

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД «УНІВЕРСИТЕТ «КРОК»
Фаховий коледж Університету «КРОК»

ДИПЛОМНА РОБОТА
За темою

«Розробка гри на платформі Unreal Engine»

Студент 4 курсу групи ППЗ-20к-1

Пишнюк Руслан Юрійович

(прізвище, ім'я та бо батькові)


(підпис студента)

Керівник дипломної роботи

к.т.н., доцент

(посада керівника)

Добришин Ю. Є.

(прізвище, ім'я та бо батькові керівника)

До захисту

(резолуція «До захисту»)

10.06.2024
(дата)


(підпис викладача)

Київ, 2024 рік

Скорочення

FPS (від англ. frames per second) — кількість кадрів в секунду на екрані монітора.

Рендерінг (від англ. rendering) — процес отримання зображення за моделлю з допомогою комп'ютерної програми.

Антиаліасінг (від англ. anti-aliasing) — технологія, що використовується в обробці зображень з метою зробити межі кривих ліній більш гладкими прибираючи зубці, що виникають на краях об'єктів.

Інтерполяція — математична функція, яка допомагає знайти проміжні значення між точками.

Зміст

Вступ	4
Розділ 1. Аналіз існуючої інформації щодо теми дипломної роботи	5
1.1. Порівняльний аналіз існуючої інформації та рішень	5
1.2. Постановка завдання на проектування.....	8
Розділ 2. Проектні і технічні рішення. Види забезпечення.....	9
2.1. Інформаційне забезпечення.....	9
2.2. Математичне забезпечення.....	11
2.3. Програмне забезпечення.....	12
Розділ 3. Реалізація гри на Unreal Engine.....	13
Розділ 4. Створення ігрових рівнів	19
Розділ 5. Розробка персонажів та об'єктів.....	25
Розділ 6. Тестування та виправлення помилок.....	29
Розділ 7. Посібник для гравця	34
Висновок.....	36
Перелік посилань	37
Додатки	38

Вступ

Ігрова індустрія у швидкому темпі розвивається останні десять років. Вже приблизно чотири мільярди людей грають у різноманітні ігри по всьому світу на різних пристроях. Тому ця галузь приваблює все більше та більше інвестицій і багато працівників ІТ сфери хочуть працювати у галузі розробки ігор.

Нарівні з кінематографом, ігри мають багато видів. Наприклад, шутери, пригодницькі ігри, стратегії, симулятори, екшн ігри, рольові ігри та ігри-головоломки.

У зв'язку з цим було прийнято рішення створити пригодницьку гру в жанрі фентезі на платформі Unreal Engine для персональних комп'ютерів. Двигун Unreal Engine було обрано тому, що у ньому є передові технології створення ігор, він постійно оновлюється та ним зручно користуватися.

Розділ 1. Аналіз існуючої інформації щодо теми дипломної роботи

1.1. Порівняльний аналіз існуючої інформації та рішень

Було проведено пошук вже існуючих пригодницьких ігор у жанрі фентезі з видом від першої та третьої особи. Найближчим аналогом можна назвати гру The Elder Scrolls V: Skyrim.

Перша версія The Elder Scrolls V: Skyrim була випущена в 2011 році компанією Bethesda Game Studios. Гра була створена на власному русії компанії Creation Engine.

Основними перевагами The Elder Scrolls V: Skyrim були:

1. Відкритий світ з багатьма локаціями такими як міста, печери, башти, великі гори та інші загадкові місця.
2. Велика кількість завдань, персонажів та предметів.
3. Система розвитку свого персонажа.
4. Свобода вибору завдань.
5. Гарна система битв з використанням різної зброї та магії.

У процесі написання дипломної роботи було прийнято рішення перейняти як можна більше переваг та орієнтуватися на перевірену часом Skyrim.

Також було проведено пошук та аналіз інших ігрових рушіїв. Найпопулярнішим конкурентом рушія Unreal Engine є рушієм Unity. Перша версія Unity була створена в 2005 році компанією Unity Technologies.

Основні переваги Unity:

1. Кросплатформеність.
2. Підтримка та спільнота розробників.
3. Розширення та магазин активів.
4. Широкий функціонал.
5. Швидкість розробки.

Основні недоліки Unity:

1. Великий розмір файлів.
2. Залежність від Unity.
3. Швидкість виконання.
4. Висока вартість.
5. Обмежена графічна якість.

Для порівняння рушіїв Unreal Engine та Unity треба описати переваги та недоліки Unreal Engine[2]. Перша версія Unreal Engine була створена в 1998 році компанією Epic Games[1].

Основні переваги Unreal Engine:

1. Кросплатформеність.
2. Підтримка та спільнота розробників.
3. Розширення та магазин активів.
4. Інтуїтивний редактор.
5. Висока якість графіки.

Основні недоліки Unreal Engine:

1. Великий розмір файлів.
2. Залежність від Epic Games.
3. Обмежена безкоштовність.
4. Важкість вивчення для новачків.
5. Високі вимоги до ресурсів комп'ютера.

При порівнянні цих двох рушіїв можна зробити висновок, що Unreal Engine має вищу якість графіки та обмежену безкоштовність розробки, але має високі вимоги до ресурсів комп'ютера та потребує більше зусиль та часу для вивчення всього функціоналу.

1.2. Постановка завдання на проектування

Постановка завдання

Назва проекту: Розробка гри на платформі Unreal Engine.

Мета проекту: Розробити гру на платформі Unreal Engine.

Опис проекту:

- Пригодницька гра в жанрі фентезі з видом від першого та третього лица, яка розроблена на ігровому рушії Unreal Engine 4.

