МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук

Кафедра інформаційних технологій

КОМПЛЕКСНА ДИПЛОМНА РОБОТА

на тему: «Ігровий додаток з використанням інтегрованої середи Unity 3D»

Склад:

Частина 1. «Розробка сцен гри»

Виконавець: Димитрук М.С.
(П.І.Б.)

Керівник: Терещенко Т.М.
(П.І.Б.)

Частина 2. «Розробка скриптів гри»

Виконавець: Нєцик Р.І.
(П.І.Б.)

Керівник: Терещенко Т.М.
(П.І.Б.)

Староста роботи: Нєцик Р.І.
(П.І.Б.)

Провідний керівник проекту: к.т.н., доц. Терещенко Т.М.
(П.І.Б.)

Рецензент: к.т.н., доц., Худенко В.П.
(П.І.Б.)

ОДЕСА 2017

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук____

Кафедра інформаційних технологій

ДИПЛОМНА РОБОТА

Рівень вищої освіти бакалавр

на тему: Розробка сцен гри

Виконав студент 4 курсу групи К-45
Напрям підготовки 6.050101
комп'ютерні науки
Димитрук Максим Сергійович
_____

Керівник к.т.н., доц., _____
Терещенко Тетяна Михайлівна

Консультант _____
_____

Рецензент к.т.н.,доц.., _____
Худенко Валентина Петрівна

Одеса 2017

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

AAA - Високобюджетні і дорогі проекти

3D - Тривимірна графіка

UI - Користувальницький інтерфейс

EM - Режим редагування

OM - Режим об'єкта

LOD - Level Of Detail - Створення декількох варіантів одного об'єкта з різними ступенями деталізації

FBX - Один із форматів Unity

ВСТУП

Ігровий движок - центральний програмний компонент комп'ютерних та відеоігор або інших інтерактивних додатків з графікою, оброблюваної в реальному часі. Він забезпечує основні технології, спрощує розробку і часто дає грі можливість запускатися на декількох платформах, таких як ігрові консолі та настільні операційні системи, наприклад, GNU / Linux, Mac OS X і Microsoft Windows.

Однак, незважаючи на велику кількість багатофункціональних інструментів, від розробника все одно потрібна дуже велика кількість праці, щоб створити дійсно цікавий продукт. Однозначним плюсом розробки комп'ютерної гри на вже готовому інструменті (движку), є швидкість розробки програмної частини. Якщо раніше програмісту потрібно було писати безліч рядків коду, щоб використовувати просту можливість перевірки зіткнення між двома об'єктами, то тепер всі ці обчислення робляться за все однією командою. Таким чином, ігрові движки сильно спростили життя програмістам. Однак, програмування це тільки одна частина розробки комп'ютерної гри. Створення якісного ігрового продукту буде так само включати в себе створення великого обсягу графічного матеріалу. І до всього іншого, буде потрібно зробити якісне озвучення гри.

Розробкою відеоігор може займатися як одна людина, так і фірма. Розробка AAA проекту (вищий клас) коштує від мільйона доларів і більше. Процес розробки звичайної сучасної гри займає близько року, для AAA-проектів може затягнутися до 2-3 років. Цикл розробки звичайних «казуальних» ігор займає близько 4-6 місяців.

Основною метою дипломного проекту є розробити survival horror гру за допомогою ігрового движка Unity 3D, але так як даний ігровий движок є лише конструктором, то доведеться так само використовувати вільний професійний пакет для створення тривимірної комп'ютерної графіки "Blender" він включає в себе засоби моделювання, анімації, рендеринга і постобробки.

Основні задачі які потрібно виконати у ході виконання дипломного проекту:

- Ознайомитись з ігровим движком Unity 3D;
- Ознайомитись з графічним редактором Blender;
- Навчитися создавати 3D объекти для нашого проекта у стилі жахів;
- Навчитися экспортувати объекти із графічного редактора Blender у Unity 3D.

- Вивчити весь функціонал Unity 3D для правильного функціонування об'єктів.
- Додати до проекту эксклюзивний функціонал Unity 3D який сприяє збільшенню FPS.
- Скомпелювати проект на OS Windows для запуску.

1 АНАЛІЗ РИНКУ КОМП'ЮТЕРНИХ ІГОР

1.1 Перспективні напрямки розробки

В еру високошвидкісного інтернету велика частина ігрової індустрії, так чи інакше, орієнтована на онлайн. На даний момент існує безліч різновидів багатокористувацьких ігор. Залежно від параметра класифікації, дані гри можна розділити на:

- Браузерні і клієнтські гри. У браузерних іграх web-браузер використовується як операційна оболонка, або виступає в ролі контейнера для віртуальної машини, яка виконує код гри.
- Ігри, побудовані на принципі пошуку партнерів та інтерактивні світи (або масові ігри для декількох гравців). До першої групи належать гри за типом шахів, шашок, морського бою і їм подібних і відрізняються простотою в порівнянні з другою.

На даний момент найбільш поширеними типами є інтерактивні світи і гри для соціальних мереж.

Крім технологій реалізації багатокористувацьких ігор, продукти даного сегмента ринку добре помітні за жанрами:

- MMORPG (Massively Multiplayer Online Role-Playing Game, масова розрахована на багато користувачів ролева онлайн-гра)
- MMOFPS (Massively Multiplayer Online First Person Shooter, масовий багатокористувацький онлайн-shooter від першої особи)
- MMORTS (Massively Multiplayer Online Real-time Strategy, масова розрахована на багато користувачів онлайн-стратегія в реальному часі)
- Онлайн-ігри в жанрі time-managment (тайм-менеджмент, відомий також під назвою «принеси-подай»)
- Інші - різні комбінації вищеописаних жанрів, а також менш поширені, «екзотичні» жанри.

Величезною популярністю серед гравців користується жанр Survival Horror. жанр комп'ютерних ігор, характерним для якого є упор на виживання ігрового персонажа і нагнітання, подібно літературі жахів і фільмів жахів, атмосфери страху і тревоги. Як правило, ігри жанру survival horror пропонують гравцеві пробиратися через лякаючі лабіринтоподібні віртуальні світи, розшукуючи способи просунутися далі і піддаючись несподіваним нападам ворогів або іншим страшним небезпекам.

Однією з унікальних рис жанру survival horror є те, що він визначається в більшій мірі загальною атмосферою гри, ніж ігровий механікою. Хоча

survival horror зазвичай асоціюється з геймплеєм Resident Evil і Silent Hill, існує досить представників жанру, що нагадують побий їх усіх, квести, рольові ігри та шутери від першої особи.

Всі ігри цього жанру об'єднують теми страху, непояснених явищ, жорстокості і тому подібного.

На відміну від ігор жанру action-adventure, до яких гри survival horror близькі, в survival horror гравець не може повністю озброїти ігрового персонажа або достатнім чином підготувати його до майбутніх битв. Гравець може стикатися з великою кількістю ворогів, при цьому кількість боєприпасів для зброї помітно менше, ніж в інших іграх, а потужна зброя на зразок ракетних установок є рідкісним або зовсім відсутня в грі. В силу цього ігровий персонаж більш вразливий, ніж в інших іграх, і перевага завжди знаходиться на стороні ворожого оточення. Таким чином, в survival horror бої відступають на другий план, а гравець повинен навчитися уникати ворогів або використовувати проти них елементи оточення.

Сюжет типової гри в жанрі survival horror включає в себе розслідування деяких таємниць і зіткнення з страшними силами, таким чином, ігри можуть використовувати теми і образи з фільмів і літератури жахів, перетворюючи їх в перешкоди для гравця. Ранні гри жанру використовували характерні для фільмів жахів прийоми, пов'язані з положенням камери - завдяки цьому вороги могли перебувати поза увагою гравця. Багато ігор цього жанру використовують закадрову музику і звуки, попереджаючи гравця про небезпеку. Такі звукові підказки допомагають гравцеві в просуванні по грі, але також підстьобують відчуття тривоги і невизначеності.

1.2 Існуючі survival horror-ігри

1.2.1 Опис The Forest

The Forest (з англ. - «Ліс») - комп'ютерна гра, в жанрі survival horror з відкритим світом і видом від першої особи, що розробляється компанією Endnight Games на движку Unity 5. Рання альфа-версія гри була випущена в Steam 30 травня 2014 року.

Гра присвячена виживанню на населеному острові: ігровий персонаж, який потрапив на острів в результаті авіакатастрофи, повинен знаходити собі їжу і дах, а також оборонятися від страшних тубільців-канібалів. Розробники

поставили перед собою завдання зробити незвичайну гру, не схожу на інші ігри в жанрі survival horror, такі як Resident Evil і Silent Hill.

Рисунок 1.1 - Скріншот аборигена із гри The Forest

Таблиця 1.1 - Оцінка проекта The Forest різними видавниками

Ігрове видання	Оцінка
Electronic Games	92 із 100
Megagame	95 із 100
PC-Gamer	89 із 100
Оцінка користувачів	8.7 із 10

Гра отримала наступні нагороди: "Golden Joystick Awards 2015" у категоріях: "Найкраща оригінальна гра", "Найкраща Графіка", "Гра року".

1.2.2 Опис Alien:Isolation

Alien: Isolation (з англ. - «Чужий: Ізоляція») - комп'ютерна гра в жанрі Survival Horror з елементами стелса, розроблена компанією «Creative Assembly» і видана компанією «Sega». Вихід гри відбувся 7 жовтня 2014 року.

Гра була розроблена з великим орієнтиром на фільм 1979 року "Чужий» (в грі дуже багато атрибутики з фільму), аж до того, що весь антураж фантастичного майбутнього зображений так, як він представлявся в 1970-х. Гра трохи йде врозріз з більше орієнтованим на екшен фільмом 1986 року «Чужі» і є першою комп'ютерною грою в серії Чужого, геймплей якої побудований на страху виживання.

Рисунок 1.2 - Скриншот із гри Aliens:Isolation

Таблиця 1.2 - Оцінка проекта Aliens:Isolation різними видавниками

Ігрове видання	Регіон	Оцінка

PC-Gamer	Міжнародний	93 із 100
New Statesman	Великобританія	96 із 100
Kotaku Australia	Австралія	8.7 із 10
Оцінка користувачів		8.9 із 10
Igromania	Міжнародний	8.6 із 10

Гра отримала безліч нагород: "Премія за найкращий звук", "Краща гра", "Краща британська гра", "Кращий геймдізайн", "Найбільш інноваційна гра".

1.2.3 Опис SOMA

SOMA - комп'ютерна гра в жанрі Survival Horror, розроблена студією Frictional Games, відомої своїми серіями ігор Penumbra і Amnesia. Дія гри розгортається на підводному дистанційному дослідному об'єкті PATHOS-2, де машини починають набувати людські риси. Анонс гри відбувся 11 вересня 2013 року. 22 вересня 2015 року гра надійшла в продаж.

SOMA перебувала в розробці з 2010 року. Це перша гра на движку HPL3 Engine. У квітні 2014 року стало відомо, що сеттинг гри буде підводним. У березні 2015 року розробники оголосили про те, що розробка гри майже завершена. 29 травня з'явився перший геймплейний трейлер гри і була оголошена дата релізу - 22 вересня 2015 року. У квітні 2015 року відбулося закрите бета-тестування. 24 серпня став доступний передзамовлення цифрової копії гри в магазинах цифрової дистрибуції Steam і GOG.com. 16 вересня гра стала доступна для предзагрузкі в цифровому сервісі Steam.

За перші шість місяців продажів було реалізовано 250 тисяч копій гри. Згідно зі статистикою сервісу SteamSpy, на 5 січня 2017 року продано 372 тисячі копій.

Рисунок 1.3 - Скриншот із гри SOMA

Таблиця 1.3 - Оцінка проекта SOMA різними видавниками

Ігрове видання	Оцінка
Metacritics	83 із 100
GameRankings	84,35 із 100
PC-Gamer	86 із 100
Оцінка користувачів	7.9 із 10

1.2.4 Опис The Evil Within

The Evil Within (рус. Зло всередині), відома в Японії як Psychobreak - комп'ютерна гра в жанрах survival horror і шутера від третьої особи, розроблена компанією Tango Gameworks і випущена компанією Bethesda Softworks для платформ Microsoft Windows, PlayStation 3, Xbox 360, PlayStation 4 і Xbox One в 2014 році. Гра була розроблена під керівництвом Шінджі Миками, творця серії Resident Evil. Вона не має сюжетної зв'язку з Resident Evil, але включає в себе ряд схожих тем, образів і елементів.

Рисунок 1.4 - Скриншот із гри The Evil Within

Гра отримала змішані відгуки ігрової преси: критики високо оцінили хоррор-складову гри і дизайн рівнів, але відзначили її вторинність по відношенню до більш старих ігор жанру і численні технічні проблеми. Будучи грою, орієнтованої на західну аудиторію, The Evil Within отримала комерційний успіх в США і Європі, але помітно менший на своїй батьківщині - в Японії. Сюжет гри був продовжений в трьох доповненнях.

