання, для цього використовується програмний пакет Blender.

Насамперед потрібно створити сцену і прибрати всі, що знаходяться в ній, предмети (куб, джерело світла, позиція камери), які відображені на рис. 4.1.

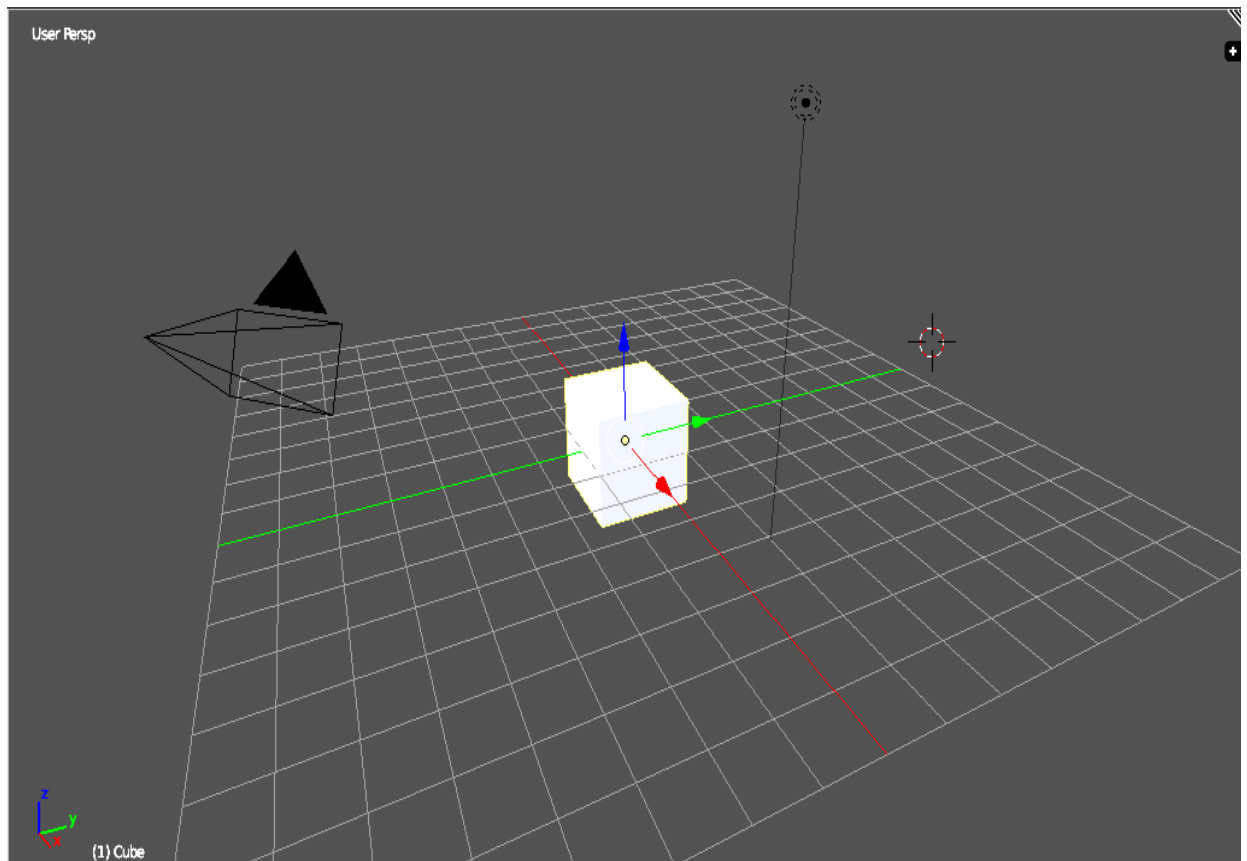


Рисунок 4.1 – Приклад сцени у редакторі Blender

Першою метою моделювання виберемо створення лісової хатини. Для початку необхідно з примітивів (куб, циліндр, площина, сфера) в Object Mode створити фундамент для будинку необхідної довжини і ширини. На цьому етапі визначається скільки місця займе наш будинок на сцені.

Для цього використовується кілька кубів, які в кінцевому підсумку деформуються за допомогою Edit Mode (рис. 4.2).

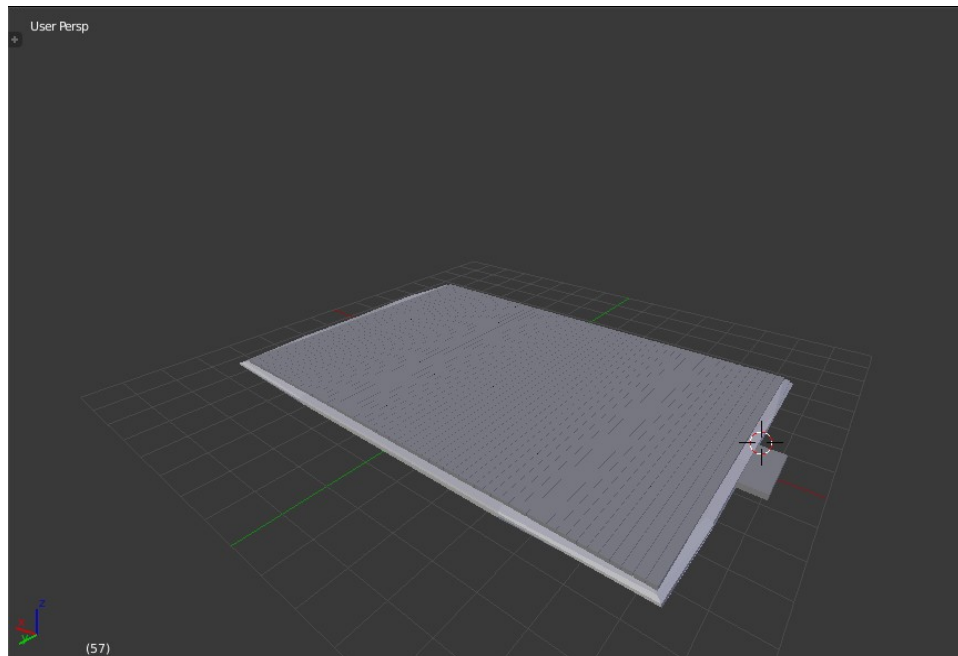


Рисунок 4.2 – Модель фундаменту для хатини в Blender

Після того, як розмічена область для хатини, необхідно додати стіни тим ж шляхом, що і фундамент. Створивши кілька примітивів і визначившись яким будинок буде в висоту, розмістимо їх над фундаментом. Потім за допомогою інструменту Knife в Edit Mode необхідний вирізати отвори в створених примітивах для розташування дверного отвору. В результаті отримуємо подобу хатини без даху (рис. 4.3).