Основні функціональні вимоги:

1. Системи здоров'я та пошкодження в головного героя та ворогів.
2. Головне меню гри.
3. Великий рівень з різними ворогами та локаціями.
4. Сюжет з завданнями та діалоги.

Технічні вимоги:

1. Стабільна робота гри без помилок та несподіваного закриття гри.
2. Зручне керування головним героєм за допомогою клавіатури та миші.

Очікуваний результат: Створення гри на Unreal Engine, яка буде відповідати усім вимогам.

Розділ 2. Проектні і технічні рішення. Види забезпечення

2.1. Інформаційне забезпечення

Під час розробки гри на Unreal Engine значну роль у процесі навчання та отримання необхідних навичок відіграли різноманітні tutorіали на YouTube. Цей відеохостинг пропонує величезну кількість навчальних матеріалів, які стали незамінним ресурсом для вивчення як основних, так і просунутих аспектів роботи з Unreal Engine.

Перш за все, слід зазначити, що YouTube-канали, присвячені розробці ігор на Unreal Engine, охоплюють широкий спектр тем, починаючи від основ налаштування середовища і закінчуючи складними техніками створення анімацій та програмування геймплейних елементів. Серед найбільш корисних каналів варто виділити офіційний канал Unreal Engine, де представлено численні відеоуроки та вебінари від розробників Epic Games. Ці матеріали охоплюють ключові аспекти роботи з рушієм та надають офіційні рекомендації, які допомогли у розумінні базових концепцій та передових можливостей Unreal Engine.

Також велике значення мали tutorіали від незалежних розробників та ентузіастів. Серед них можна виділити такі популярні канали, як "Virtus Learning Hub", "Ryan Laley", "Unreal Engine Learning", "Matthew Wadstein" та інші. Відеоуроки на цих каналах відзначаються своєю практичною спрямованістю та доступністю для розуміння. Завдяки детальним покроковим інструкціям, вдалося опанувати техніки створення складних візуальних ефектів, налаштування фізики об'єктів, інтеграції анімацій та розробки штучного інтелекту для персонажів.

Окремо слід зазначити важливість туторіалів, які зосереджуються на вузькоспеціалізованих темах. Наприклад, туторіали з оптимізації продуктивності гри дозволили значно підвищити ефективність роботи проекту, зменшивши навантаження на апаратне забезпечення без втрати якості візуалізації. Туторіали з використання Blueprint-системи дали можливість розробляти складні логічні схеми без необхідності написання коду, що прискорило процес розробки та дозволило зосередитися на творчих аспектах гри.

Крім того, туторіали на YouTube стали платформою для ознайомлення з найкращими практиками та інноваційними підходами у розробці ігор. Відеоуроки часто включали приклади реальних проектів, що дозволило перейняти досвід професіоналів індустрії та застосувати його у власній роботі. Завдяки активним коментарям під відео, з'являлася можливість спілкування з авторами туторіалів та іншими глядачами, отримання додаткових порад та розв'язання проблем, що виникали під час роботи.

2.2. Математичне забезпечення

Математичні моделі у розробці гри на Unreal Engine використовуються при створенні фізики гри, анімацій та кінетики, обробці графіки, генерації та оптимізації мапи, розробці штучного інтелекту.

Наприклад, математична модель обробки графіки складається з:

1. Тривимірної координатної системи.
2. Моделей об'єктів.
3. Трансформації об'єктів.
4. Рендерінгу.
5. Освітлення.
6. Тіней та ефектів.
7. Згладжування країв та антиаліасінгу.

Математична модель анімації персонажа складається з:

1. Положення персонажа.
2. Обертання персонажа.
3. Масштабування персонажа.
4. Анімаційних кадрів.
5. Інтерполяції кадрів.
6. Фізичних обмежень.

2.3. Програмне забезпечення

У процесі розробки гри програмування відбувалося у системі Blueprint. Вона дає широкий функціонал для зручного програмування в Unreal Engine.

Переваги програмування за допомогою системи Blueprint:

1. Візуальне програмування.
2. Швидкість прототипування.
3. Низький поріг входу.
4. Відладка в реальному часі.
5. Можливість комбінування з кодом на C++.
6. Гнучкість на можливості розширення.

Використання системи Blueprint може значно полегшити розробку та зробити її доступнішою для широкого кола розробників, незалежно від їх рівня досвіду в програмуванні.

Розділ 3. Реалізація гри на Unreal Engine

Розробка гри на Unreal Engine є складним і захоплюючим процесом, який вимагає ретельного планування, технічних знань та креативності. Unreal Engine – це потужний рушій, який надає розробникам інструменти для створення високоякісних ігор з приголомшливою графікою та складною логікою. У цьому тексті ми детально розглянемо етапи реалізації гри на Unreal Engine, починаючи від концептуалізації та дизайну, до програмування, тестування та випуску гри.

Концептуалізація та дизайн

Ідея гри

Розробка гри починається з ідеї. Це може бути оригінальна концепція, натхненна іншими іграми, або новий підхід до існуючого жанру. На цьому етапі важливо чітко визначити основні механіки гри, сюжетну лінію, стилістику та цільову аудиторію. Концептуалізація також включає створення геймдизайнерського документа (GDD), який описує всі аспекти гри: від ігрової механіки до візуальних елементів та аудіо.

Прототипування

Після того як основна ідея та концепція гри визначені, наступним кроком є прототипування. Прототипи дозволяють перевірити основні механіки та ігровий процес на ранньому етапі. Це можуть бути прості моделі з мінімальною кількістю деталей, які дають уявлення про те, як гра буде виглядати та працювати. Unreal Engine надає інструменти для швидкого створення прототипів, що дозволяє розробникам експериментувати з різними ідеями та вибирати найкращі з них.