2 ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ РОЗРОБКИ

2.1 Порівняльна характеристика ігрових движків

Unity - найпопулярніший движок для створення 2D- і 3D-ігор. Безперечно, він став лідером індустрії, і, як тільки з'являється нова ігрова, або графічна технологія, розробники негайно реалізують її в Unity. Движок Unity особливо цінний за низький поріг входження для початківців користувачів, завдяки цьому, а також тому, що інді-версія безкоштовна, навколо движка організувалося величезне співтовариство. Низький поріг входження є результатом грамотного дизайну програми, багато речей можна виконати за допомогою різних редакторів, не написавши при цьому ні строчки коду.

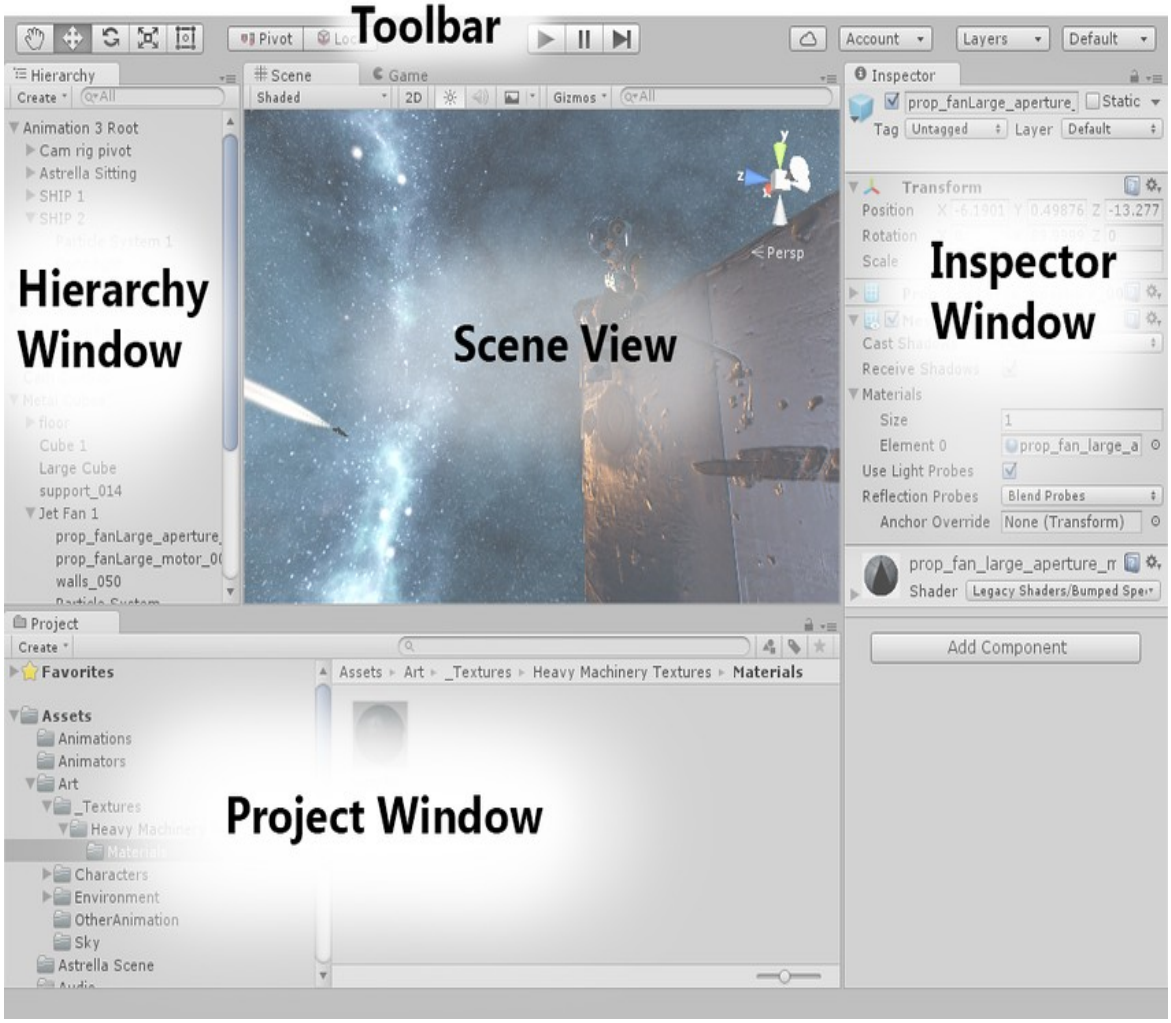

Рисунок 2.1 - Основний інтерфейс Unity 3D

Таблиця 2.1 - Переваги і недоліки Unity 3D

Недоліки	Переваги
Є недоліки для різних платформ	Безкоштовний для Indie-розробників
Рендер має нарікання	Багатоплатформовий
Немає креслень	Простий в освоєнні
Немає вихідного коду	Багатий магазин активів
	Швидка компіляція
	Багато документації та уроків

Torque 2D / 3D, був свого часу лідером, але під натиском Unity втратив свої позиції. Проте до цих пір на ньому розробляється безліч успішних проектів, оскільки він активно розвивається спільнотою. Відмінності між двовимірної і тривимірної версіями досить значні, але є і загальні елементи, наприклад розвинена мережева підсистема. Після виходу в світ open source T3D зберіг і навіть збільшив свої можливості, а T2D, навпаки, багато втратив. Наприклад, він втратив абсолютно всі вбудовані редактори, які, очевидно, були вилучені за певних юридичних угод. Проте на ньому можна розробляти ігри для трьох платформ: Windows, OS X і, що найцікавіше, iOS (і продавати ігри в App Store, не відраховуючи ні копійки авторам движка). Даний движок - це одна кодова база на C ++ без додаткових експортерів. Як видно, фундаментальні відмінності 2D і 3D - версій полягають в графічної підсистемі: T2D для візуалізації використовує OpenGL, а T3D - Direct.

В якості скриптової мови в T2D, як і в T3D, використовується Torque Script. Разом з тим в T2D для опису ігрових елементів служить XML-подібна мова TAML. Вона дозволяє визначити властивості об'єктів на стадії ініціалізації рівня гри. Для відтворення звуків T2D використовує бібліотеку OpenAL. Симуляція фізики здійснюється за допомогою движка Box2D, що став стандартом в двовимірних фізичних обчисленнях. Незважаючи на те що в двовимірному TORK ще поки немає конструктора GUI, за допомогою засобів движка (в скриптовому коді) можна створювати призначений для користувача інтерфейс звичними компонентами, а не простими спрайтами.

Однак, якщо необхідний компонент відсутній, його можна створити на основі спрайтів. Маючи аналогічну з 3D-версією мережеву систему, на T2D можна розробляти мультиплеєрні ігри, які набирають популярність, - наприклад P2P з планшетів.

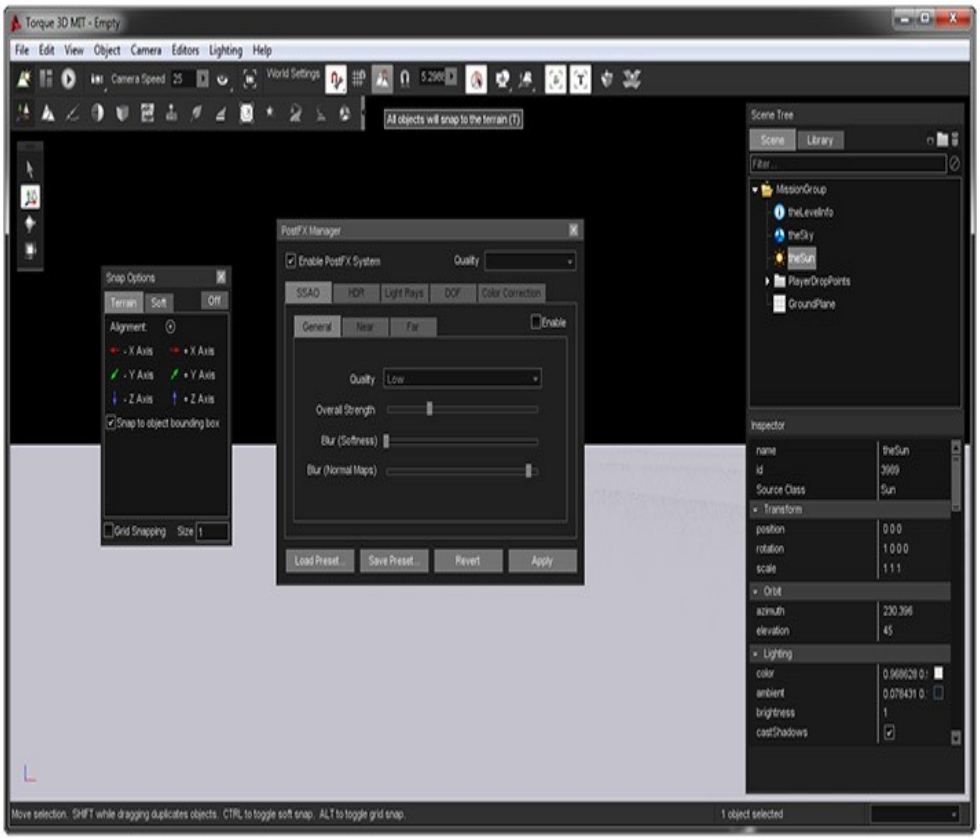

Рисунок 2.2 - Основний інтерфейс Torque 3D

Таблиця 2.2 - Переваги і недоліки Torque 3D

Недоліки	Переваги
Немає якісного редактора рівнів	Якісний
Повільна компіляція	Багатоплатформовий
Високі вимоги до заліза	Легкий в освоєнні
Немає вихідного коду	Безкоштовний

CryEngine 3 бере початок своєї історії в 2001 році, коли була анонсована перша розроблена на ньому гра Far Cry. З тих пір минуло багато часу, і поточна на даний момент п'ята - остання версія була випущена в жовтні 2016-го. Розробники цього движка з самого початку мали на меті не самим створювати на ньому ігри, а продавати його як технологію. Отже, всі розробленні Crytek'ом ігрові програми - це з метою зробити додаткову рекламу своєму головному продукту. Хоча для вивчення він доступний безкоштовно, щоб розробляти на ньому комерційні проекти, необхідно заплатити, при чому ціна публічно не оголошується. В результаті ліцензіат отримує движок, документацію (навчальні матеріали), вихідний код, а також оперативну підтримку. Крім того, процес ліцензування движка таїть в собі безліч підводних каменів - хоча б те, що ліцензувати його може тільки юридична особа, яка має надати дані про розроблені продукти і в окремих випадках про всіх своїх співробітників.