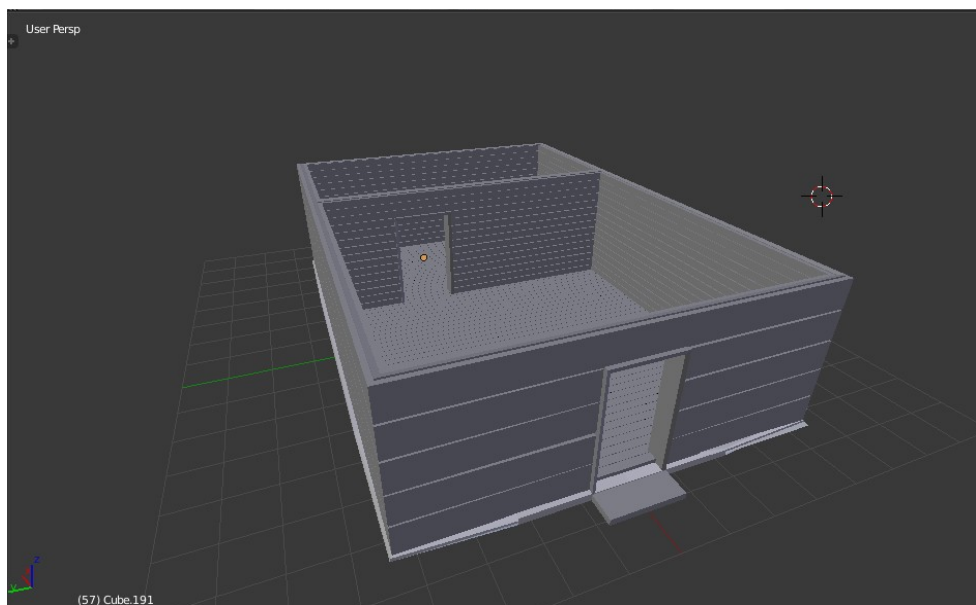


Рисунок 4.3 – Модель основної частини хатини в Blender

Наступним етапом є закриття отриманої коробки, тобто додавання даху. Для цього необхідно в Object Mode додати кілька примітивів, повернувши їх на 30 градусів і розташували їх над стінами схрещеними між собою. В результаті отримуємо дах для нашої хатини (рис. 4.4).

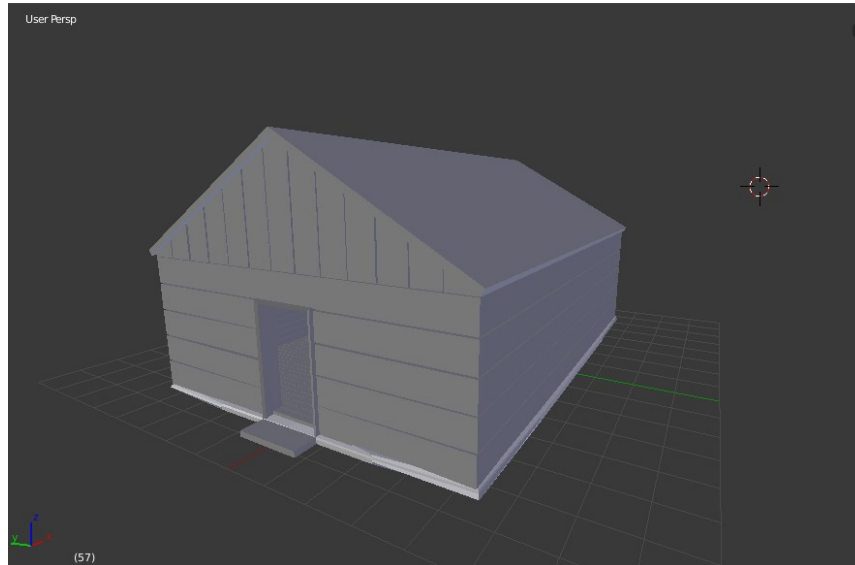


Рисунок 4.4 – Готова модель хатини в Blender

Завершальним етапом в моделюванні хатини є експорт в ігровий рушій Unity. Завдяки тому, що Unity добре сприймає моделі різних форматів, для додавання моделі на сцену необхідно перетягнути вихідний .blend файл з Blender'a. Завдяки цьому одержуємо результат на сцені Blender.

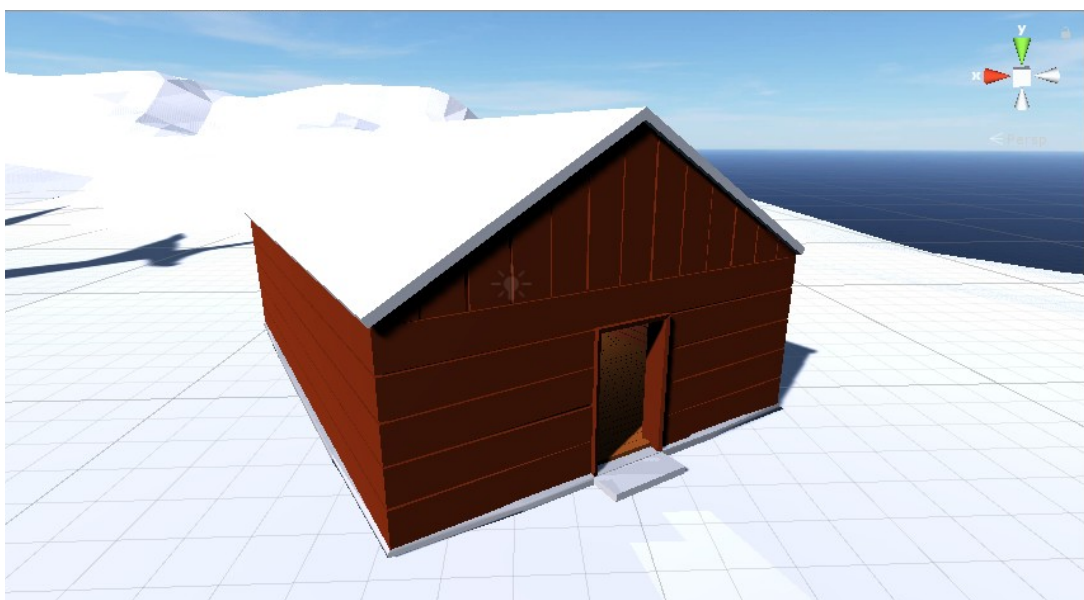


Рисунок 4.5 – Зовнішній вигляд моделі хатини в Unity

Для того, щоб на сцену Unity не потрапили зайві предмети необхідно використати стандартний експортер Blender у формат .fbx.

Налаштувавши експорт в форматі .fbx необхідно зберегти файл на жорсткий диск в папку з проектом Unity. Так як на сцені Blender відсутня камера або додаткові джерела світла через їх непотрібність, вказуємо для експорту тільки Mesh, завдяки чому в файлі .fbx залишиться лише модель хатини без зайвих об'єктів (рис. 4.6).

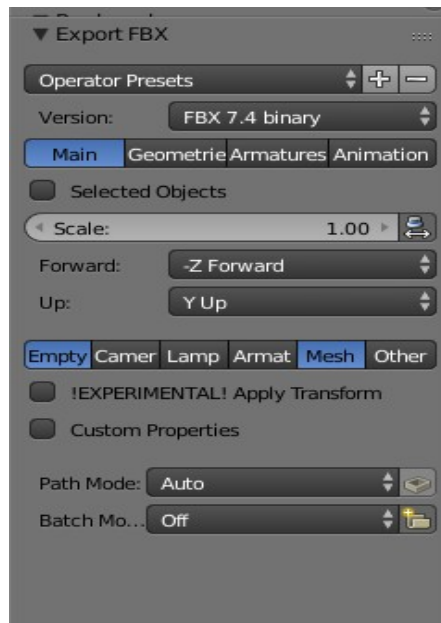


Рисунок 4.6 – Вікно налаштувань експорту в Blender

Розмістивши джерело світла всередині хатини, є можливість оглянути її повністю (рис. 4.7).