Розробка ігрового дизайну

Ігровий дизайн включає створення детальних планів рівнів, персонажів, об'єктів та інших елементів гри. На цьому етапі дизайнери працюють над створенням графічних концептів, 3D моделей, анімацій та текстур. Unreal Engine підтримує різні формати 3D моделей та текстур, що дозволяє інтегрувати ресурси з інших програм, таких як Blender або Maya.

Створення середовища та рівнів

Планування рівнів

Планування рівнів – це важливий етап, який включає визначення структури та логіки кожного рівня. Це включає розміщення об'єктів, визначення шляхів персонажів та створення ігрових ситуацій. На цьому етапі важливо враховувати як візуальні, так і геймплейні аспекти, щоб забезпечити цікавий та збалансований ігровий процес.

Створення ландшафту

Unreal Engine надає потужні інструменти для створення ландшафту, які дозволяють розробникам створювати великі та детальні ігрові світи. Інструменти для роботи з ландшафтом включають можливості для створення гір, долин, річок та інших природних елементів. Крім того, Unreal Engine підтримує різні методи текстуровання, що дозволяє створювати реалістичні поверхні.

Розміщення об'єктів та освітлення

Розміщення об'єктів на рівні є критичним етапом, який визначає, як гравці будуть взаємодіяти з середовищем. Unreal Engine надає різні інструменти для розміщення об'єктів та налаштування їх властивостей. Освітлення є ще одним важливим аспектом, який впливає на атмосферу гри та її візуальне сприйняття. Unreal Engine підтримує як динамічне, так і статичне освітлення, що дозволяє досягти різних ефектів.

Програмування та логіка гри

Використання Blueprint

Blueprint – це візуальна скриптова система в Unreal Engine, яка дозволяє створювати ігрову логіку без написання коду. За допомогою Blueprint можна створювати складні логічні схеми, які визначають поведінку об'єктів, персонажів та інших елементів гри. Це значно спрощує процес розробки, особливо для тих, хто не має досвіду програмування.

Програмування на C++

Хоча Blueprint є потужним інструментом, іноді виникає потреба в написанні коду для реалізації більш складних або оптимізованих функцій. Unreal Engine підтримує програмування на C++, що дозволяє розробникам мати повний контроль над ігровим процесом та ресурсами. Використання C++ може бути особливо корисним для створення складних систем, таких як штучний інтелект, фізика та мережеві функції.

Інтеграція системи штучного інтелекту

Штучний інтелект (ШІ) відіграє важливу роль у створенні захоплюючого ігрового досвіду. Unreal Engine надає різні інструменти для створення ШІ, включаючи Behavior Trees, які дозволяють моделювати складну поведінку персонажів. За допомогою цих інструментів можна створювати ворогів, союзників та інших NPC, які будуть взаємодіяти з гравцем відповідно до заданих правил.

Анімація та фізика

Анімація персонажів

Анімація є важливим аспектом, який впливає на реалістичність та захопливість гри. Unreal Engine підтримує різні методи анімації, включаючи скелетну анімацію, анімацію руху та морфінг. Використання системи анімаційних графів дозволяє створювати плавні переходи між різними анімаціями, забезпечуючи реалістичну поведінку персонажів.

Фізика та взаємодія об'єктів

Unreal Engine має вбудовану фізичну систему, яка дозволяє реалізувати реалістичну взаємодію об'єктів у грі. Це включає симуляцію зіткнень, гравітації, сил та інших фізичних явищ. Налаштування фізичних властивостей об'єктів дозволяє створювати різноманітні ефекти, від простих зіткнень до складних симуляцій, таких як руйнування будівель або поведінка рідин.

Звук та музика

Інтеграція звукових ефектів

Звукові ефекти грають ключову роль у створенні атмосфери гри та покращенні ігрового досвіду. Unreal Engine підтримує інтеграцію звукових файлів різних форматів та надає інструменти для їх відтворення та обробки. Це включає можливості для налаштування гучності, панорамування, реверберації та інших звукових параметрів.

Музичний супровід

Музичний супровід є важливою частиною ігрового досвіду, яка підсилює емоційний вплив гри. Розробники можуть інтегрувати музичні треки, які відповідають різним сценам та подіям у грі. Unreal Engine дозволяє створювати динамічний музичний супровід, який змінюється в залежності від дій гравця та ситуацій у грі.

Інтерфейс користувача

Створення HUD

Інтерфейс користувача (HUD) надає гравцям необхідну інформацію під час гри, таку як здоров'я персонажа, кількість боєприпасів, карта та інше. Unreal Engine має вбудовані інструменти для створення та налаштування HUD, які дозволяють розробникам створювати інтерактивні та інтуїтивні інтерфейси.

Меню та налаштування

Меню та налаштування є важливими елементами, які дозволяють гравцям взаємодіяти з грою, змінювати параметри та вибирати різні режими. Unreal Engine надає можливості для створення різноманітних меню, включаючи головне меню, меню налаштувань, меню паузи та інші.

Тестування та оптимізація

Функціональне тестування

Функціональне тестування забезпечує перевірку того, чи відповідає гра своїм технічним вимогам і чи працюють усі її елементи належним чином. Це включає перевірку механіки гри, поведінки персонажів, взаємодії об'єктів та правильності виконання всіх запланованих сценаріїв.

Тестування продуктивності

Тестування продуктивності визначає, наскільки ефективно гра використовує апаратні ресурси. Це включає оцінку частоти кадрів (FPS), часів завантаження, використання пам'яті та загальної стабільності гри під час різних сценаріїв. Unreal Engine надає інструменти для моніторингу продуктивності, такі як профайлер (Profiler).