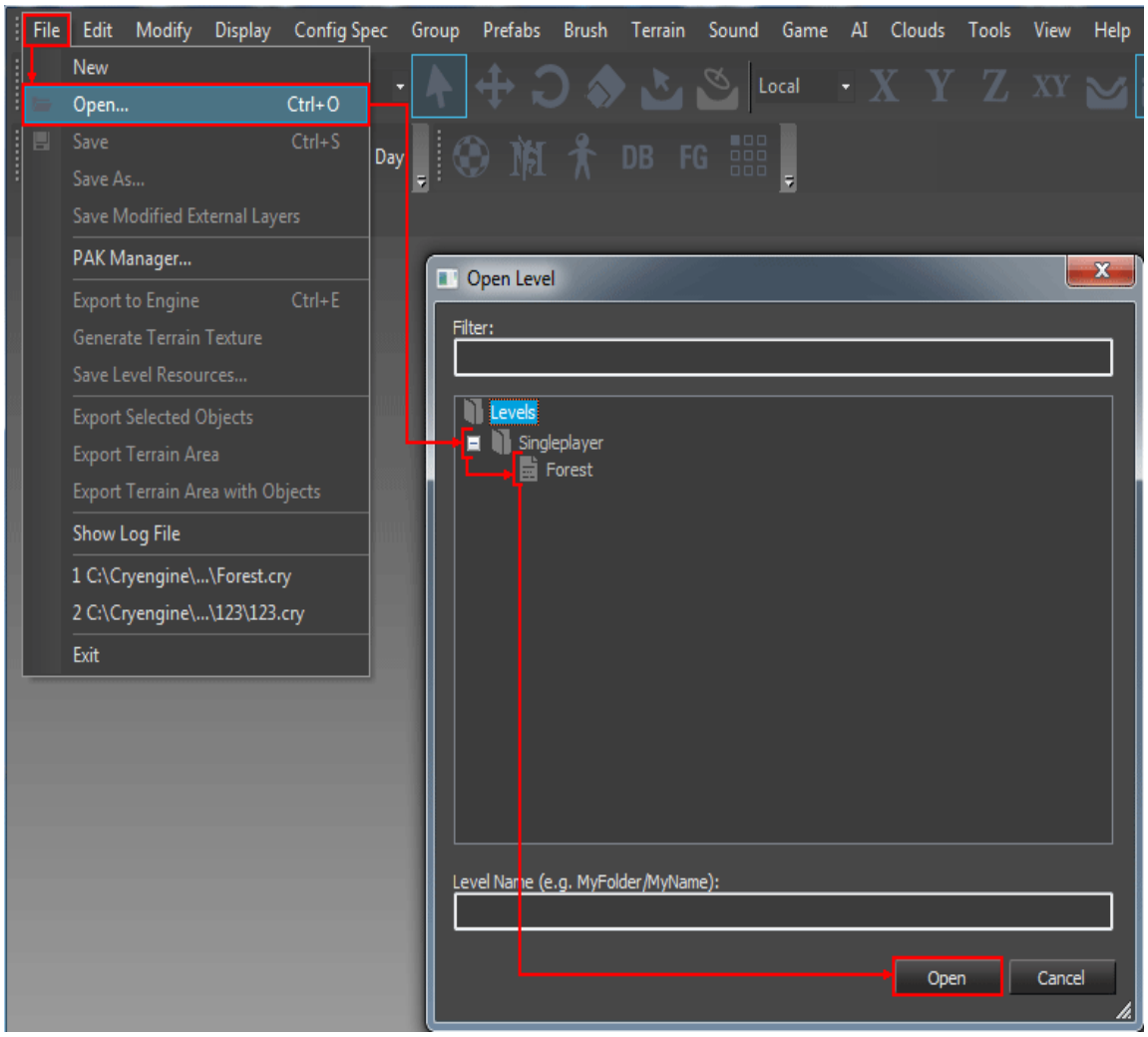

Рисунок 2.3 - Основний інтерфейс CryEngine 3

Таблиця 2.3 - Переваги і недоліки CryEngine 3

Недоліки	Переваги
Не безкоштовний	Якісний
Повільна компіляція	Багато документації та уроків
Високі вимоги до заліза	Легкий в освоєнні
Не багатоплатформовий	
Неможливо продавати свої ассети та скріпти	

На відміну від попередніх движків лінійки (які були виключно PC-орієнтованими), CryEngine 3 орієнтований на створення крос-платформних ігор, призначених для PC і консолей. В даний час підтримуються платформи Xbox 360, Xbox One, PlayStation 3-4, WiiU, а також технології візуалізації настільної Windows - DirectX 9-11. Як можна помітити, підтримки мобільних платформ немає. У ньому спочатку присутня підтримка глобальних мультіплеєрних (MMO) ігор. CryEngine 3 володіє приголомшливим списком технологій візуалізації, ось деякі з них: динамічне освітлення і затінення в реальному часі, затуманення, карти нормалей і паралакс-маппінг, підповерхневе розсіювання, світлові промені і хвилі, управління рівнем деталізації ландшафту, а також багато іншого. Фізичний компонент движка CryPhysics також працює незалежно від фізичних API, таких як PhysX. Вбудована система анімації пропонує кілька відмінних підсистем: індивідуалізація персонажів, параметрична скелетна анімація, процедурне деформування руху. Також заслуговує на окрему увагу вбудована система AA, яка дозволяє обробляти поведінку не тільки персонажів, але і транспортних засобів. Вона складається з трьох модулів: розумні об'єкти, алгоритми динамічного виявлення шляху, а також система, керована сценаріями.

UDK - це безкоштовна версія движка Unreal Engine 3, що володіє всім успадкованим інструментарієм останнього для створення ігрових світів. Список підтримуваних платформ не такий широкий, як у Unity, але цього цілком вистачає, щоб окупити розробку: Windows PC, Windows Store, OS X,

iOS, Android і консолі передостаннього покоління. Для скриптинга в движку використовується власна мова - UnrealScript. На сайті розробників представлено багато навчальних матеріалів, як текстових, так і відео, як по редактору, так і по скриптингу. UE3 отримав безліч нагород на індустріальних заходах, а також в кінематографі і не раз ставав кращим ігровим / графічним движком року. Можна сказати, що UDK відрізняється від UE3 тільки відсутністю вихідного коду.

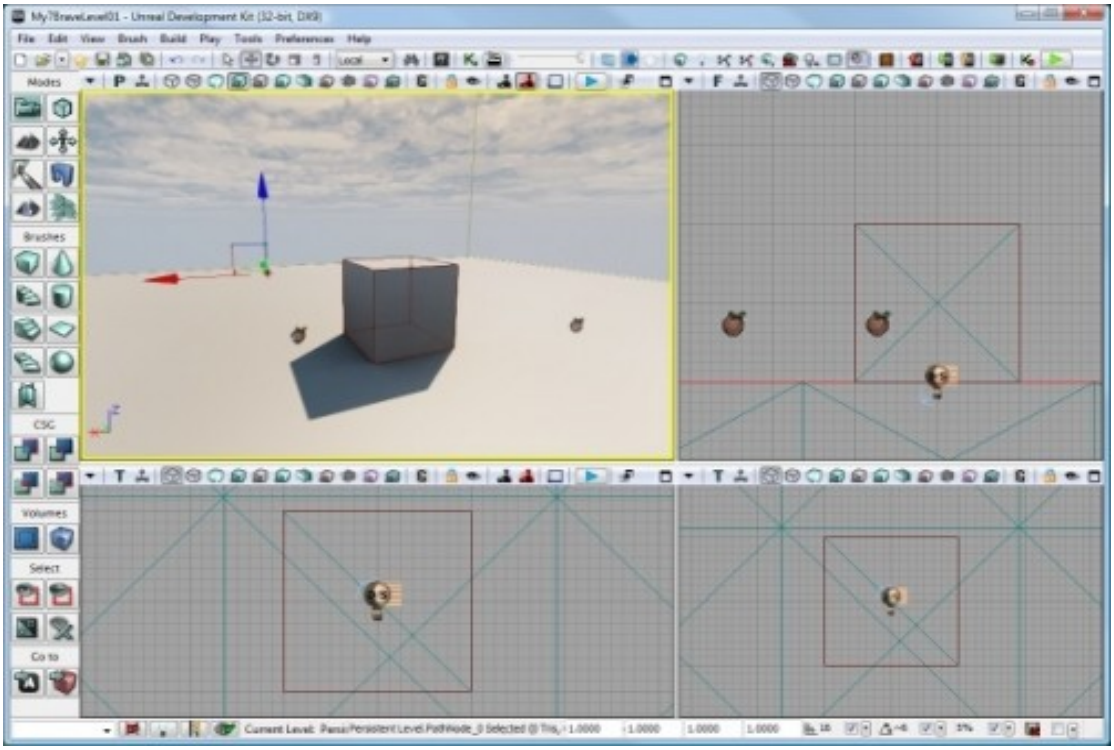

Рисунок 2.4 - Основний інтерфейс UDK

Таблиця 2.4 - Переваги і недоліки UDK

Недоліки	Переваги
Тільки для професіоналів	Безкоштовне розповсюдження
Повільна компіляція	Простий, зручний інтерфейс;
Не самі передові технології	Великий набір інструментів для створення ігор;
Не багатоплатформовий	Багатоплатформовий

Що до функціоналу, гнучка система анімації дозволяє контролювати кожну деталь анімованого об'єкта. Анімаційна модель контролюється системою AnimTree, яка включає наступні механізми: контролер змішання (Blend), контролер керований даними, фізичні, процедурно-скелетні контролери. Для імпортування об'єктів використовується формат FBX, що став стандартом для експорту моделей між редакторами. Для візуалізації UE3 використовує 64-бітний кольоровий HDR графічний конвеєр, який здійснює гамма-корекцію, розмиття рухомих об'єктів, зовнішню окклюзию і інші ефекти постобробки. Движком підтримуються всі сучасні ефекти освітлення і технології візуалізації: нормалізовані карти, параметризоване освітлення по Фонгу, різні анізотропні ефекти та інше. UE3 відомий своєю високо оптимізованою мережевою архітектурою, що включає підтримку онлайнових баталій для ігор різних жанрів.

Frostbite Engine - ігровий движок, розроблений компанією EA Digital Illusions CE; застосовується як у власних розробках, так і проектах інших філіалів Electronic Arts (EA). Першою грою, створеною з використанням цього движка стала Battlefield: Bad Company 2008 року. Движок був розроблений для заміни технічно застарілої технології Refractor Engine, яка використовувалася в попередніх іграх фірми.

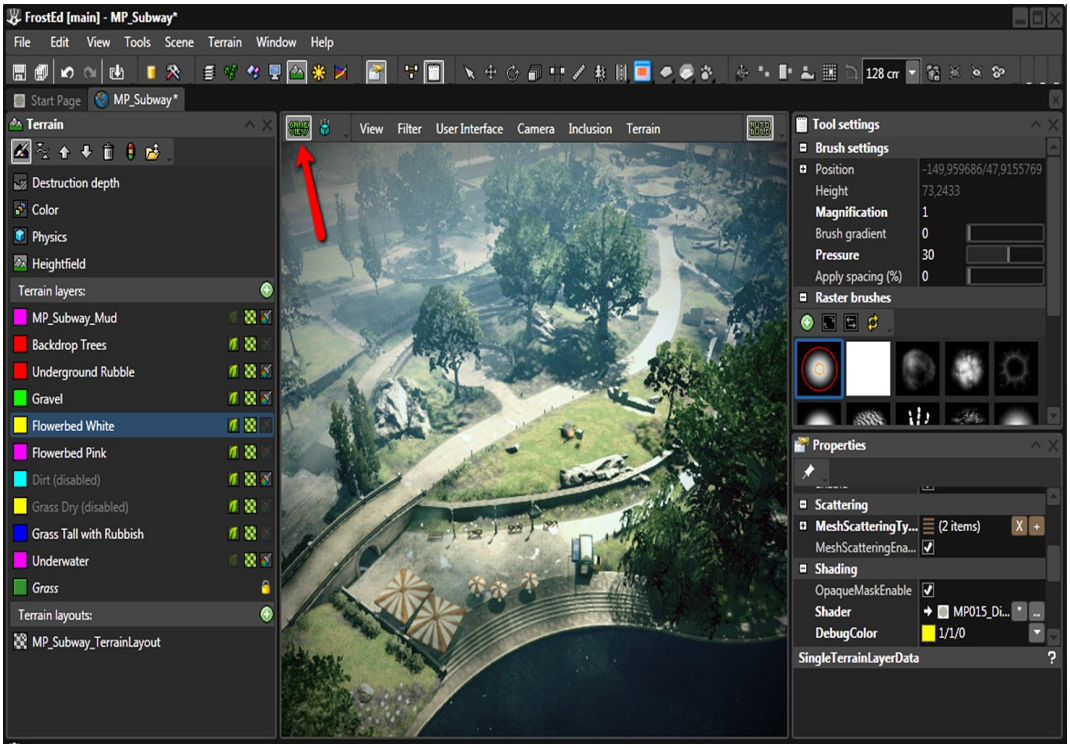

Рисунок 2.5 - Основний інтрефейс Frostbite Engine

Таблиця 2.5 - Переваги і недоліки Frostbite Engine

Недоліки	Переваги
Не безкоштовний	Ідеальне руйнування ландшафту
Не багатоплатформовий	Динамічне освітлення
Високі вимоги до заліза	Ігровий редактор FrostED
Неможливо продавати свої ассети та скріпти	Власний звуковий движок

Движок відноситься до типу подпрограммного забезпечення (англ. Middleware) і являє собою зв'язку декількох компонентів, таких як графічний движок, звуковий движок і т.д. В операційній системі Microsoft Windows ігровий движок підтримує відображення графіки за допомогою Mantle починаючи з версії 3, DirectX 9, DirectX 10, DirectX 10.1, а починаючи з версії 1.5 - і DirectX 11. Однією із заявлених особливостей є оптимізація для роботи на багатоядерних процесорах.

Технологія здатна обробляти знищення ландшафту і оточення (наприклад, будівель, дерев, автомобілів). Підтримується динамічне освітлення і затінення з функцією HBAO, процедурний шейдинг, різні пост-ефекти (наприклад, HDR і depth of field), система частинок і техніки текстурирования, такі, як бамп-маппінг. Максимальний розмір локації становить обмеження в 32 × 32 кілометри відображаємої площі і 4 × 4 кілометри ігрового простору. Крім цього, за твердженням творців, максимальна дистанція промальовування дозволяє побачити рівень аж до горизонту. Також вбудований власний звуковий движок, що не вимагає використання спеціалізованих засобів, подібних EAX.

Згадана лише мізерна частина доступних для використання ігрових движків.