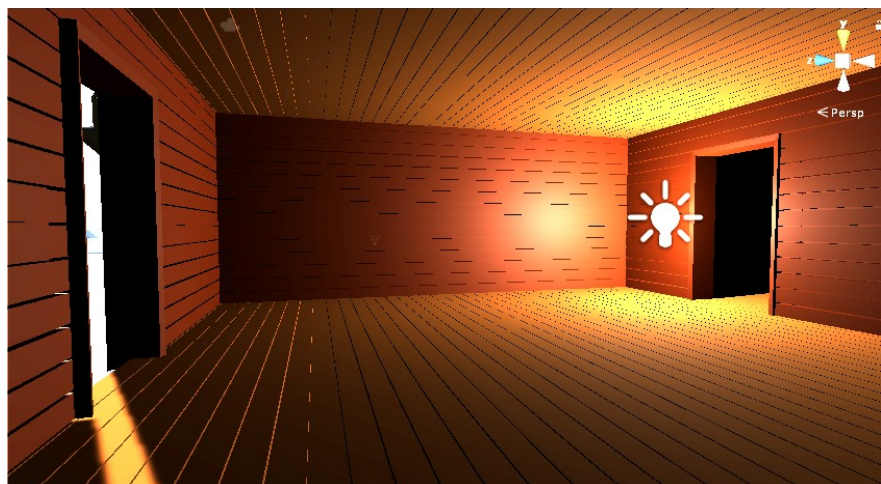


Рисунок 4.7 – Внутрішній вигляд моделі хатини в Unity

На даному етапі хатина є достатньо реалістичною за винятком її порожнечі. Для цього необхідно заповнити інтер'єр типовими для неї предметами і речами. Використовуючи стандартні інструменти Blender, такі як Extrude і Subdivide можна домогтися необхідної деталізації при створенні як дрібних так і великих об'єктів деформуючи їх необхідним чином. Таким чином були створені і експортовані всі необхідні моделі для обстановки інтер'єру та деталізації хатини (рис. 4.8).

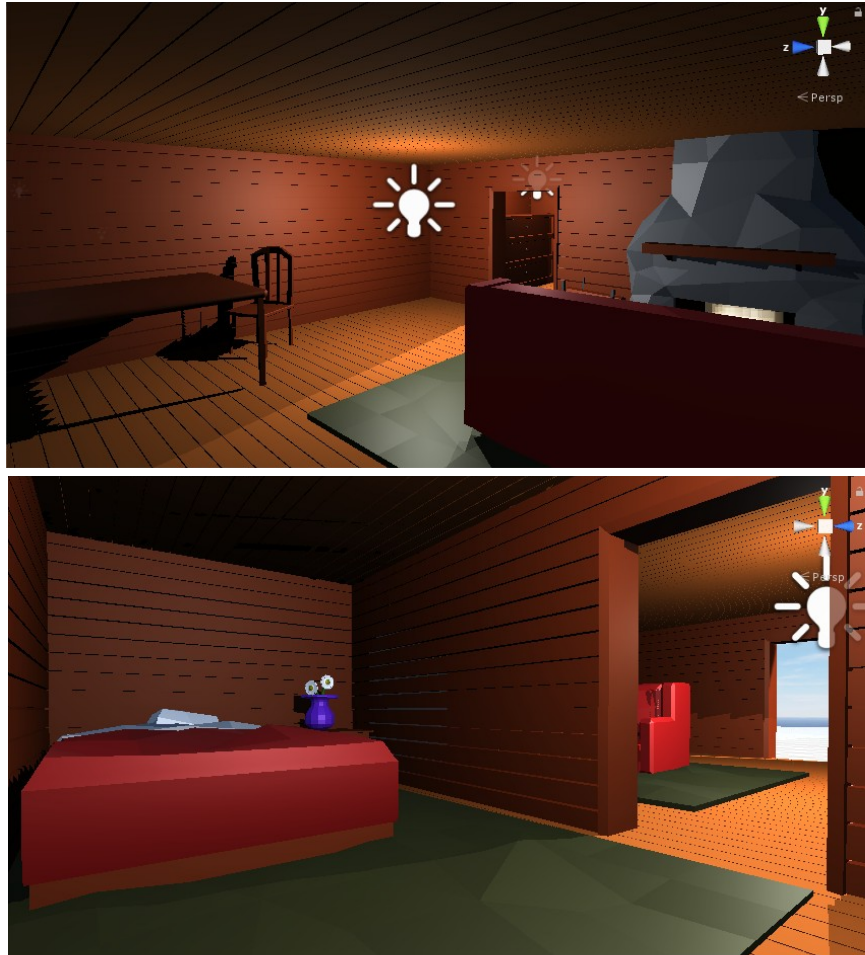


Рисунок 4.8 – Вигляд інтер'єру хатини в Unity

Згідно концепції гри, наступним кроком буде створення рослинності та ландшафту, тобто створення моделей оточення. Насамперед необхідно створити ландшафт для подальшого розміщення моделей на ньому. Ландшафт може бути створений як у вбудованому редакторі рушія Unity, так і безпосередньо в програмному пакеті Blender. Для того, щоб ландшафт був деталізованим і досить гнучким при роботі з ним в своєму проекті я використовував Blender. Для цього досить створити порожню сцену, додати

прімінівную площину (Plane), потім за допомогою інструменту Subdivide розділити поверхню площині на кілька десятків полігонів (рис. 4.9).

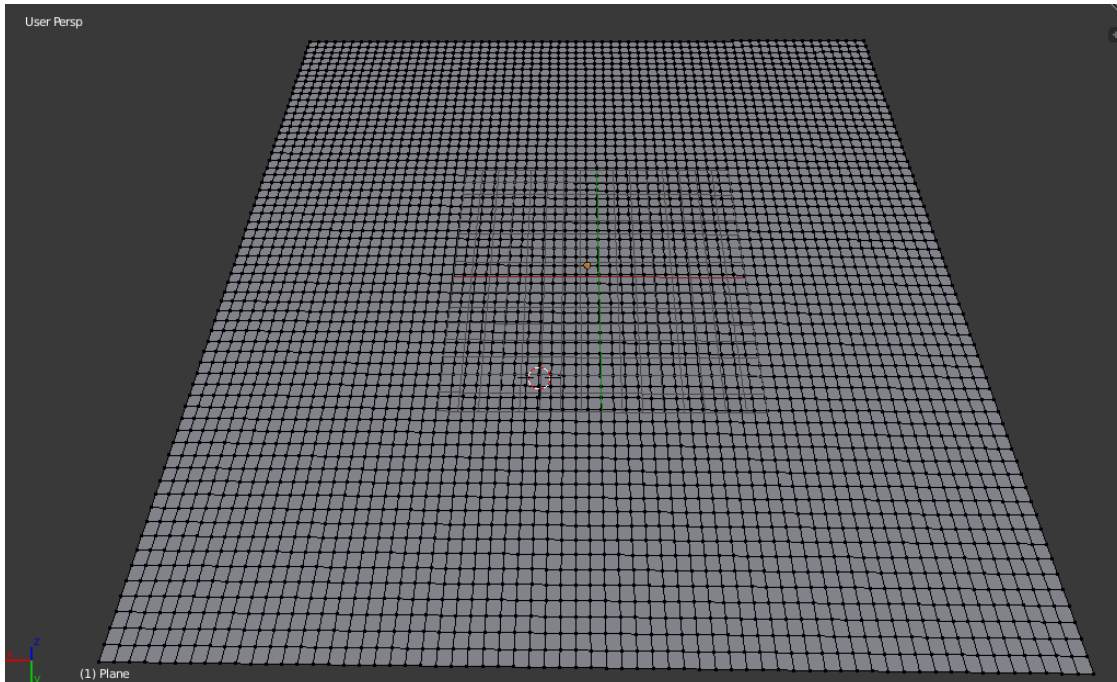


Рисунок 4.9 – Площина після розбивання в Blender

Після того як ми отримали площину з достатньою кількістю полігонів, використовуємо Sculpture Mode і за допомогою пензля деформуємо поверхню. В даному режимі з'являється можливість в необхідному місці підняти або опустити полігони з різною інтенсивністю, завдяки чому отримуємо реалістичну поверхню для імпорту в Unity (рис. 4.10).