Оптимізація

Оптимізація гри є важливим етапом, який забезпечує її плавну роботу на різних платформах та конфігураціях. Це включає зменшення використання ресурсів, оптимізацію коду та налаштування графічних параметрів. Unreal Engine надає різні інструменти для оптимізації, включаючи LOD-системи (Level of Detail), які дозволяють зменшувати деталізацію об'єктів на великих відстанях, та інструменти для зменшення використання пам'яті.

Реліз та підтримка

Підготовка до релізу

Підготовка до релізу включає завершення всіх етапів розробки, тестування та оптимізації гри. Це також включає створення маркетингових матеріалів, таких як трейлери, скріншоти та опис гри. Важливо також підготувати документацію та інструкції для гравців.

Випуск гри

Випуск гри може здійснюватися через різні платформи, такі як Steam, Epic Games Store, PlayStation, Xbox та інші. Unreal Engine підтримує експорт проектів для різних платформ, що дозволяє розробникам досягти ширшої аудиторії. Важливо також забезпечити належну підтримку гри після релізу, включаючи випуск оновлень та виправлення помилок.

Підтримка спільноти

Підтримка спільноти гравців є важливим аспектом після релізу гри. Це включає взаємодію з гравцями через форуми, соціальні мережі та інші канали зв'язку. Важливо враховувати зворотний зв'язок від гравців та оперативно реагувати на їхні запити та проблеми.

Розділ 4. Створення ігрових рівнів

Створення ігрових рівнів є однією з найважливіших складових процесу розробки ігор. Якісно розроблені рівні забезпечують захопливий ігровий процес, сприяють зануренню в ігровий світ та надають гравцям задоволення від гри. Процес створення ігрових рівнів включає кілька етапів, таких як концептуалізація, планування, розміщення об'єктів, налаштування освітлення та оптимізація. У цьому тексті ми детально розглянемо кожен з цих етапів, а також використання інструментів Unreal Engine для реалізації ігрових рівнів.

Концептуалізація ігрових рівнів

Ідея та концепція

Перший етап створення ігрових рівнів включає розробку ідеї та концепції рівня. Це включає визначення загальної тематики, стилю, атмосфери та головних цілей рівня. На цьому етапі важливо враховувати загальну концепцію гри, її сюжетну лінію та механіки, щоб рівень органічно вписувався в ігровий світ.

Створення геймдизайнерського документа

Геймдизайнерський документ (GDD) є важливим інструментом, який описує всі аспекти ігрового рівня. Він включає опис сценарію, механік, об'єктів, персонажів, ворогів та інших елементів рівня. GDD допомагає команді розробників мати єдине уявлення про рівень та координувати свої дії під час його створення.

Прототипування рівня

Прототипування є важливим етапом, який дозволяє перевірити основні ідеї та механіки рівня на ранньому етапі. Прототипи можуть бути простими моделями без деталізації, які дають уявлення про структуру рівня та його основні елементи. Unreal Engine надає інструменти для швидкого створення прототипів, що дозволяє експериментувати з різними ідеями та вибирати найкращі з них.

Планування ігрових рівнів

Структура рівня

Планування структури рівня включає визначення його основних частин, таких як стартова точка, ключові місця, точки інтересу, шлях гравця та кінцева точка. Важливо забезпечити логічну та цікаву послідовність подій, яка буде підтримувати інтерес гравця протягом всього рівня.

Баланс та складність

Баланс та складність рівня є критичними аспектами, які впливають на ігровий досвід. Важливо створити рівень, який буде викликати у гравця відчуття виклику, але при цьому не буде надто складним або надто легким. Це включає розміщення ворогів, пасток, головоломок та інших перешкод, а також надання гравцеві ресурсів та інструментів для їх подолання.

Сценарії та події

Сценарії та події додають динаміки та цікавості рівню. Це можуть бути сюжетні події, катсцени, скриптовані сцени або випадкові події, які відбуваються під час гри. Важливо інтегрувати ці елементи таким чином, щоб вони не порушували ігровий процес та органічно вписувалися в рівень.

Створення середовища рівня

Ландшафт та геометрія

Ландшафт та геометрія рівня визначають його загальну структуру та форму. Unreal Engine надає потужні інструменти для створення ландшафту, які дозволяють розробникам створювати великі та детальні ігрові світи. Це включає інструменти для моделювання рельєфу, розміщення гір, долин, річок та інших природних елементів.

Розміщення об'єктів

Розміщення об'єктів на рівні є важливим етапом, який визначає, як гравці будуть взаємодіяти з середовищем. Це включає розміщення будівель, дерев, скель, меблів та інших об'єктів, які створюють реалістичне та цікаве середовище. Важливо забезпечити логічне та естетично привабливе розміщення об'єктів, яке буде сприяти зануренню гравця у світ гри.

Текстури та матеріали

Текстури та матеріали визначають зовнішній вигляд об'єктів на рівні. Unreal Engine підтримує різні формати текстур та надає інструменти для їх налаштування. Це включає налаштування кольорів, відблисків, прозорості та інших властивостей матеріалів. Використання якісних текстур та матеріалів значно покращує візуальне сприйняття рівня.

Освітлення та ефекти

Налаштування освітлення

Освітлення є важливим аспектом, який впливає на атмосферу та візуальне сприйняття рівня. Unreal Engine підтримує як динамічне, так і статичне освітлення, що дозволяє досягти різних ефектів. Важливо налаштувати освітлення таким чином, щоб воно підкреслювало ключові елементи рівня та створювало бажану атмосферу.