Таблиця 2.6 - Допоміжна порівняльна характеристика усіх движків

Характеристика	Вимоги	Unity 3D	Torque 2D/3D	CryEngine 3	UDK	Frostbite Engine
Режим рендеринга	3D	+	-	+	+	+
Звуковий движок		-	-	-	+	+
Фізичний движок		+	+	+	+	+
Ігровий II		+	-	-	+	+
Цільовий жанр	RPG	+	+	+	-	+
Мова розробки	C# или JavaScript	+	-	+	-	+
Інтеграція з IDE	Visual Studio	+	+	+	+	-
ОС для розробки	Windows	+	+	+	+	+
Цільові платформи	Windows, OSX	+	+	+	+	+
Ціна	безкоштовно	+	+	-	+	-
Цільова аудиторія	Для професіоналів	Так	Так	Так	Ні	Так
Якість документації		3	2	2	1	3
Інтерфейс користувача		3	2	3	3	2
Результуючий рейтинг		6	4	5	4	5

2.2 Функціональні можливості Unity 3D

Unity - це інструмент для розробки двох-і тривимірних додатків та ігор, що працює під операційними системами Windows, Linux і OS X. Створені за допомогою Unity програми працюють під операційними системами Windows, OS X, Windows Phone, Android, Apple iOS, Linux, а також на ігрових приставках Wii, PlayStation 3, PlayStation 4, Xbox 360, Xbox One і MotionPar-allax3D дисплеях (пристрої для відтворення віртуальних голограм), наприклад, Nettlebox. Є можливість створювати додатки для запуску в браузерах за допомогою спеціального модуля Unity (Unity Web Player), а також за допомогою реалізації технології WebGL. Раніше була експериментальна підтримка реалізації проектів в рамках модуля Adobe Flash Player, але пізніше команда розробників Unity прийняла складне рішення щодо відмови від цього.

Рисунок 2.6 - Браузерна гра запущена за допомогою Unity Web Player

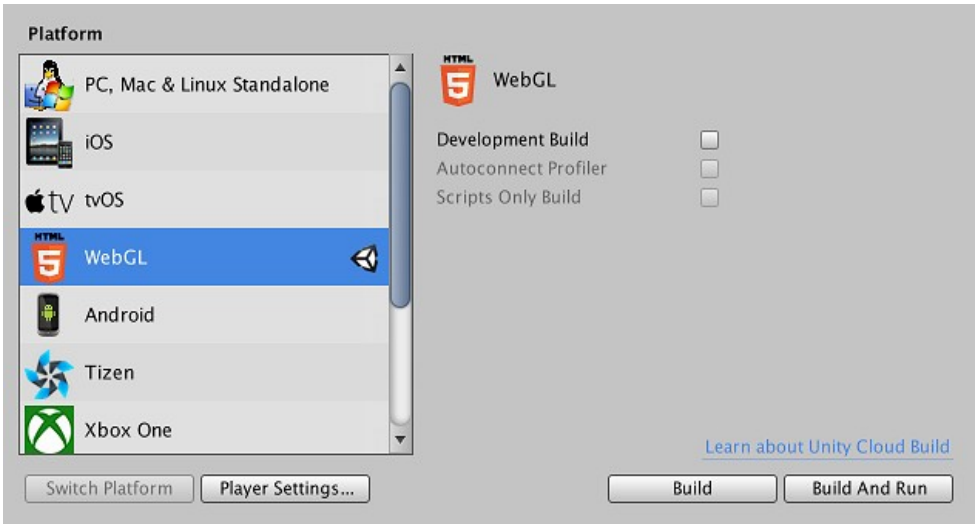

Рисунок 2.7 - Технологія WebGL для запуску браузерних ігр

Додатки, створені за допомогою Unity, підтримують DirectX і OpenGL. Активно движок використовується як великими розробниками (Blizzard, EA, QuartSoft, Ubisoft), так і розробниками Indie-ігор (наприклад, Pathologic, Kerbal, Space Program, Slender: The Eight Pages, Slender : The Arrival, Surgeon Simulator 2013, Baeklyse Apps: Guess the actor та інші) в силу наявності безкоштовної версії, зручного інтерфейсу і простоти роботи з движком.

Редактор Unity має простий Drag & Drop інтерфейс, який легко налаштовувати, що складається з різних вікон, завдяки чому можна

проводити налагодження гри прямо в редакторі. Движок підтримує три сценарних мови: C #, JavaScript (модифікація), Boo (діалект Python). Boo прибраний в 5-ій версії. Розрахунки фізики виробляє фізичний движок від NVIDIA.

Unity 5 володіє величезною кількістю переваг перед іншими ігровими движками. Ком'юніті Unity 5 на сьогоднішній момент є найбільшим в світі. На офіційному сайті Unity є спеціальний розділ, в якому можна знайти статистику по ігровим движкам. За цими даними Unity 5 використовує понад 50% розробників відеоігор, 20% належать Unreal Engine, а решта ігрових движків - 30%. Для розробки 2D або 3D інді-ігор Unity 5 підходить за всіма параметрами.

Розробка AAA-проектів в Unity - найскладніший процес. По-перше, будь-який скрипт відразу тягне за собою купу помилок, які в майбутньому необхідно виправити, або переписати скрипт заново. По-друге, Unity 5 все ще має погану оптимізацію. Весь контент який стоїть у вікні Project, але не стоїть в сцені, буде запечений, а значить що гра буде важити в рази більше, ніж передбачалося. Unity в найближчому оновленні виправить цю проблему. У движку є ряд проблем зі скроллингом. При приближенні до об'єкта в певний момент камера наближається повільніше. Якщо потрібно максимально близько наблизиться до землі, то іноді це буває дуже складно зробити. Швидше за все в найближчих оновленнях скролінг так само виправлять.

3 3D РЕДАКТОР BLENDER ТА СТВОРЕННЯ СХОДІВ ДЛЯ ГРИ

Для 3D-моделювання був використаний редактор Blender - вільний, професійний пакет для створення тривимірної комп'ютерної графіки, що включає в себе засоби моделювання, анімації, рендеринга, постобробки і монтажу відео зі звуком, компонування за допомогою «вузлів» (Node Compositing), а також для створення інтерактивних ігор. В даний час користується найбільшою популярністю серед безкоштовних 3D редакторів в зв'язку з його швидким і стабільним розвитком, якому сприяє професійна команда розробників..

Характерною особливістю пакету Blender є його невеликий розмір в порівнянні з іншими популярними пакетами для 3D-моделювання. У базову поставку не входить розгорнута документація і велика кількість демонстраційних сцен.

Функції пакета:

- Підтримка різноманітних геометричних примітивів, включаючи полігональні моделі, систему швидкого моделювання в режимі subdivision surface (SubSurf), криві Безьє, поверхні NURBS, metaballs (метасфери), скульптурне моделювання та векторні шрифти.
- Універсальні вбудовані механізми рендеринга і інтеграція з зовнішнім рендерер YafRay, LuxRender і багатьма іншими.
- Інструменти анімації, серед яких інверсна кінематика, скелетна анімація і сіткова деформація, ключові кадри, нелінійна анімація, редагування вагових коефіцієнтів вершин, обмежувачі, динаміка м'яких тіл (включаючи визначення колізій об'єктів при взаємодії), динаміка твердих тіл на основі фізичного движка Bullet і система волосся на основі частинок.
- Python використовується як засіб створення інструментів і прототипів, системи логіки в іграх, як засіб імпорту, та експорту файлів (наприклад COLLADA), автоматизації завдань.
- Базові функції нелінійного редагування і комбінування відео.
- Blender Game Engine - підпроект Blender, що надає інтерактивні функції, такі як визначення колізій, движок динаміки і програмована логіка. Також він дозволяє створювати окремі real-time додатки починаючи від архітектурної візуалізації до відео ігор.

Blender мав репутацію програми, складної для вивчення. Практично кожна функція має відповідне їй поєднання клавіш, і враховуючи кількість можливостей, що надаються Blender, кожна клавіша включена в більш ніж одне поєднання (shortcut). С тих пір як Blender став проектом з відкритим вихідним кодом, були додані повні контекстні меню до всіх функцій, а використання інструментів зроблено більш логічним і гнучким. Додамо сюди подальше поліпшення користувальницького інтерфейсу з введенням колірних схем, прозорих плаваючих елементів, новою системою перегляду дерева об'єктів і різними дрібними змінами.

Відмінні риси інтерфейсу користувача:

– Режими редагування. Два основні режими Об'єктний режим (Object mode) і Режим редагування (Edit mode), які перемикаються клавішею Tab. Об'єктний режим в основному використовується для маніпуляцій з індивідуальними об'єктами, в той час як режим редагування - для маніпуляцій з фактичними даними об'єкта. Наприклад, для полігональної моделі в об'єктному режимі ми можемо переміщати, змінювати розмір і обертати модель цілком, а режим редагування використовується для маніпуляції окремих вершин конкретної моделі. Також є кілька інших режимів, таких як Vertex Paint та UV Face select.
– Широке використання гарячих клавіш. Більшість команд виконується за допомогою 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 більш повне графічне меню.
– Управління робочим простором. Графічний інтерфейс Blender'a складається з одного або декількох екранів, кожен з яких може бути розділений на секції і 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 і збільшувати кнопки інструментів тим же шляхом, що і в 3D перегляді. Користувач повністю контролює розташування і організацію графічного інтерфейсу, це робить можливим налаштування інтерфейсу під конкретні завдання, такі як редагування відео, UV mapping і текстурування, і приховування

елементів інтерфейсу які не потрібні для даного завдання. Цей стиль графічного інтерфейсу дуже схожий на стиль, використовуваний в редакторі UnrealEd карт для гри Unreal Tournament.

Робочий простір Blender'a вважається одним з найбільш новаторських концепцій графічного інтерфейсу для графічних інструментів і натхненним дизайном графічного інтерфейсу патентованих програм, таких як Luxology's Modo.

Додаткові особливості:

- У програмі Blender сутність, що взаємодіє з навколишнім світом і її дані (форма або функції об'єкта) разделяемості. Ставлення Об'єкт-Дані представляється відношенням 1: n (термін, що відноситься до теорії баз даних, позначає можливість декількох об'єктів використовувати одні і ті ж дані - один до багатьох або сюрьекція).
- Внутрішня файлова система, що дозволяє зберігати кілька сцен в єдиному файлі (званому .blend файл).
- Всі «.blend» файли сумісні як із старішими, так і з більш новими версіями Blender. Так само всі вони переносяться з однієї платформи на іншу і можуть використовуватися як засіб перенесення створених раніше робіт.
- Blender робить резервні копії проектів під час всієї роботи програми, що дозволяє зберегти дані при непередбачених обставинах.
- Всі сцени, об'єкти, матеріали, текстури, звуки, зображення, post-production ефекти можуть бути збережені в єдиний «.blend» файл.
- Налаштування робочого середовища можуть бути збережені в «.blend» файл, завдяки чому при завантаженні файлу ви отримаєте саме те, що зберегли в нього. Ви можете зберегти файл як «власні за замовчуванням», і кожен раз при запуску Blender ви будете отримувати необхідний набір об'єктів та підготовлений до роботи інтерфейс.

Один із об'єктів створених для гри за допомогою Blender це "Сходи".

Покрокова інструкція по створенню сходів:

1. Виділімо куб, перейдемо в режим редагування (Tab), розтягнімо його по осі-Y і зменшимо по осі-Z до розмірів майбутньої сходинки. Потім, за допомогою екструдування (E), створимо виступ спереду і по краях.