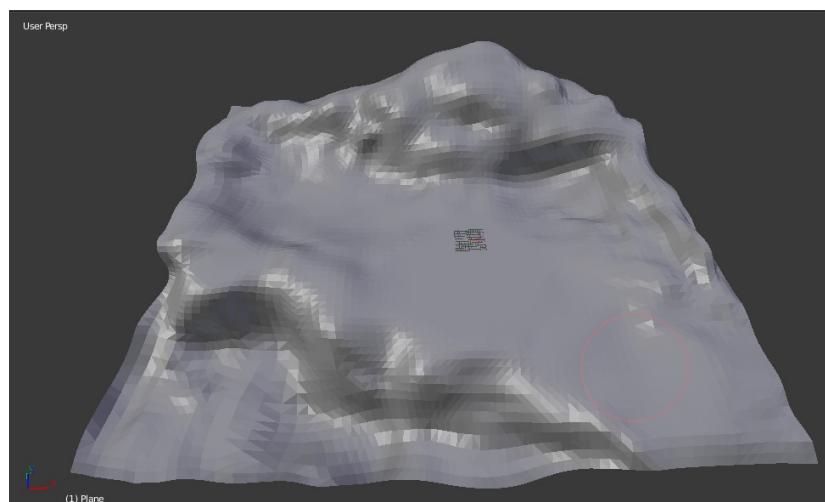


Рисунок 4.10 – Деформована поверхність в Blender

По завершенню маніпуляцій з площиною переходимо до імпорту в Unity для перевірки коректного відображення (рис. 4.11).

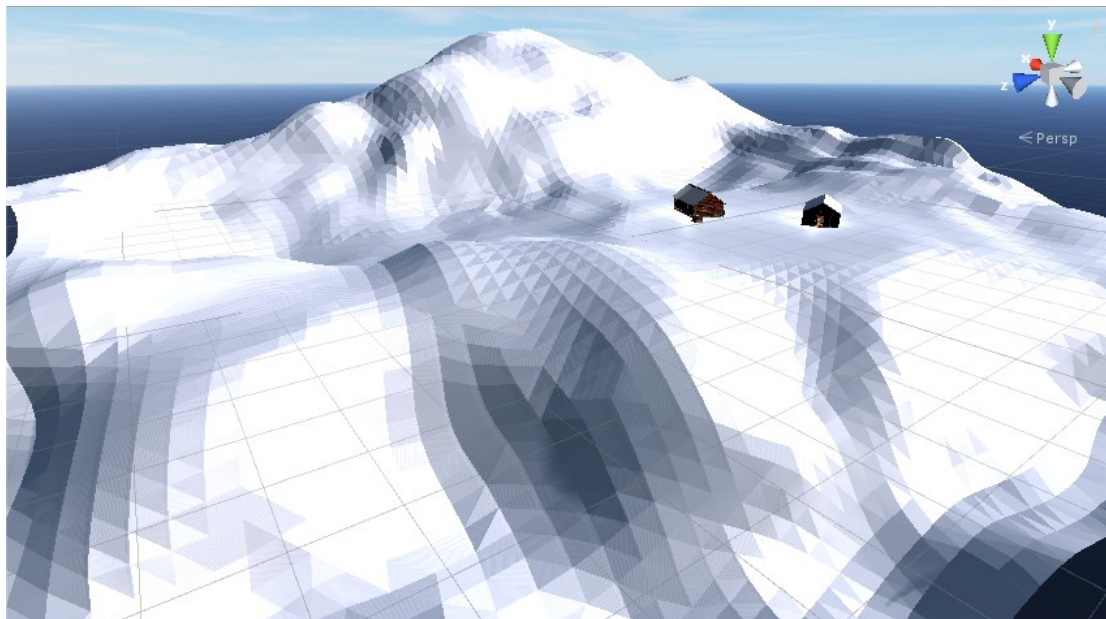


Рисунок 4.11 – Перевірка поверхні в Unity

Після створення поверхні можна приступити до створення об'єктів на ній тобто рослинності і додаткових об'єктів для поліпшення візуалізації загальної картини. Насамперед була створена кам'яна поверхню за допомогою примітивів і Sculpture Mode і імпортована на ігрову сцену Unity (рис. 4.12).

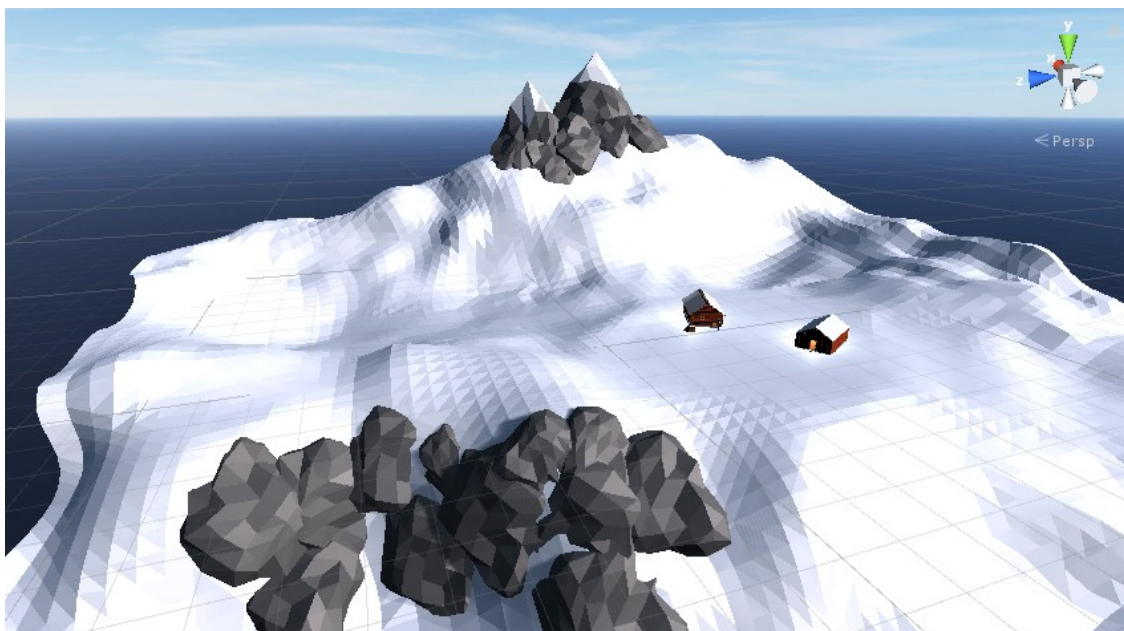


Рисунок 4.12 – Каміння та скали в Unity

Останнім етапом є створення різних видів лісової рослинності. У нашому випадку це 3 різних видів дерев. За допомогою примітивів створюємо перше дерево (рис. 4.13).