Ефекти частинок

Ефекти частинок додають динаміки та реалізму рівню. Це можуть бути ефекти диму, вогню, води, пилу, вибухів та інших явищ. Unreal Engine надає інструменти для створення та налаштування ефектів частинок, які дозволяють розробникам додавати вражаючі візуальні ефекти до своїх рівнів.

Післяобробка

Післяобробка включає застосування різних фільтрів та ефектів до фінального зображення рівня. Це включає налаштування контрасту, яскравості, насиченості, глибини різкості, ефектів відблиску та інших параметрів. Використання післяобробки дозволяє значно покращити візуальне сприйняття рівня та створити бажану атмосферу.

Ігрова логіка та інтерактивність

Використання Blueprint

Blueprint – це візуальна скриптова система в Unreal Engine, яка дозволяє створювати ігрову логіку без написання коду. За допомогою Blueprint можна створювати складні логічні схеми, які визначають поведінку об'єктів, персонажів та інших елементів рівня. Це значно спрощує процес розробки, особливо для тих, хто не має досвіду програмування.

Програмування на C++

Хоча Blueprint є потужним інструментом, іноді виникає потреба в написанні коду для реалізації більш складних або оптимізованих функцій. Unreal Engine підтримує програмування на C++, що дозволяє розробникам мати повний контроль над ігровим процесом та ресурсами. Використання C++ може бути особливо корисним для створення складних систем, таких як штучний інтелект, фізика та мережеві функції.

Тригери та події

Тригери та події додають інтерактивності до рівня. Це можуть бути тригери, які активують певні події, коли гравець проходить через певну зону, або взаємодії з об'єктами, які запускають певні дії. Використання тригерів та подій дозволяє створювати динамічні та цікаві рівні, які реагують на дії гравця.

Анімація та фізика

Анімація об'єктів

Анімація об'єктів додає руху та динаміки до рівня. Це можуть бути анімації дверей, які відкриваються, механізмів, які рухаються, або інших об'єктів, які змінюють своє положення чи стан. Unreal Engine надає інструменти для створення та налаштування анімацій об'єктів, що дозволяє розробникам додавати реалістичні рухи до своїх рівнів.

Фізика та взаємодія об'єктів

Unreal Engine має вбудовану фізичну систему, яка дозволяє реалізувати реалістичну взаємодію об'єктів у грі. Це включає симуляцію зіткнень, гравітації, сил та інших фізичних явищ. Налаштування фізичних властивостей об'єктів дозволяє створювати різноманітні ефекти, від простих зіткнень до складних симуляцій, таких як руйнування будівель або поведінка рідин.

Тестування та оптимізація рівнів

Тестування геймплею

Тестування геймплею включає перевірку всіх механік та елементів рівня на предмет їхньої коректної роботи. Це включає перевірку поведінки персонажів, взаємодії об'єктів, тригерів та подій. Важливо переконатися, що всі елементи рівня працюють належним чином та відповідають запланованому ігровому процесу.

Оптимізація продуктивності

Оптимізація продуктивності рівня є важливим етапом, який забезпечує його плавну роботу на різних платформах та конфігураціях. Це включає зменшення використання ресурсів, оптимізацію коду та налаштування графічних параметрів. Unreal Engine надає різні інструменти для оптимізації, включаючи LOD-системи (Level of Detail), які дозволяють зменшувати деталізацію об'єктів на великих відстанях, та інструменти для зменшення використання пам'яті.

Тестування на різних платформах

Важливо тестувати рівні на різних платформах, на яких планується випуск гри. Це включає перевірку сумісності з різними операційними системами, консолями та мобільними пристроями. Unreal Engine підтримує експорт проектів для різних платформ, що дозволяє розробникам досягти ширшої аудиторії.

Підготовка до релізу та підтримка

Завершення та полірування рівнів

Завершення та полірування рівнів включає внесення останніх правок, виправлення помилок та покращення якості рівнів. Важливо переконатися, що всі елементи рівня відповідають високим стандартам якості та забезпечують найкращий ігровий досвід.

Розділ 5. Розробка персонажів та об'єктів

Розробка персонажів та об'єктів є ключовим етапом у створенні гри, який визначає не лише візуальну привабливість, але й функціональність та взаємодію усього ігрового світу. У цій частині дипломної роботи ми розглянемо процес створення персонажів та об'єктів у Unreal Engine, починаючи від концептуалізації і до остаточного впровадження у гру. Ми охопимо всі етапи, включаючи 3D-моделювання, текстурування, анімацію, інтеграцію в Unreal Engine, налаштування ігрової логіки та тестування.

Концептуалізація персонажів та об'єктів

Ідея та концепція

Концептуалізація є першим і, можливо, найважливішим етапом у створенні персонажів та об'єктів. На цьому етапі визначаються загальний стиль, тематика та функціональність. Розробка починається з ідеї, яка потім втілюється у концептуальні малюнки. Ці малюнки відображають зовнішній вигляд, деталі та пропорції майбутніх персонажів та об'єктів.

Створення концепт-арту

Концепт-арт є візуальним уособленням ідеї персонажа чи об'єкта. Це можуть бути як прості ескізи, так і детальні малюнки з різних ракурсів. Концепт-арт допомагає усій команді розробників мати єдине уявлення про зовнішній вигляд та стиль елементів гри. На основі концепт-арту художники та моделювальники створюють 3D-моделі.