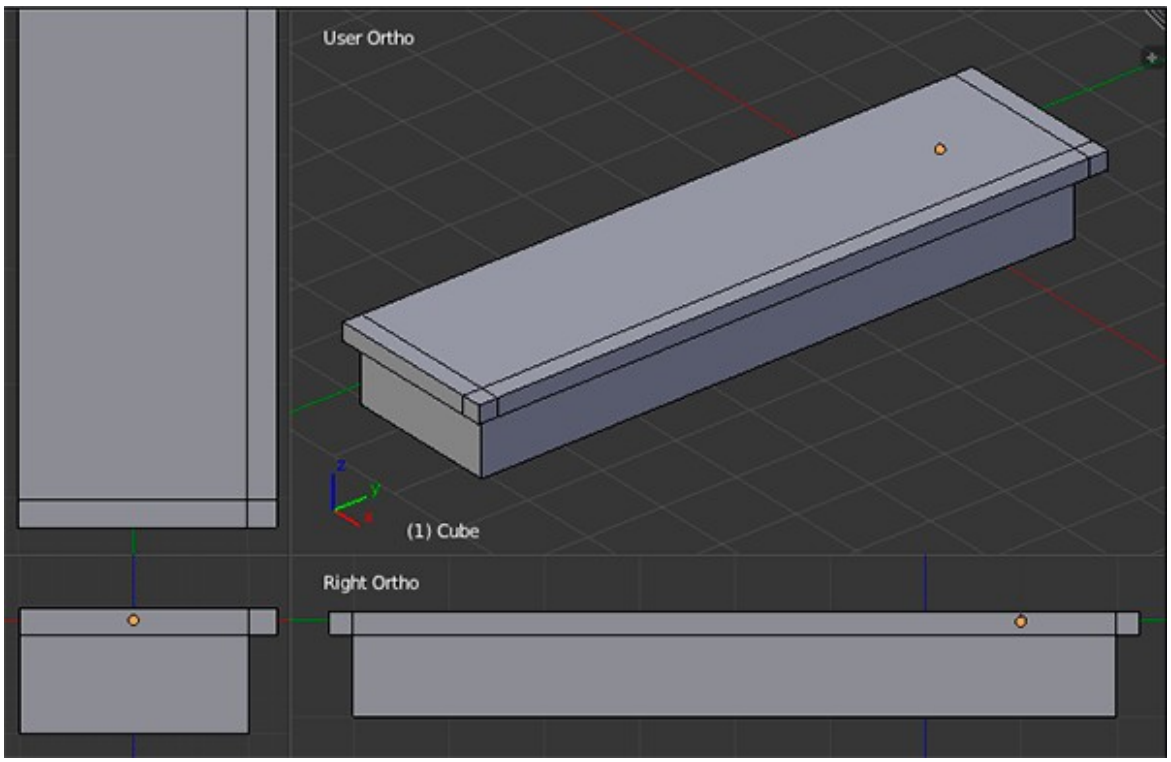

Рисунок 3.1 - Створення сходинки із кубу

2. У режимі виділення ребер, виділимо верхнє ребро сходинки і округлимо його, за допомогою інструменту Bevel (Ctrl + B).

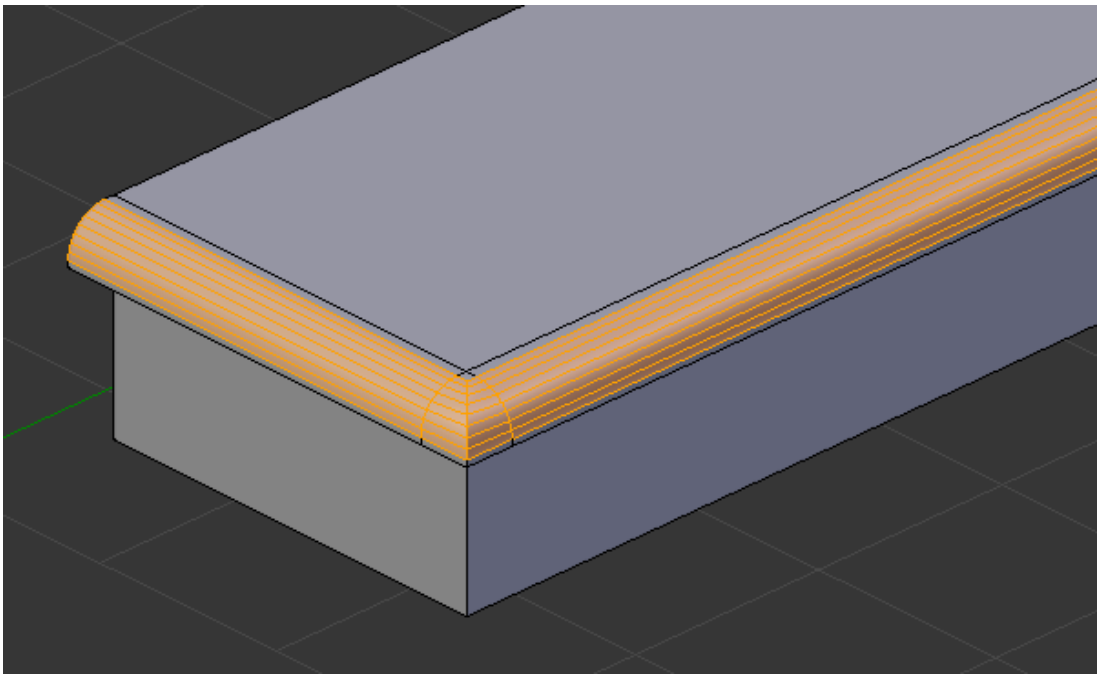

Рисунок 3.2 - Округлення сходинок

3. Після цього додамо шейдер Smooth і модифікатор Edge Split. На цьому створення сходинки закінчено.
4. Тепер потрібно створити кілька копій даної сходинки. Для цих цілей ідеально підійде модифікатор Array. Додамо його, вкажемо параметр Z = 1, а параметр X такий, щоб не було зазорів між сходинками. Значення Count виставимо 12.

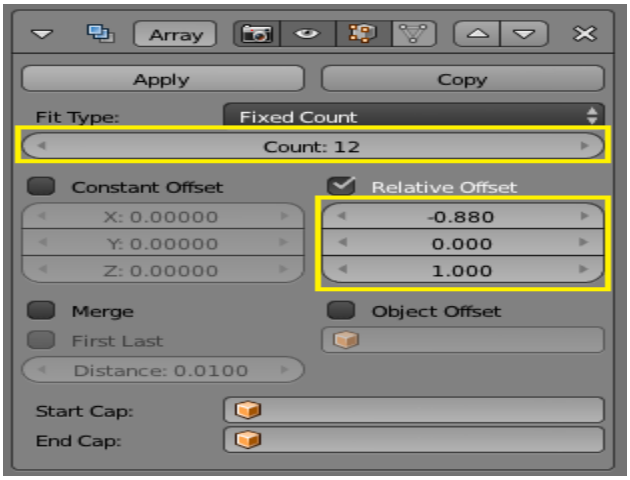

Рисунок 3.3 - Встановлюємо кількість сходинок 12

5. В результаті, вийде ось така драбинка:

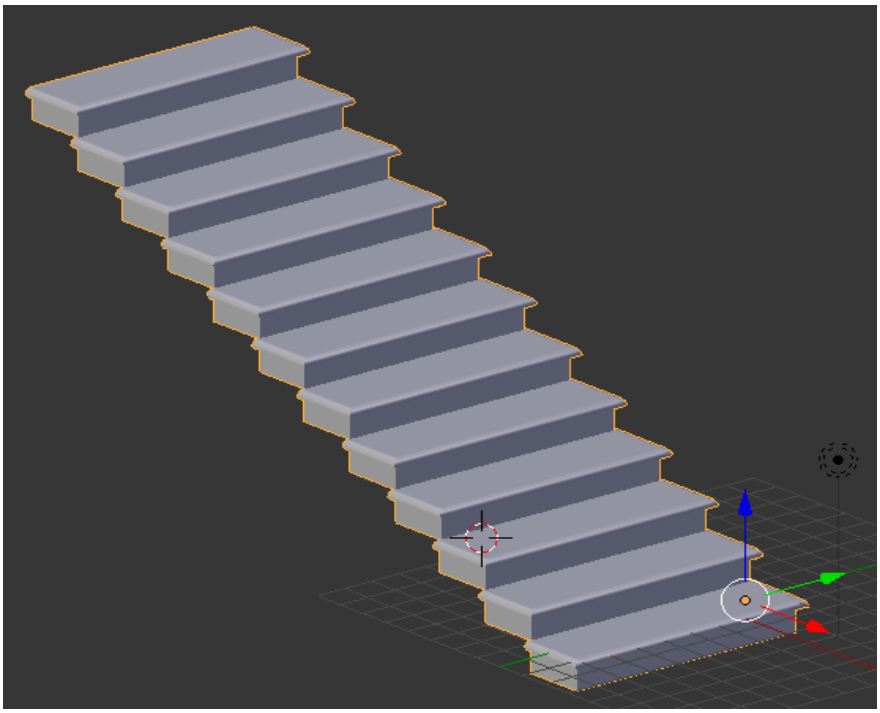

Рисунок 3.4 - Кінцевий результат створення 12 сходинок

6. Тепер займемося створенням опорного стовпа. Додамо в сцену циліндр (Shift + A> Mesh> Culinder), і відразу ж виставимо для нього наступні значення:

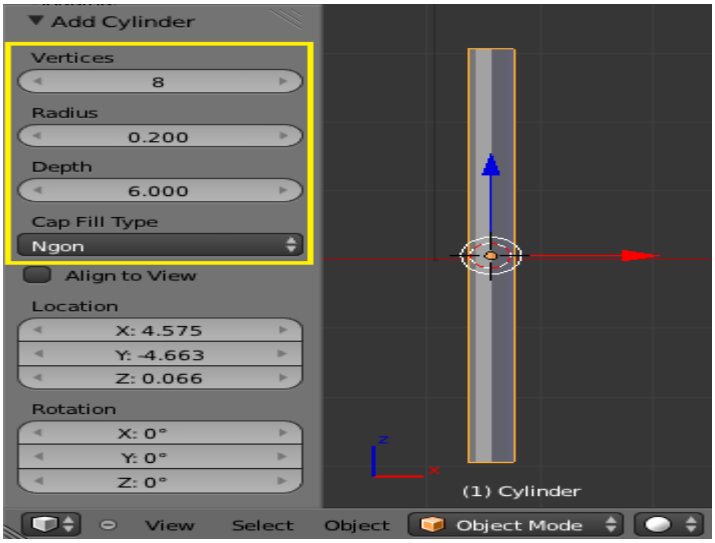

Рисунок 3.5 - Налаштування вершин, радіусу та глубини

7. Тепер знову повторимо процедуру з модифікатором Array. Виставимо даний циліндр на першій сходинці, а потім додамо модифікатор і відрегулюємо значення X і Z, щоб в результаті вийшло так:

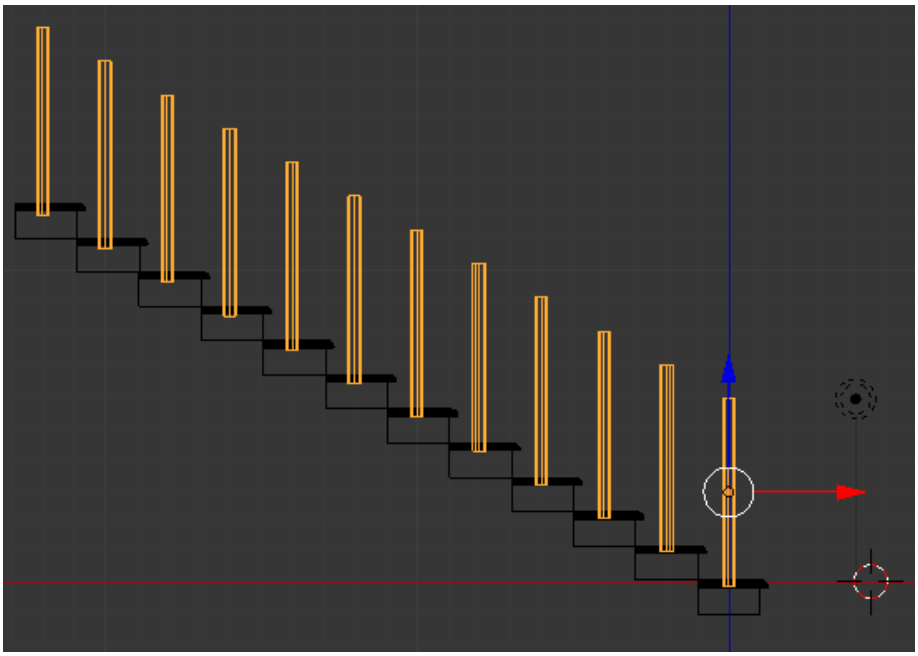

Рисунок 3.6 - Установка опору для кожної сходинки

8. Створимо дублікат опорного стовпа для іншої сторони сходів:
9. Останньою деталлю буде створення перил. Додамо в сцену куб (Shift + A> Mesh> Cube), зменшімо його масштаб і розтягнемо по всій довжині сходів.

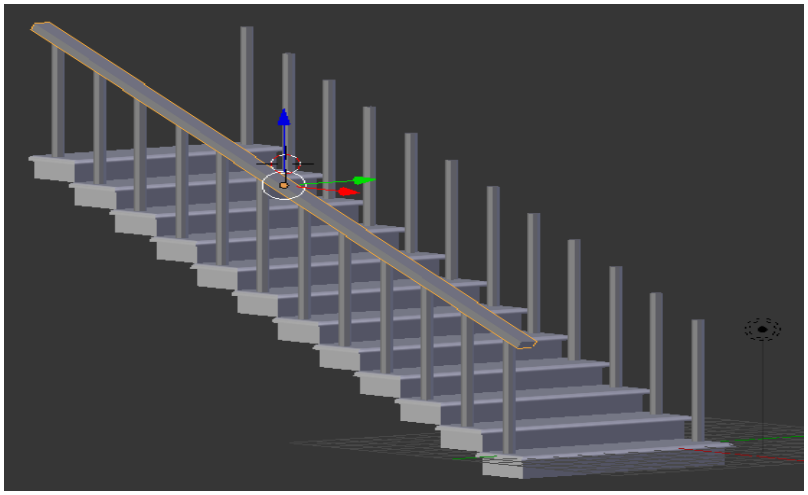

Рисунок 3.7 - Створення перил з одного боку за допомогою куба

10. Потім створимо пов'язаний дублікат (Alt + D), і розташуєм його на протилежному боці. Виділивши два верхніх ребра закруглімо їх, за допомогою інструменту Bevel (Ctrl + B).