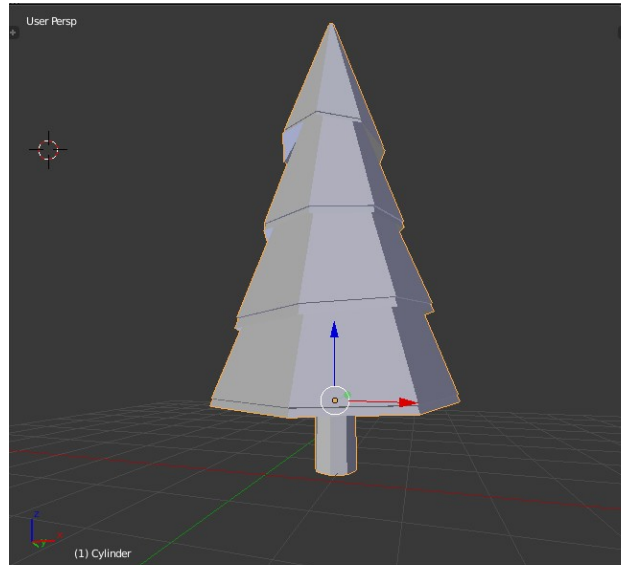


Рисунок 4.13 – Дерево, створене в Blender

Після того як ми створили модель, необхідно приступити до її зовнішньої деталізації а саме до покриття її текстурою. Для цього в редакторі Blender необхідно створити текстуру необхідного розміру (в нашому випадку 1024x1024). Для того, щоб текстура правильно відображалася на тривимірній моделі виконується UV-розгортка моделі (рис. 4.14). Для цього спеціальним інструментом в Blender ми розрізаємо модель по шву для того, щоб вона правильно відображалася в площині.

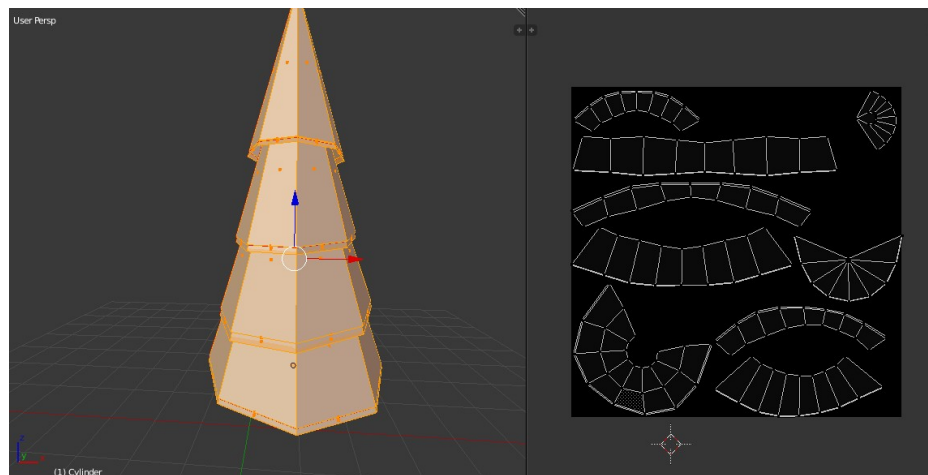


Рисунок 4.14 – UV-розгортка в Blender

Після створення розгортки необхідно за допомогою інструментів Blender створити Ambient Occlusion текстуру для правильного відображення тіней в місцях, куди не потрапляє світло. В результаті чого отримуємо чорно-білу текстуру затінення моделі (рис. 4.15).

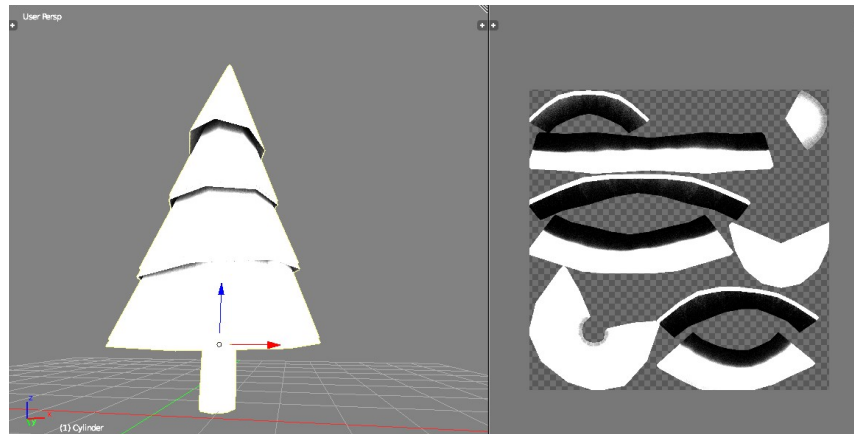


Рисунок 4.15 – Ambient Occlusion текстура в Blender

Після цього зберігаємо отримані текстури в .png форматі зображення і приступаємо до їх редагування в будь-якому графічному редакторі, наприклад Adobe Photoshop. Для правильного відображення кольору текстури разом з тінню отриманої від Ambient Occlusion текстури необхідно розмістити шар з АО-текстурою поверх інших і змінити його режим змішування (blending mode) в режим Множення (Multiply). Після чого шари, що знаходяться під ним можна заповнити необхідними кольорами, в результаті чого отримаємо коректну текстуру дерева (рис. 4.16).

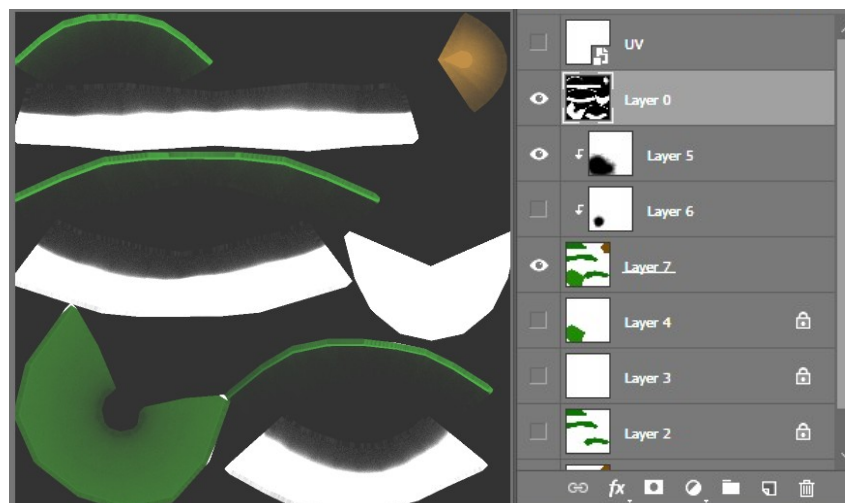


Рисунок 4.16 – Текстура в Adobe Photoshop

Після того, як закінчено редагування текстури, необхідно зберегти її в .png формат і експортувати назад в Blender (рис. 4.17).

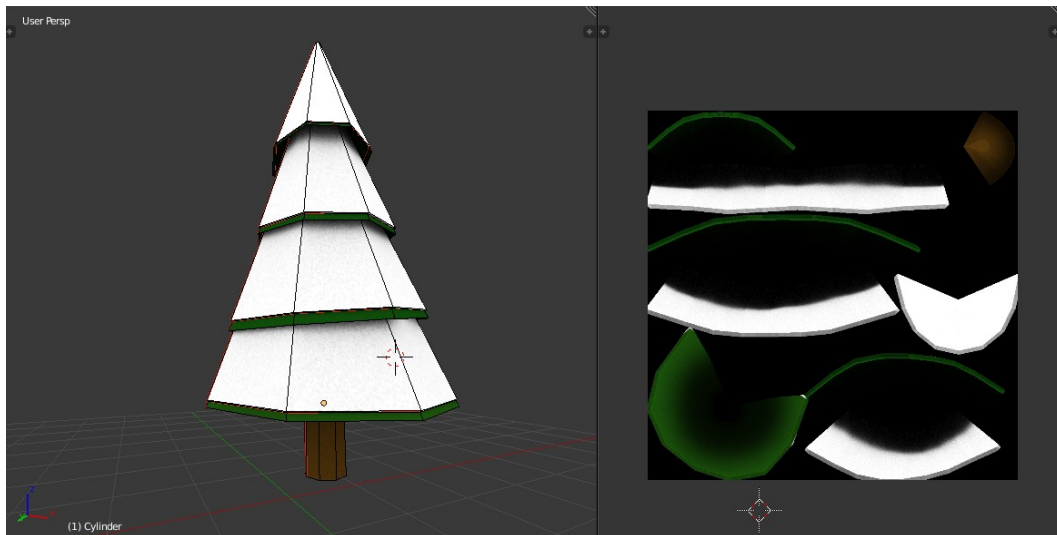


Рисунок 4.17 – Текстура дерева в Blender

Переходимо до створення такого вигляду дерев. Для цього знову створимо порожню сцену в Blender, розмістивши на ній примітив (циліндр) переходимо до його деформації. Для цього використовуємо інструмент Extrude, службовець для «видавлювання» полігонів. В результаті чого отримуємо необхідну форму (рис. 4.18).