Опрацювання деталей

Деталізація персонажів та об'єктів включає розробку всіх аспектів їх зовнішнього вигляду та функціональності. Це можуть бути елементи одягу, зброї, інтер'єру або аксесуари, які підкреслюють індивідуальність персонажів або реалізм об'єктів. На цьому етапі важливо враховувати загальну естетику гри та її стиль.

3D-моделювання

Створення базової моделі

3D-моделювання починається зі створення базової моделі персонажа або об'єкта. Для цього використовуються програми, такі як Blender, Maya або 3ds Max. Базова модель містить основні форми та пропорції без деталізації. Цей етап є важливим, оскільки правильні пропорції забезпечують реалістичність та природність майбутньої моделі.

Деталізація моделі

Після створення базової моделі додаються деталі, які надають моделі завершеного вигляду. Це можуть бути текстури, складки одягу, зморшки на шкірі, шви на тканині та інші дрібні деталі. Для додавання деталей використовуються такі інструменти, як ZBrush, які дозволяють створювати високополігональні моделі з великою кількістю деталей.

Топологія та оптимізація

Оптимізація моделі є важливим етапом, який забезпечує її ефективне використання у грі. Це включає створення правильної топології моделі, зменшення кількості полігонів та налаштування LOD (Level of Detail). Оптимізовані моделі забезпечують кращу продуктивність гри та зменшують навантаження на апаратне забезпечення.

Текстурування та матеріали

Створення текстур

Текстурування є процесом нанесення зображень на поверхню 3D-моделі, щоб надати їй кольору та деталізації. Для створення текстур використовуються програми, такі як Substance Painter або Photoshop. Текстури можуть включати кольорові мапи, мапи нормалей, мапи відблисків та інші. Важливо створити якісні текстури, які будуть відповідати загальному стилю гри.

Налаштування матеріалів

Матеріали визначають, як текстури будуть взаємодіяти зі світлом у грі. Unreal Engine надає потужний редактор матеріалів, який дозволяє налаштовувати властивості матеріалів, такі як відблиски, прозорість, металевість та інші. Використання правильно налаштованих матеріалів забезпечує реалістичний вигляд моделей у грі.

UV-розгортка

UV-розгортка є процесом проєкції 3D-моделі на 2D-площину для створення текстур. Це дозволяє розмістити текстури на поверхні моделі без спотворень. Правильна UV-розгортка є важливим етапом, який забезпечує якісне текстурування та відсутність артефактів на моделі.

Анімація персонажів

Створення ригу

Ригування є процесом створення скелету для 3D-моделі, який дозволяє її анімувати. Скелет складається з кісток та суглобів, які відповідають за рухи персонажа. Для створення ригу використовуються такі програми, як Maya або Blender. Правильний риг забезпечує реалістичні та природні рухи персонажа.

Анімація рухів

Після створення ригу додаються анімації, які визначають рухи персонажа. Це можуть бути анімації ходьби, бігу, стрибків, атак та інших дій. Анімації можуть створюватися вручну або за допомогою захоплення руху (motion capture). Unreal Engine підтримує обидва методи, що дозволяє створювати реалістичні та динамічні анімації.

Анімаційні графи

Анімаційні графи визначають логіку переходів між анімаціями. Це дозволяє забезпечити плавні переходи між різними станами персонажа, такими як перехід від бігу до стрибка або від атаки до захисту. Unreal Engine має потужну систему анімаційних графів, яка дозволяє налаштовувати складні анімаційні переходи.

Інтеграція в Unreal Engine

Імпорт моделей та текстур

Імпорт моделей та текстур у Unreal Engine є важливим етапом, який забезпечує їх використання у грі. Unreal Engine підтримує різні формати файлів, такі як FBX, OBJ, PNG та інші. Після імпорту моделі та текстури можуть бути налаштовані та використані у проєкті.

Налаштування скелетів та анімацій

Після імпорту моделей необхідно налаштувати скелети та анімації. Це включає прив'язку анімацій до скелету, налаштування анімаційних графів та створення анімаційних блупрінтів (animation blueprints). Це дозволяє забезпечити коректну роботу анімацій у грі.

Налаштування поведінки та логіки

Налаштування поведінки та логіки персонажів та об'єктів включає використання Blueprint або C++ для створення ігрової логіки. Це можуть бути скрипти, які визначають поведінку ворогів, інтелектуальну поведінку NPC, взаємодію об'єктів та інші елементи геймплею. Unreal Engine надає потужні інструменти для створення складної ігрової логіки без необхідності писати код.

Розділ 6. Тестування та виправлення помилок

Процес розробки гри на Unreal Engine включає в себе безліч етапів, одним з найважливіших серед яких є тестування та виправлення помилок. Тестування гри – це критично важливий аспект, що забезпечує її стабільність, продуктивність та відповідність задуму розробників. Виправлення помилок, у свою чергу, гарантує високу якість кінцевого продукту та задоволення користувачів. У цьому тексті детально розглянемо основні методи тестування, інструменти, які використовуються для цього в Unreal Engine, а також підходи до виявлення та виправлення помилок.

Основи тестування гри

Тестування гри можна поділити на кілька основних типів: функціональне тестування, тестування продуктивності, тестування зручності використання та тестування на сумісність. Кожен з цих типів має свої цілі та методи.

Функціональне тестування

Функціональне тестування спрямоване на перевірку того, чи відповідає гра своїм технічним вимогам і чи працюють усі її елементи належним чином. Це включає перевірку механіки гри, поведінки персонажів, взаємодії об'єктів та правильності виконання всіх запланованих сценаріїв.

В Unreal Engine функціональне тестування може здійснюватися за допомогою системи автоматизованих тестів, що підтримує платформа. Ці тести дозволяють перевіряти різні аспекти гри на кожному етапі розробки, мінімізуючи ризик виникнення помилок на пізніх стадіях.