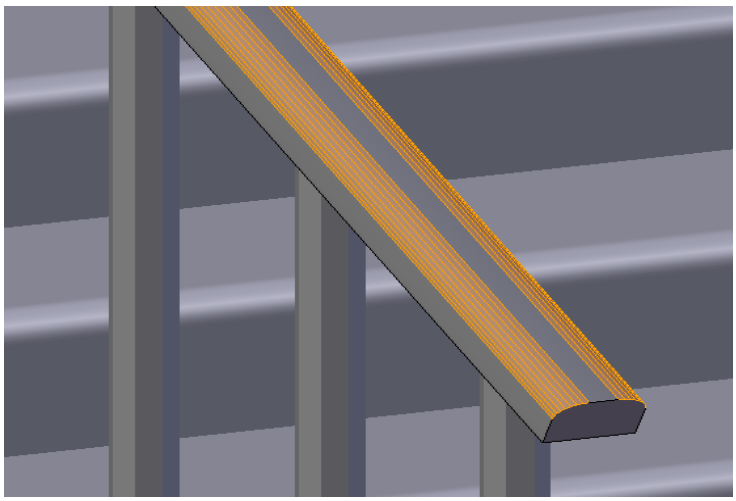

Рисунок 3.8 - Створення дублікату з другого боку

11. Сходи готові і тепер можна налаштувати освітлення і світ. На вкладці світу виставимо повністю чорний колір. Додамо в сцену площину (Shift + A> Mesh> Plain), розташуємо її над сходами, збільшемо її масштаб у 5 разів (S | 5 | Enter) і повернемо приблизно на 30 градусів.Створимо для неї матеріал Emission і силу світіння (Strength) 30. Виставимо камеру перед сходами і зробим пробний рендер.

Рисунок 3.9 - Кінцевий вигляд сходів

12. Тепер можна накласти будь-яку текстуру дерева або ж скористатися процедурною генерацією текстури, і виконати фінальний рендер.

4 РОЗРОБКА СЦЕНИ В СЕРЕДОВИЩІ UNITY 3D

4.1 Збірка сцен. Gameobjects і Components

Ігрові об'єкти (GameObjects) - це найважливіші об'єкти в Unity. Дуже важливо розуміти, що таке GameObject і як його використовувати.

Кожен об'єкт в грі - це GameObject. Однак, GameObject'и нічого не роблять самі по собі. Вони вимагають спеціальної настройки, перш ніж стати персонажами, предметами оточенням або спеціальними ефектами.

GameObject'и є контейнерами. Порожня коробка, яка може містити всередині різні елементи, такі як острів з запеченими тінями або фізично коректний автомобіль. Щоб дійсно зрозуміти GameObject'и, потрібно зрозуміти їх складові, який називаються компонентами (Components). Залежно від того, який ми хочемо створити об'єкт, ми будемо додавати різні комбінації компонентів до GameObject'у.

Відкриймо будь-яку сцену Unity, створімо новий GameObject (використовуючи Shift-Control-N в Windows або Shift-Command-N в Mac), виберімо його і поглянимо в інспектор (Inspector).

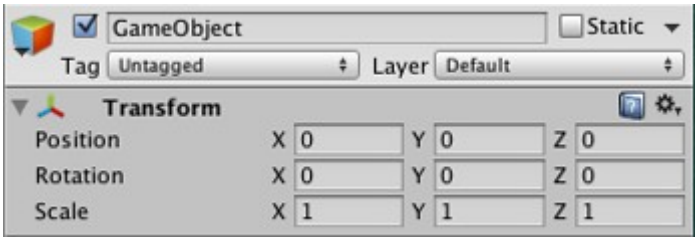

Рисунок 4.1 - Інспектор порожнього GameObject'а

Звернімо увагу, що навіть порожні GameObject'и мають ім'я, тег (Tag), і шар (Layer). Кожен GameObject також містить компонент Transform.

У Unity неможливо створити GameObject без компонента Transform. Компонент Transform - один з найважливіших компонентів, так як всі властивості GameObject'а пов'язані з трансформаціями використовують цей компонент. Він визначає положення, обертання і масштаб GameObject'а в ігровому світі / вікні Scene. Якщо GameObject не матиме компонента Transform, він буде не більше ніж деякою інформацією в пам'яті комп'ютера. Він не зможе ефективно існувати в ігровому світі.

Компонент Transform важливий для всіх GameObject'ов, тому він є у кожного GameObject. Але GameObject'и можуть містити й інші компоненти.

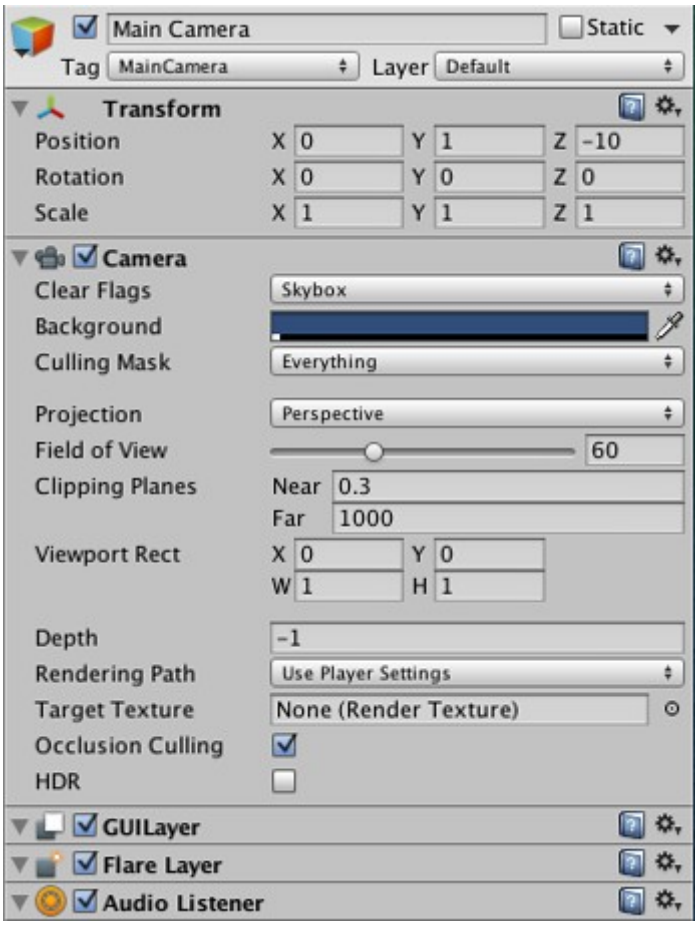

Рисунок 4.2 - GameObject під назвою Main Camera, який додається в кожну нову сцену за замовчуванням

Поглянувши на GameObject Main Camera, ми можемо побачити, що він містить різні колекції компонентів. Особливо, компонент Camera, GUILayer, Flare Layer, і Audio Listener. Всі ці компоненти надають додаткову функціональність GameObject'у. Без них не було б кому займатися рендерингом ігрової графіки для граючої людини! Тверді тіла, колайдери, частинки, аудіо - це все різні компоненти (або комбінації компонентів), які можуть бути додані до будь-якого GameObject'у.

Однією з чудових особливостей компонентів є гнучкість. При підключенні компонента до ігрового об'єкту, існують різні значення або властивості (Properties) в компоненті, які можуть змінюватися в редакторі при створенні гри або через скрипти в запущеній грі. Є два основних типи властивостей: значення (Values) і посилання (References).

Подивимося на рис 4.3., це порожній Ігровий Об'єкт з компонентом Audio Source. Всі параметри компонента Audio Source в Інспектора виставлені за замовчуванням.

Рисунок 4.3 - порожній Ігровий Об'єкт з компонентом Audio Source

Компонент містить одну властивість-посилання і сім властивостей-значень. Audio Clip - це властивість-посилання. Коли це аудіо джерело починає грати, він буде намагатися програти файл, на який посилається властивість Audio Clip. Якщо такого посилання не виявиться, то виникне помилка, так як ніяке аудіо НЕ буде програно. Ми повинні призначити файл в інспектор. Перетягнемо файл з Project View на властивість-посилання або за допомогою вибору об'єкта (Object Selector).

Всі інші властивості після Audio Clip це властивості-значення. Вони можуть бути відрегульовані прямо в інспектор. Властивості значення після Audio Clip - це все перемикачі, числові значення і випадаючі меню, але властивості-значення можуть також бути текстовими рядками, кольорами, кривими і іншими типами.

4.1.1 Публікація збірок

У будь-який момент розробки гри можна захотіти подивитися на те, як вона виглядає поза редактора, при складанні в якості самостійного додатка або веб-програвача.

Пункт меню File-> Build Settings ... дозволяє відкрити вікно Build Settings. У ньому виводиться редагований список сцен для включення в збірку гри.

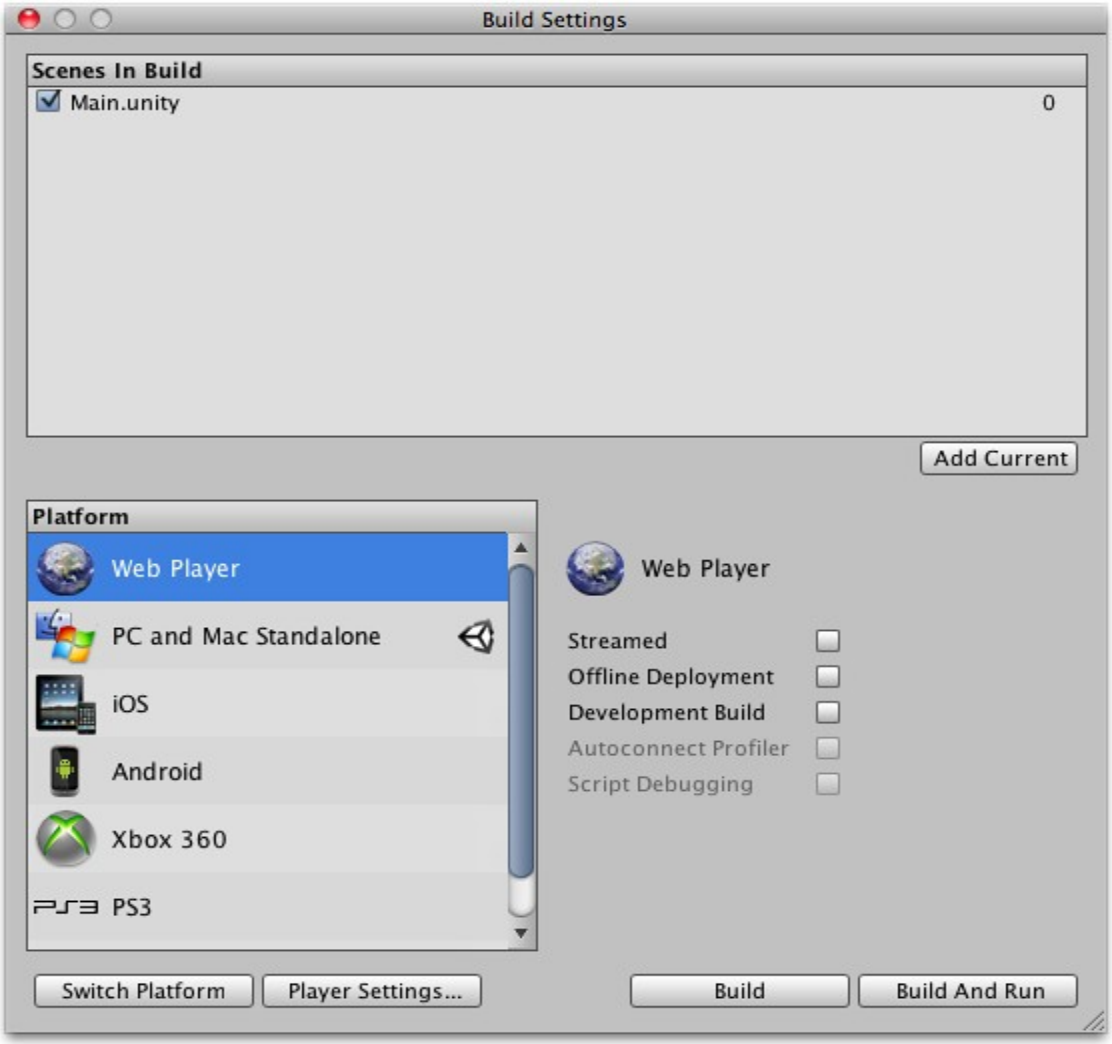

Рисунок 4.4 - Build Setting у якому можно скомпілювати наш проект

Список буде порожній при першому відкритті цього вікна в проекті. В такому випадку, при складанні, в гру буде включена лише поточна відкрита сцена.