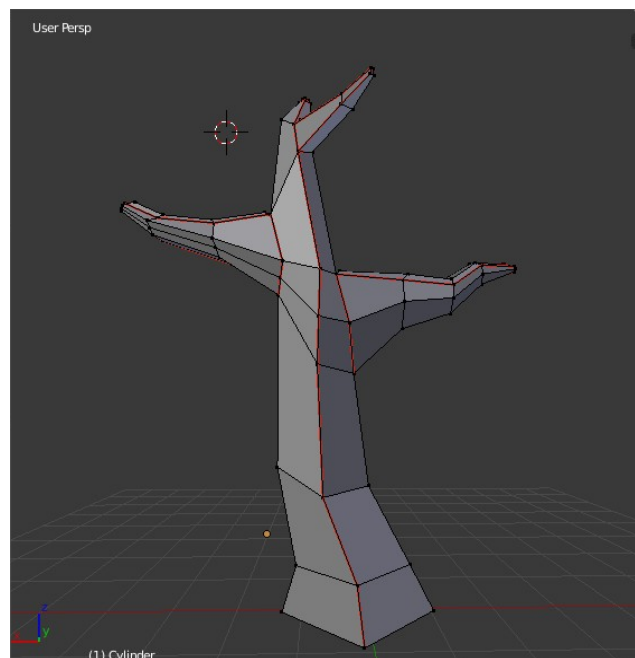


Рисунок 4.18 – Дерево в Blender

Далі виконаємо UV-розгортку моделі і створення для неї текстури (рис. 4.19).

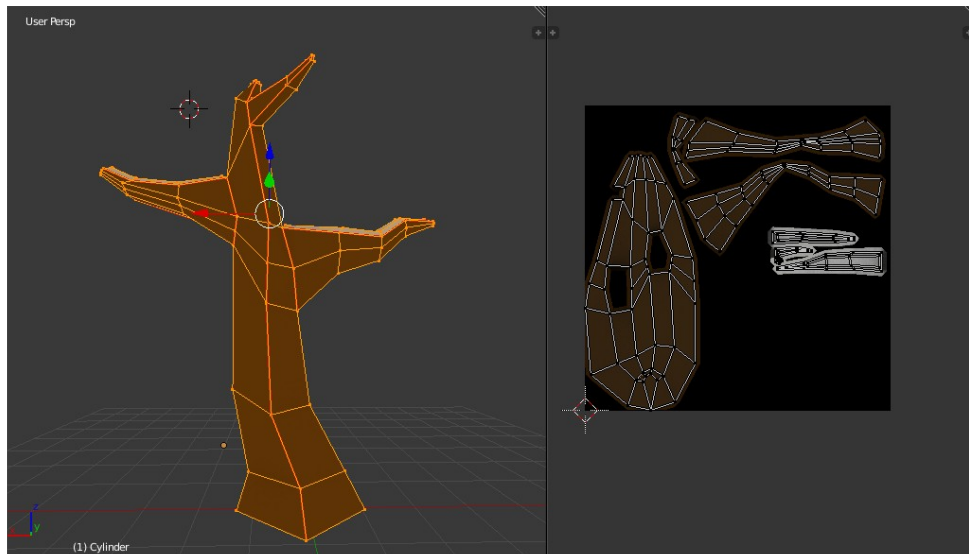


Рисунок 4.19 – Модель з текстурою в Blender

Аналогічним чином створюється третій вид рослинності (рис. 4.20).

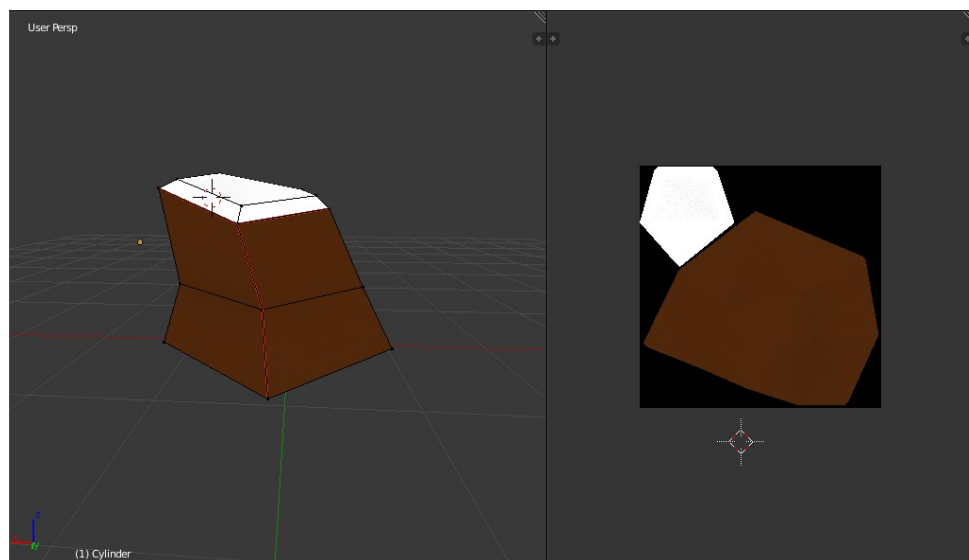


Рисунок 4.20 – Пеньок з текстурою в Blender

Так як моделі виглядають коректно, іпортуємо їх на ігрову сцену в Unity для перевірки відображення. Для цього переносимо модель на сцену і в параметрах матеріалу вказуємо файл з текстурами для кожної моделі. У деяких випадках буває, що модель має прозорі елементи, яких не повинно

бути. В такому випадку необхідно в редакторі Blender «перевернути» полігони за допомогою інструменту Make Normals Consistent або за допомогою клавіш Ctrl + N (рис. 4.21).