Тестування продуктивності

Тестування продуктивності визначає, наскільки ефективно гра використовує апаратні ресурси. Це включає оцінку частоти кадрів (FPS), часів завантаження, використання пам'яті та загальної стабільності гри під час різних сценаріїв.

Unreal Engine надає потужні інструменти для моніторингу продуктивності, такі як профайлер (Profiler), який дозволяє відслідковувати використання ресурсів у реальному часі та ідентифікувати вузькі місця в продуктивності.

Тестування зручності використання

Тестування зручності використання (usability testing) фокусується на взаємодії користувача з грою. Це включає перевірку інтуїтивності інтерфейсу, зручності керування та загальної привабливості гри для цільової аудиторії.

Проведення цього типу тестування часто вимагає залучення реальних гравців, які можуть надати зворотний зв'язок щодо свого досвіду. Такий підхід дозволяє розробникам вносити необхідні корективи для поліпшення загального користувацького досвіду.

Тестування на сумісність

Тестування на сумісність гарантує, що гра коректно працює на різних платформах і конфігураціях апаратного забезпечення. Це особливо важливо для багатоплатформних ігор, де різні системи можуть мати різні характеристики та обмеження.

Unreal Engine дозволяє тестувати гру на різних платформах, таких як Windows, macOS, консолі та мобільні пристрої, що забезпечує її стабільність і якість на кожній з них.

Інструменти для тестування в Unreal Engine

Unreal Engine пропонує безліч інструментів та функцій для ефективного тестування гри. Деякі з них заслуговують особливої уваги через їхню функціональність та зручність використання.

Automation Framework

Automation Framework – це потужний інструмент для автоматизованого тестування, що дозволяє створювати та виконувати тести для різних аспектів гри. Він підтримує функціональні, регресійні та стресові тести, забезпечуючи всебічну перевірку проекту.

За допомогою Automation Framework можна створювати сценарії тестування, які будуть виконуватися автоматично, що значно прискорює процес виявлення помилок та підвищує ефективність тестування.

Profiler

Profiler – це інструмент для моніторингу продуктивності гри в реальному часі. Він надає детальну інформацію про використання CPU, GPU, пам'яті та інших ресурсів, дозволяючи розробникам ідентифікувати та вирішувати проблеми з продуктивністю.

Debugger

Debugger в Unreal Engine дозволяє розробникам відслідковувати виконання коду в реальному часі, знаходити та виправляти помилки. Він підтримує різні типи точок зупинки (breakpoints), спостереження за значеннями змінних та інші корисні функції для діагностики проблем.

Visual Logger

Visual Logger – це інструмент, що допомагає візуалізувати події та дії, які відбуваються під час виконання гри. Він дозволяє розробникам бачити, як змінюються стани об'єктів та параметри гри в реальному часі, що полегшує виявлення логічних помилок.

Виявлення та виправлення помилок

Виправлення помилок є важливою частиною процесу розробки гри. Цей етап включає в себе декілька ключових кроків: виявлення помилок, їх аналіз, виправлення та повторне тестування.

Виявлення помилок

Виявлення помилок може здійснюватися різними способами, включаючи автоматизовані тести, мануальне тестування та отримання зворотного зв'язку від користувачів. Важливо мати добре організовану систему для відслідковування знайдених помилок, таку як система відстеження багів (bug tracking system).

Аналіз помилок

Аналіз помилок включає розуміння причин їх виникнення та визначення найбільш ефективних способів їх виправлення. Використання інструментів, таких як Debugger та Visual Logger, може значно спростити цей процес.

Виправлення помилок

Виправлення помилок передбачає внесення змін у код, налаштування гри або її ресурсів. Важливо впевнитися, що виправлення не призведуть до появи нових проблем або конфліктів.

Повторне тестування

Після виправлення помилки необхідно провести повторне тестування, щоб переконатися, що проблема дійсно вирішена та не вплинула негативно на інші аспекти гри. Цей процес може включати виконання автоматизованих тестів, мануальне тестування або використання профайлерів для оцінки продуктивності.

Системи відстеження багів

Використання систем відстеження багів, таких як JIRA, Trello або Bugzilla, дозволяє організовано зберігати інформацію про помилки, їх пріоритети, статуси та відповідальних за їх виправлення. Це забезпечує ефективний менеджмент процесу тестування та виправлення помилок.

Розділ 7. Посібник для гравця

Для запуску гри необхідно зробити подвійний клік на виконавчий файл, що має ім'я «Diplom.uproject» (рис. 1).

Ось так виглядає кореневий каталог з виконавчим файлом:

Имя	Дата изменения	Тип	Размер
Build	15.05.2024 17:53	Папка с файлами	
Config	15.05.2024 17:53	Папка с файлами	
Content	15.05.2024 19:47	Папка с файлами	
DerivedDataCache	15.05.2024 19:41	Папка с файлами	
Intermediate	15.05.2024 19:57	Папка с файлами	
Saved	15.05.2024 19:52	Папка с файлами	
Script	15.05.2024 19:47	Папка с файлами	
Diplom.uproject	15.05.2024 17:53	Unreal Engine Proj...	1 КБ
SylizedProvencal.png	15.05.2024 17:53	Файл "PNG"	72 КБ

Рисунок 1. Кореневий каталог

Гра розроблялася та тестувалася на комп'ютері з такими характеристиками:

- Операційна система Windows 10;
- Відеокарта GeForce GTX 2060 12 gb;
- Процесор AMD Ryzen 5 3600;
- 16 gb оперативної пам'яті;
- Диск SSD Aрасer AS340 480GB;
- Блок живлення Chieftec 650W;