Коли будемо готові до публікації своєї збірки, вибераємо потрібну платформу в списку Platform, навпроти неї знаходиться логотип Unity; якщо

це не так, тоді натиснимо кнопку Switch Platform, щоб повідомити Unity про те, під яку платформу ми бажаємо здійснювати збірку. Після цього, натиснимо кнопку Build. Відкриється стандартне діалогове вікно збереження файлу, в якому ми можемо вибрати ім'я і розташування для гри. Після натискання кнопки "Зберегти" Unity збере ваш додаток.

4.1.2 Редагування властивостей

Properties (властивості) - параметри і опції компонентів, які можна редагувати в інспекторі.

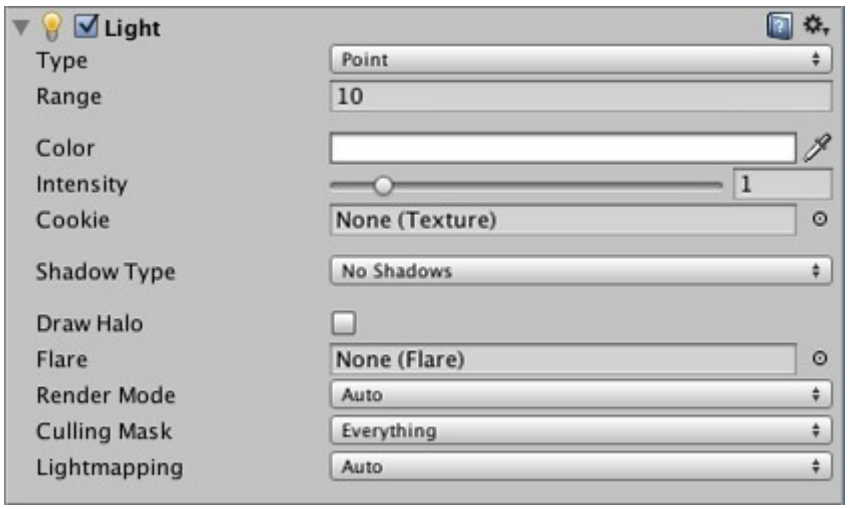

Рисунок 4.5 - Компонент Light відображає різні значення і властивості-посилання

Властивості можна широко розділити на категорії "посилання" (зв'язки з іншими об'єктами і Ассетами) або величини (числа, прапорці, кольори і т.д.).

Посилання можуть бути призначені шляхом перетягування об'єкта або Ассета відповідного типу на відповідний пункт в інспекторі. Наприклад, компоненту Mesh Filter треба послатися на Ассет Mesh десь в цьому проекті. Коли компонент тільки створено, посилання ще не призначена:

Рисунок 4.6 - Посилання не призначене

але ми можемо призначити йому Mesh шляхом перетягування Ассет-міша на нього:

Рисунок 4.7 - Посилання призначене

Ми також можемо використовувати Object Picker, щоб вибрати об'єкт і зв'язати його з властивістю. Якщо ми клацнімо на значок невеликого кола в правій частині властивості в інспекторі, ми побачимо подібне вікно:

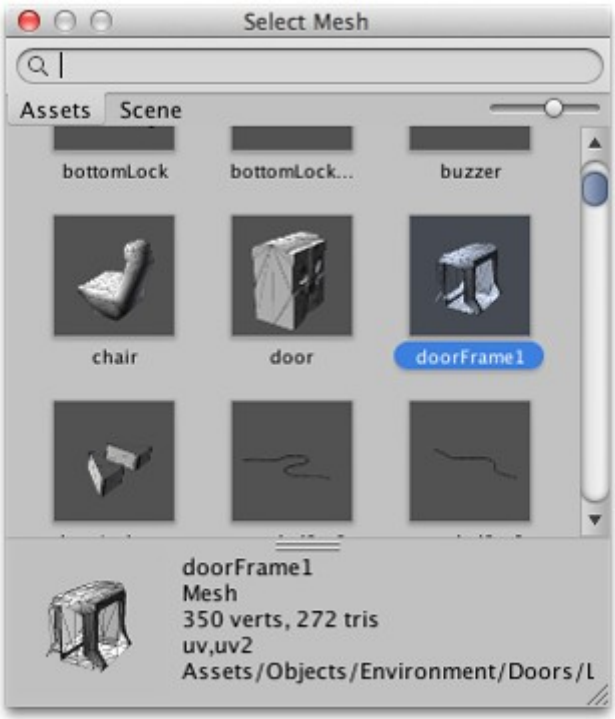

Рисунок 4.8 - Object Picker для пошуку об'єкта в сцені або ассетах

Object picker (вибір об'єкта) дозволяє шукати и вібіраті об'єкти в сцені або в Ассет проекту (інформаційні панелі в Нижній части вікна можна підніматі и опускаті за Бажанов). Для Вибори об'єкта на властівість-ПОСИЛАННЯ, просто необходимо двічі натіснуті на него в Object Picker.

Коли властівість-ПОСИЛАННЯ відноситься до компоненту типу (например Transform), Ми можемо помістіті в Цю властівість будь-який об'єкт; Unity Визначіть перший компонент цього типу зустрічаючійся в цьом

об'єкті и призначила на ПОСИЛАННЯ. Если в об'єкта НЕ виявило компонентів відповідного типу, призначення буде перервати.

При кліку по колірної Властивості, відкриється Палітра кольорів (Color Picker).

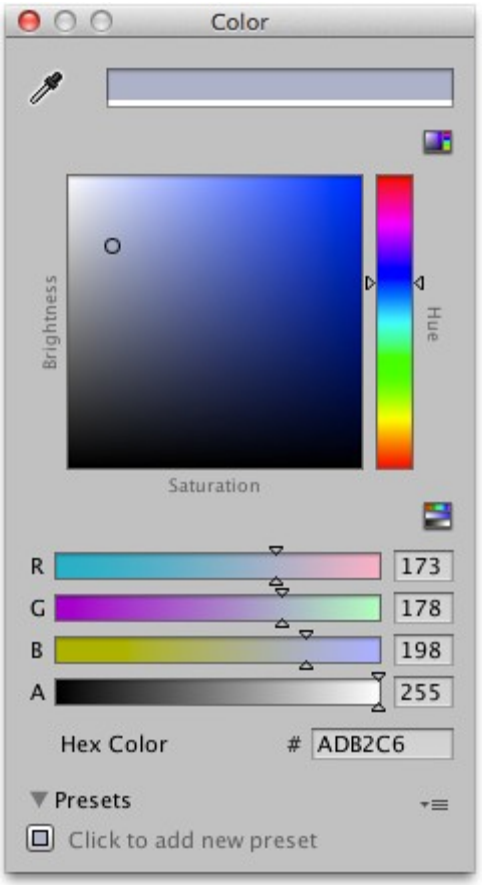

Рисунок 4.9 - Вікно палітри кольорів

Unity використовує власну систему кольорової палітри, але на Mac OS X ми можемо вибрати системну систему з меню Preferences (меню: Unity> Preferences і потім Use OS X Color Picker з панелі General).

4.2 Використання вікна Scene View

Вікно Scene View - це інтерактивна пісочниця. Будемо використовувати Scene View для вибору і розташування оточень, гравця, камери, ворогів, і всіх інших ігрових об'єктів. Управління та маніпулювання об'єктами з використанням вікна Scene View є однією з найбільш важливих функцій Unity, тому важливо вміти робити це швидко.

4.2.1 Навігація у вікні Scene

Вікно Scene має набір навігаційних елементів управління, які допомогають орієнтуватися в ньому швидко і ефективно.

Для переміщення по сцені можна використовувати клавіші зі стрілками. Клавіші вгору і вниз переміщують камеру вперед і назад в тому напрямку, в яке вона дивиться. Клавіші вліво або вправо повертають камеру в сторони. При використанні стрілок, затиснимо клавішу Shift для прискореного переміщення.

Якщо вибрати геймоб'єкт в ієрархії, потім помістити миш над Scene View, і натиснути Shift + F, вид концентрується на вибраному об'єкті. Ця можливість називається кадрувати виділене (frame selection).

Рисунок 4.10 - Элементи фокусування

Переміщення, обертання по орбіті і масштабування - ключові операції навігації у вікні Scene View, тому Unity надає кілька альтернативних шляхів їх виконання для максимальної зручності.

4.2.2 Позиціонування ігрових об'єктів

Під час створення гри, я повинен розмістити багато різних об'єктів у ігровому світі. Для цього будуть використані інструменти трансформацій в панелі інструментів для переміщення, обертання і масштабування окремих об'єктів. Кожен інструмент має відповідне Гизмо, яке з'являється навколо виділеного ігрового об'єкта у вікні Scene. Можна використовувати миш і маніпулювати будь якою віссю Гизмо для зміни компонента Transform ігрового об'єкта, або можна ввести значення безпосередньо в числові поля компонента в інспекторі.

Кожен з трьох режимів може бути обраний гарячими клавішами - W для переміщення, E для обертання і R для масштабування.

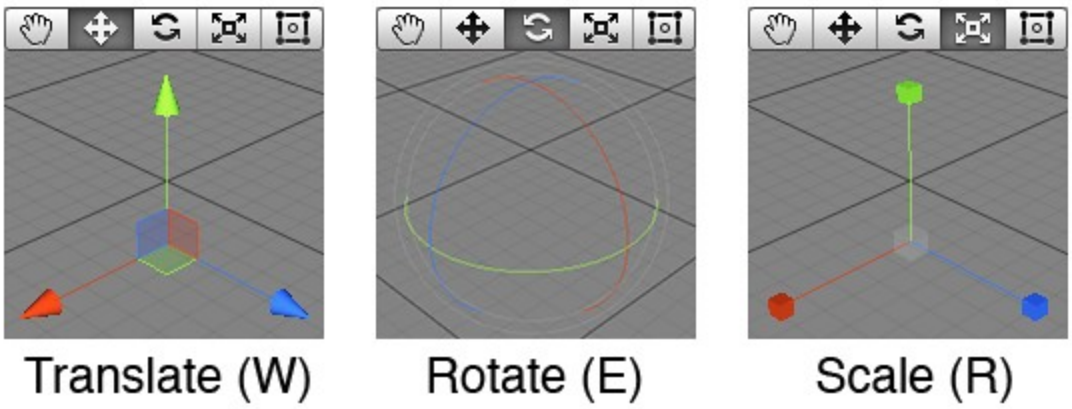

Рисунок 4.11 - Переміщення, обертання і масштабування

4.2.3 Пошук в Scene

При роботі з великими і складними сценами буде корисний пошук певних об'єктів. Використовуючи можливість Search в Unity, можна відфільтрувати об'єкт або групу об'єктів. Можна шукати Ассет по імені, по типу компонента, і, в деяких випадках, по мітках Ассет (див. Нижче). Можна вибрати режим пошуку з випадаючого меню Search.

У вікнах Scene і Hierarchy є поле пошуку, що дозволяє фільтрувати об'єкти по імені. Так як ці вікна, по суті, просто два варіанти відображення одних і тих же об'єктів, будь-якій пошуковій запит буде дублюватися в обох полях пошуку і застосовуватися однаково до обох вікон.