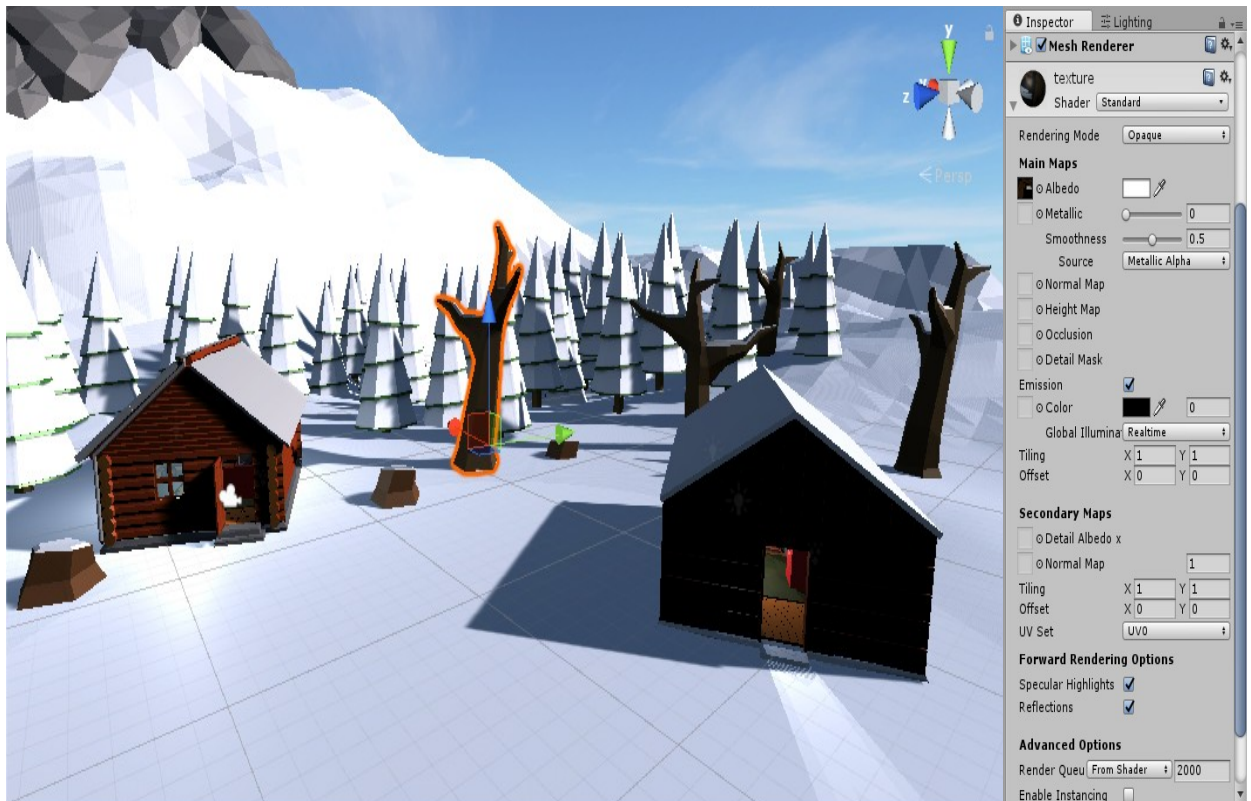


Рисунок 4.21 – Ігрова сцена в Unity з налаштуваннями моделі

4.2 Створення моделі ігрового персонажа

Етап створення тривимірної моделі ігрового персонажа починається зі створення порожньої сцени з примітивом (кубом) який буде використовуватися в якості основи для цілої моделі.

Далі необхідно за допомогою інструментів Blender імпортувати референс людини для точного уявлення про анатомію людини.

Далі для примітиву використовується модифікатор Mirror для дзеркального відображення, так як людське тіло є симетричним. Після цього за допомогою інструментів Subdivide, Loop Cut and Slide і Extrude досягається необхідна форма тіла, «видавлюються» руки, ноги і шия.

Потім створюється додатковий примітив (сфера) для голови. Так як наша гра не передбачає відображення персонажа в зовнішньому світі і геймплей відображається від першої особи, лицьова частина моделі може бути відсутня (рис. 4.22).

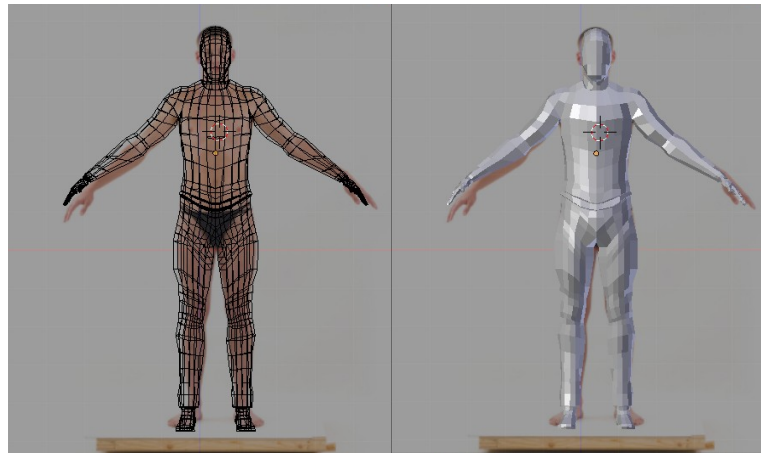


Рисунок 4.22 – Ігрова модель персонажу в Blender

Після того, як була створена модель, вона експортується у пакет для скульптингу ZBrush. За допомогою цього пакету можна підкреслити необхідні риси моделі а також виправити помилки та додати деталів моделі. За допомогою панелі Geometry додамо кількість полігонів для збільшення детаізації. Для цього слугує налаштування SDivide, яке виконує розбивання полігонів на декілька частин, збільшуючи їх кількість. Завдяки цьому процесу модель стає бiль гладкою та деталізованою, що дозволяє виконувати гнучкі маніпуляції пензлями з полігональною сіткою. За допомогою інструменту Brush, використовуючи різні насадки (Clay, Blob, Polish, Planar, Smooth) додамо деталі для нашої моделі, виконавши форму, що буде нагадувати одяг (рис. 4.23).

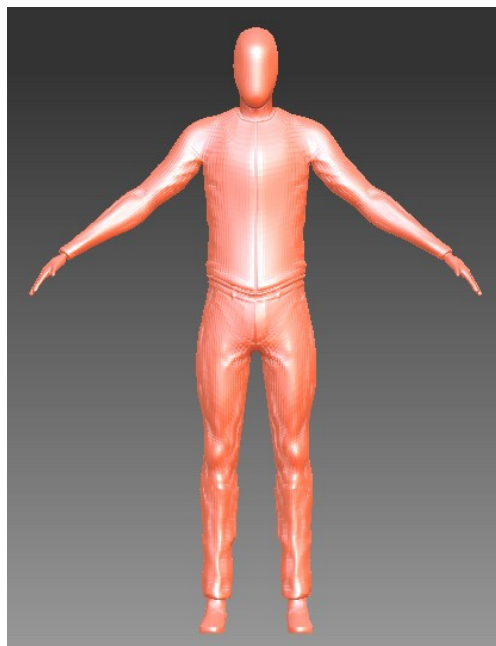


Рисунок 4.23 – Модель ігрового персонажу в ZBrush

По завершенню роботи над моделлю, виконується експорт до програмного пакету Blender, де продовжується робота над текстурованням. Для цього виконується UV-розгортка моделі, розрізаючи її на частини та утворюючи поле для текстуровання. За чим створюється Ambient Occlusion текстура для правильної світлопередачі. Відреагувавши отримані слої в графічному редакторі отримуємо повноцінну текстуру для експорту в Blender (рис. 4.24).



Рисунок 4.24 – Текстура ігрового персонажу в Blender

Після перевірки текстури на відображення переходимо до етапу створення анімації моделі. Так як у людей (гуманоїдів) схожа будову тіла, вони мають практично однакові методи для анімації. Завдяки цьому в Blender існує величезна кількість інструментів і плагінів для спрощення створення анімації ігрового персонажа, один з яких Rigify.

Даний плагін дозволяє створити готову структуру людських кісток для подальшої анімації. У Blender все анімації відбуваються за допомогою маніпуляції над поліготами. Для спрощення цього завдання в пакеті є елемент Bone (кістка).

Розмістивши «кістки» в ключових місцях анімації ми прив'язуємо полігони в певній області для створеної «кісток» за допомогою яких ми будемо їх переміщувати. Етап розміщення і прив'язки кісток до полігональної сітки називається ріггінг (рис. 4.25).