Також на комп'ютері було встановлено таке програмне забезпечення для роботи сучасних ігор:

- DirectX;
- Microsoft net;
- Microsoft Visual C++;

Кнопки керування у грі:

- W – вперед;
- A – вліво;
- D – вправо;
- S – назад;
- Space – стрибати;
- V – змінити камеру;
- E – взаємодіяти;
- F – підібрати;
- P – головне меню;
- LShift – бігати;
- Ctrl – сісти;
- F11 – повний екран;
- F9 – зберегти;
- F12 – загрузити;
- Tab – прицілитися;
- СКМ – атакувати магією;
- ЛКМ – атакувати мечем;
- ПКМ – блокувати;
- G – відштовхнути;
- Q – дістати зброю;

Висновок

Під час написання дипломної роботи було покращено практичні навички роботи в Unreal Engine, програмування в Blueprint, створення та налаштування ігрових локацій, тестування ігор, виявлення різних помилок та їх виправлення.

Також були розвинуті навички пошуку рішення проблем у процесі розробки ігор за допомогою мережі Інтернет та навички розподілу робочого часу, планування роботи і доведення проектів до кінця.

Дипломна робота стане основою цінного досвіду, який буде поповнюватися майбутніми проектами та буде допомагати у роботі в цій професії.

Перелік посилань

1. Epic Games //Режим доступу: <https://store.epicgames.com/en-US/>
[переглянуто 04.05.2024]
2. Unreal engine // Режим доступу: <https://www.unrealengine.com/en-US>
[переглянуто 06.05.2024]
3. Aram Cookson — «Unreal Engine Game Development in 24 Hours» / Aram Cookson США, 2019 р.
4. Satheesh PV. — «Unreal Engine Game Development Essentials» / Satheesh PV. США, 2020 р.
5. Joanna Lee — «Learning Unreal Engine Game Development» / Joanna Lee США, 2020 р.
6. Peter L. Newton. — «Unreal Engine AI Programming Essentials» / Peter L. Newton. США, 2019 р.
7. Katax Emperore — «Unreal Engine Physics Essentials» / Katax Emperore США, 2020 р.
8. Lauren S. Ferro — «Unreal Engine Blueprints Visual Scripting Projects» / Lauren S. Ferro США, 2020 р.
9. Simon Goodwin — «Mastering Unreal Engine 5.X» / Simon Goodwin США, 2020 р.

Додатки
Скріншоти з гри

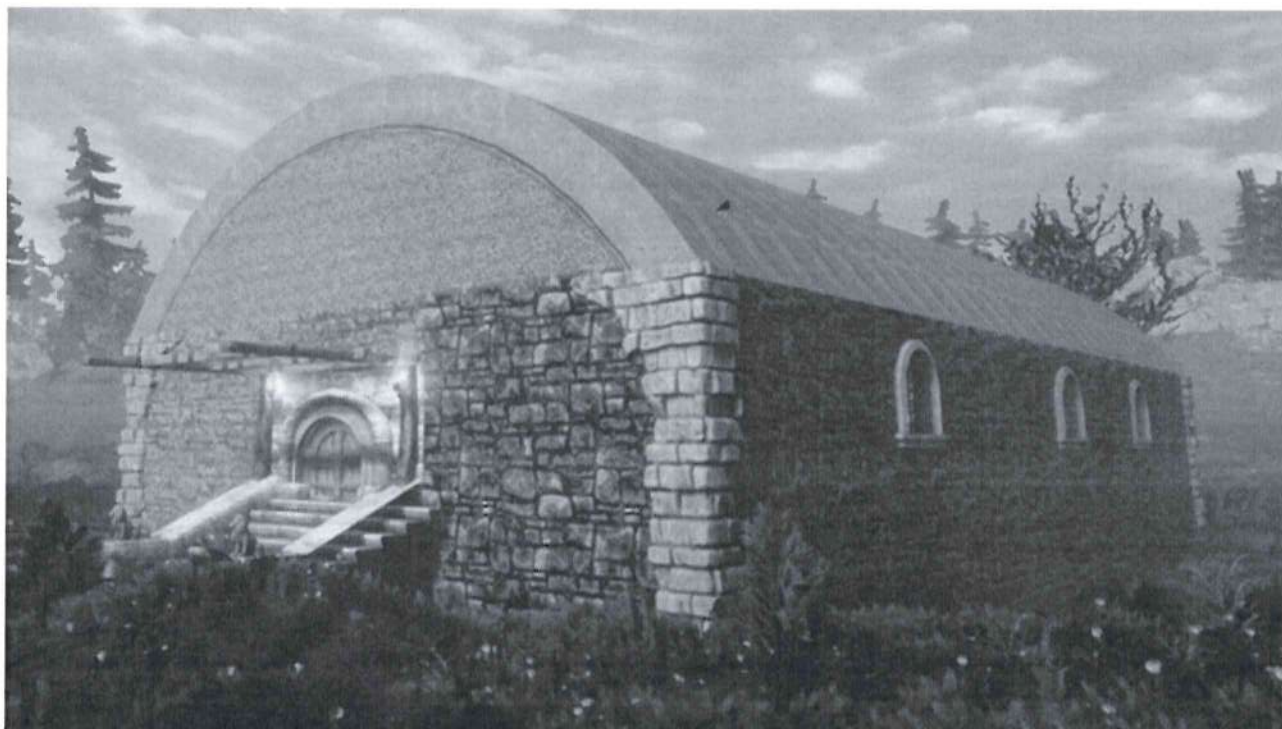


Рисунок 2. Стартова локація