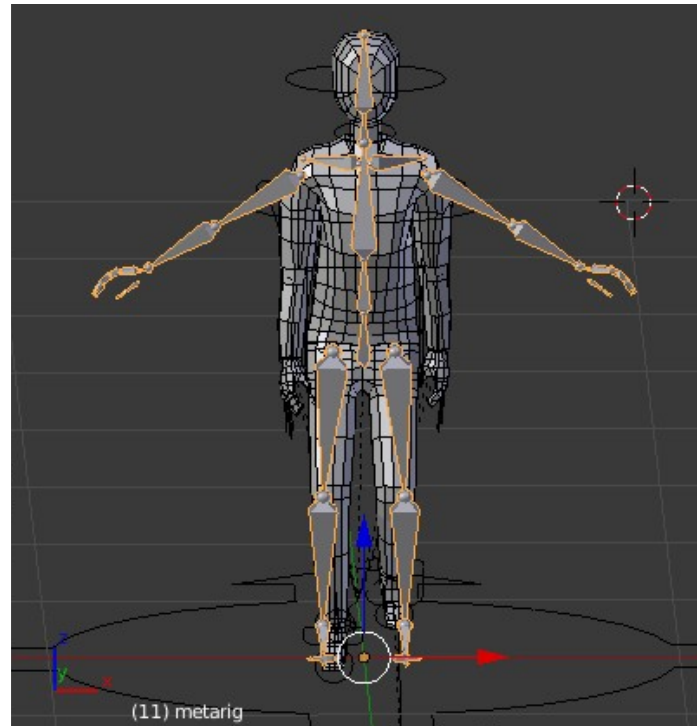


Рисунок 4.25– Рігінг ігрового персонажу в Blender

Далі на тимчасовій панелі (time-line panel) переміщається «кістки» відзначаємо ключові точки в якості позиції, в якій вони будуть знаходитися в конкретний відрізок часу. Завдяки цьому кожна «кістка» переходить з початкового стану в заданий що і є анімацією (рис. 4.26).



Рисунок 4.26 – Анімація ігрового персонажа в Blender

Після створення анімації модель експортується в Unity для її перевірки (рис. 4.27).

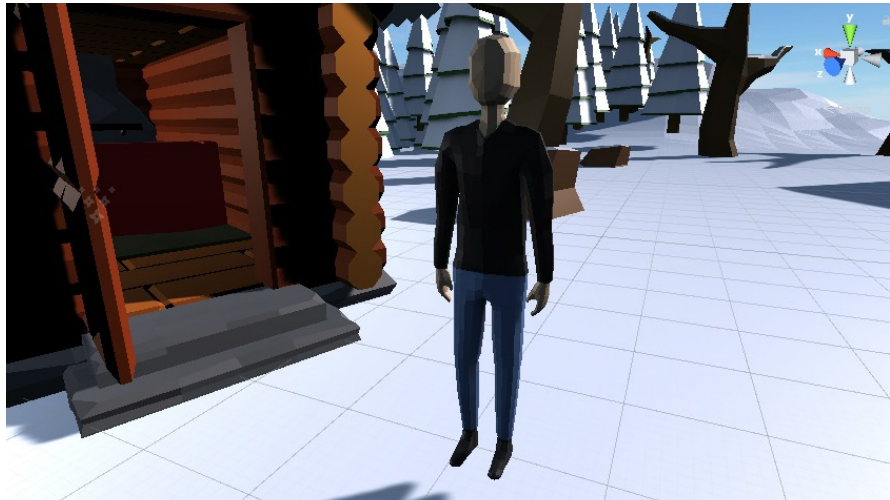


Рисунок 4.27 – Ігрова модель в Unity

На даному етапі готові всі моделі для подальшої роботи над створенням гри. Завершенням роботи може слугувати створення скриптів для анімації моделі при переміщенні ігрового персонажу по сцені а також для взаємодії його з навколишнім середовищем.

ВИСНОВКИ

В результаті виконання дипломної роботи було змодельовано та здійснена програмна реалізація інтерактивного ігрового середовища.

В ході розробки була розглянута предметна область та сформульовані вимоги до проектування та створення системи.

При здійсненні дипломного проектування були розглянуті програмні засоби для здійснення моделювання інтерактивного ігрового середовища та обрані продукти Blender та ZBrush. Завдяки використанню сучасних засобів реалізації створені моделі малого розміру для можливості здійснення рендерінгу на комп'ютерах з мінімальними характеристиками. Розроблені моделі представляють собою розташування в межах контурів відповідних векторних карт об'єктів. В результаті створений набір моделей з необхідними анімаціями та текстур для створення на їх базі повноцінної інтерактивної гри. Всі створені моделі відповідають жанру і стилістиці та сприяють передачі сюжетної лінії.

Використання програмного пакету Blender дозволило реалізувати всі необхідні елементи тривимірного простору достатньою для проекту деталізацією.

В даному проекті була затронута актуальна і розвиваючись область індустрії розробки комп'ютерних ігрових програм.

В рамках дипломного проекту були отримані наступні теоретичні і практичні результати:

- проведено аналіз основних етапів розробки ігор і створення тривимірних моделей;
- розглянуті різні графічні пакети і їх можливості, що дозволяють зробити програмну реалізацію моделей;
- в ході аналізу предметної області визначені основні цілі і завдання;
- розглянуті основні складові тривимірних моделей;
- розроблені моделі тривимірного простору на базі створених раніше концептуальних зображень;
- проведено тестування моделей на правильність відображення в ігровому рушії.

ПЕРЕЛІК ПОСИЛАНЬ

1. Коротаев А.В., Малков С.Ю., Гринин Л.Е. История и Математика: Анализ и моделирование социально-исторических процессов. – изд.3 – М.: URSS, 2016. 360 с.
2. В. Є. Михайленко, В. В. Ванін, С. М. Ковальов. Інженерна та комп'ютерна графіка: підруч. для студ. ВНЗ. – 5-те вид. – К.: Каравела, 2010. – 360 с.
3. M. De Aguilera and A. Mendiz, "Video games and education: (Education in the Face of a 'Parallel School'), " Computers in Entertainment (CIE) 1, no. 1 (2003). P. 74 – 85.
4. Дж. Ли, Б. Уэр. Трёхмерная графика и анимация. – 2-е изд. – М.: Вильямс, 2002. – 640 с.
5. В. П. Иванов, А. С. Батраков. Трёхмерная компьютерная графика. Под ред. Г. М. Полищука. – М.: Радио и связь, 1995. – 224 с.
6. L. Piegl, W. Tiller: The NURBS Book, Berlin: Springer-Verlag. 1995 – 1997 – 135 p.
7. Immel, David S.; Cohen, Michael F. & Greenberg, Donald P. (1986), "A radiosity method for non-diffuse environments". – N.Y.: Siggraph 1986 – 133 p.
8. Kajiyu, James T., "The rendering equation". – N.Y.: Siggraph 1986 – 143 p.
9. Тозик В.Т. Л.М. Корпан. Компьютерная графика и дизайн: Учебник для нач. проф. Образования. – М.: ИЦ Академия, 2013. – 208 с.
10. Залогова Л.А. Компьютерная графика. Элективный курс: Практикум. – М.: БИНОМ. ЛЗ, 2011. – 245 с.
11. Емельянов С.Г., П.Н. Учаев, К.П. Учаева Начертательная геометрия. Инженерная и компьютерная графика в задачах и примерах: Учебное пособие; под общ. ред. проф. П.Н. Учаева. – Ст. Оскол: ТНТ, 2013. – 288 с.
12. Дегтярев В.М. Инженерная и компьютерная графика: Учебник для учреждений высшего профессионального образования. – М.: ИЦ Академия, 2011. – 240 с.