Звернемо увагу, що коли пошук активний, обидва вікна трохи відрізняються: вікно Scene буде показувати об'єкти, які не пройшли по фільтру сірим, а у вікні Hierarchy будуть відображатися тільки ті об'єкти, імена яких відповідають пошуковому запиту:

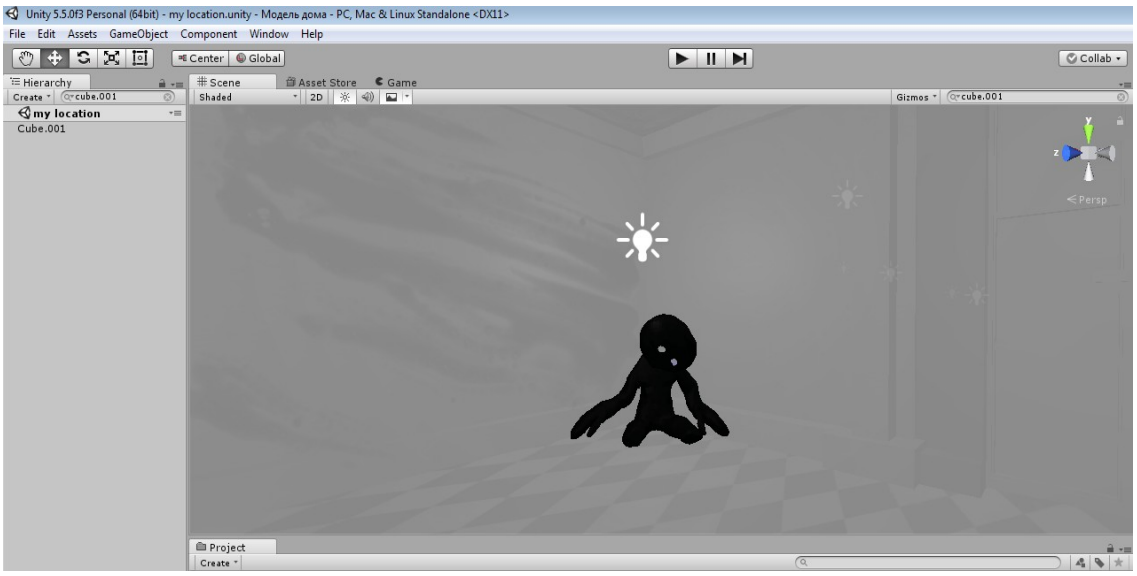

Рисунок 4.12 - Пошук певного об'єкта у проекті

Невеликий хрестик праворуч від поля пошуку видаляє пошукової запит і повертає вікно до нормального стану. В меню зліва від поля можна вибрати фільтр пошуку - по імені об'єкта, за його типу, або обом цим параметрам.

4.3 Префаб (Prefabs)

Досить зручно працювати з GameObject в сцені, додаючи компоненти і змінюючи їх значення на потрібні у інспекторі. Однак, це може створити ряд проблем в таких випадках, коли працюємо над створенням NPC, об'єктом або предметом, який часто зустрічається в сцені. Звичайно, можна просто скопіювати ці об'єкти для створення дублікатів, але всі вони будуть редагуватися незалежно один від одного.

В Unity можна створювати префаб. Це особливий тип Ассета, що дозволяє зберігати весь GameObject з усіма компонентами і значеннями властивостей. Префаб виступає в ролі шаблону для створення екземплярів зберігаючого об'єкта в сцені. Будь-які зміни в префабі негайно відображаються і на всіх його примірниках, при цьому можна перевизначати компоненти і настройки для кожного екземпляра окремо. Коли перетягуємо файловий Ассет (наприклад, меш) в сцену, буде створений новий екземпляр такого об'єкта і всі такі екземпляри зміняться при зміні оригінального Ассета. Однак, хоч його поведінка і схожа, але Ассет - це не префаб, так що не вийде додати до нього компоненти або використовувати будь-які інші описані нижче властивості префабов.

Можна створити префаб, вибравши Asset> Create Prefab і перетягнувши об'єкт зі сцени в "порожній" префаб, що з'явився в проекті. Після чого можна створювати екземпляри префаба просто перетягуючи його з вікна Project на сцену. Імена об'єктів реалізованих префабом, будуть підсвічуватися синім у вікні Hierarchy (імена звичайних об'єктів мають чорний колір).

Як уже згадувалося вище, зміни в префаб автоматично застосуються до всіх її екземплярів, однак можна змінювати і окремі екземпляри. Це корисно наприклад в разі, коли бажаємо створити кілька схожих NPC, але з зовнішніми відмінностями, щоб додати реалістичності. Щоб було чітко видно, що властивість в екземплярі префаб змінено, воно показується в інспекторі жирним шрифтом (якщо до примірника префаба доданий абсолютно новий компонент, то всі його властивості будуть написані жирним шрифтом).

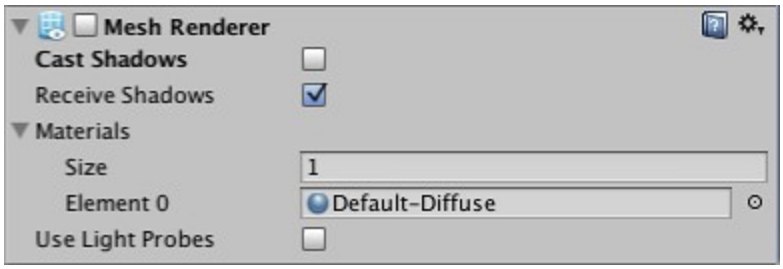

Рисунок 4.13 - Mesh Renderer на екземплярі префаба з перевизначеною властивістю "Cast Shadows"

4.4 Джерела світла

Джерела світла є невід'ємною частиною кожної сцени. У той час як меши і текстури визначають форму і зовнішній вигляд сцени, джерела світла визначають колір і атмосферу вашого 3D оточення. Організація їх одночасної роботи вимагає невеликої практики, але результат може бути приголомшливим.

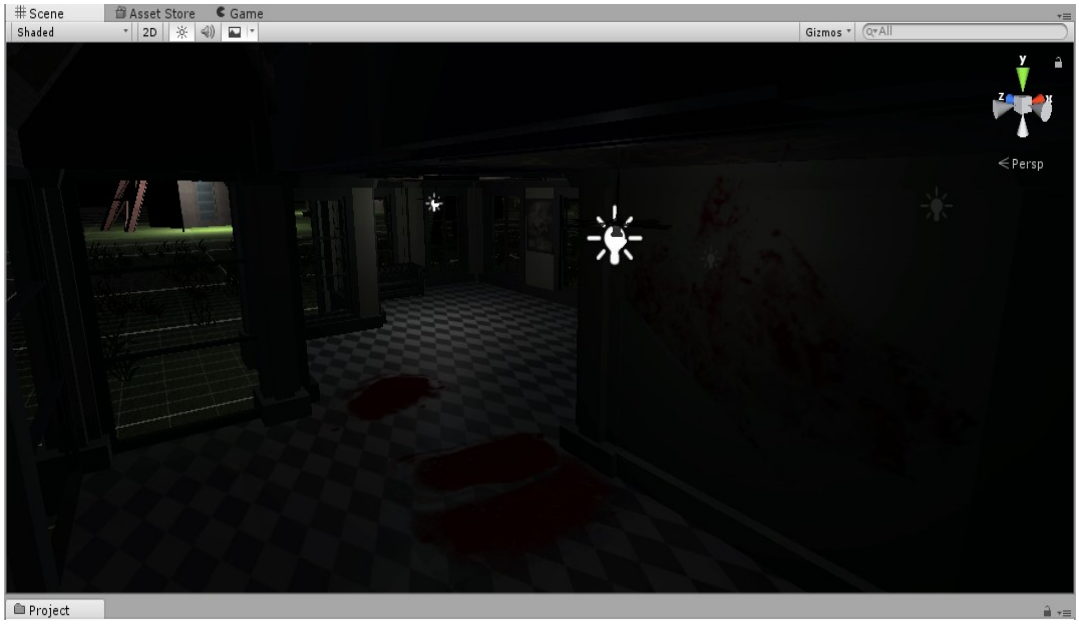

Рисунок 4.14 - Проста компоновка двох джерел світла

Джерела світла можуть бути додані в сцену з використанням меню GameObject-> Create Other. Після того, як джерело світла буде додано до сцени, можна керувати ним як будь-яким іншим GameObject'ом. Також, можна додати компонент Light до будь-якого виділеного об'єкту використовуючи Component-> Rendering-> Light.

Компонент Light має багато різних властивостей, які можна змінювати в інспектора (Inspector).

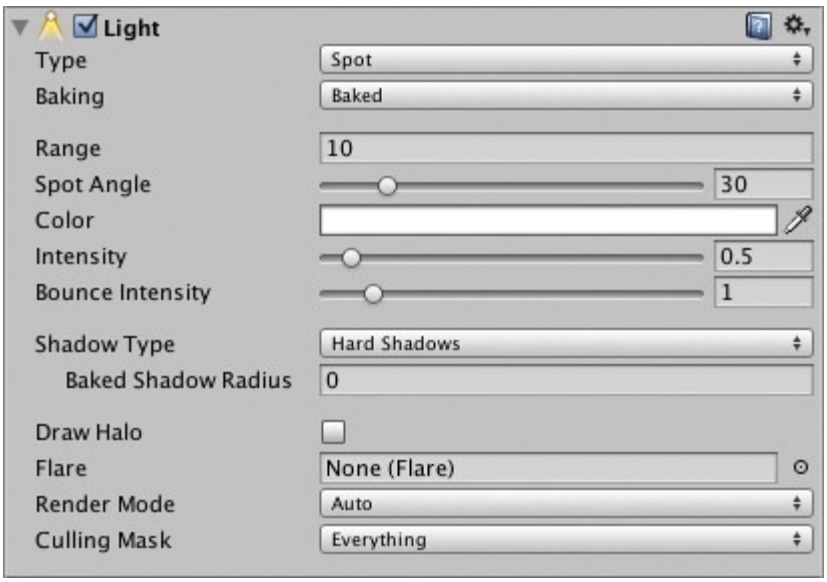

Рисунок 4.15 - Властивості компонента Light в інспектора

Просто змінюючи властивість Color (колір) джерела світла, можна додати сцені зовсім інший настрій.

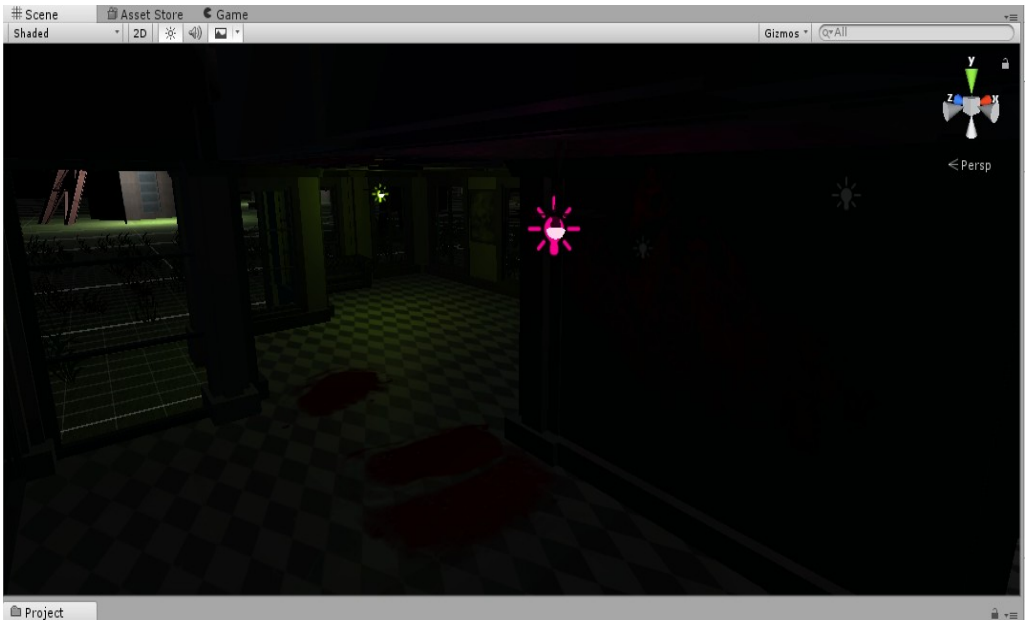

Рисунок 4.16 - Похмурі джерела світла

4.5 Створення та редагування terrain'ов

Можна додати об'єкт terrain'a в сцену, вибравши GameObject> Create Other> Terrain з меню (це також додасть відповідний Accet terrain'a в вікно Project). При цьому, ландшафт спочатку буде нічим іншим, як просто великий, плоскою рівниною. Однак, якщо подивитися на інспектор при виділеному об'єкті terrain'a, побачимо , що там представлений ряд інструментів, які можна використовувати для створення будь-яких потрібних ландшафтів.

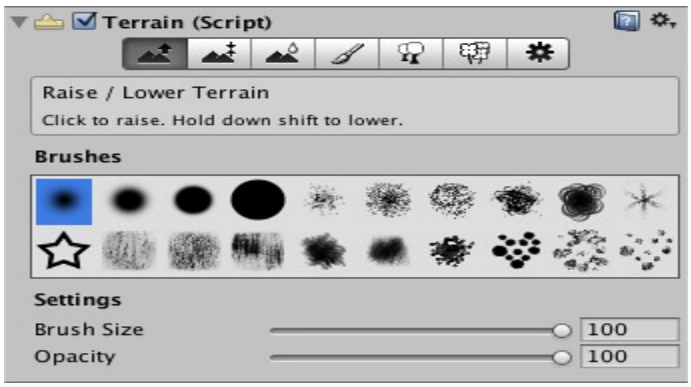

Рисунок 4.17 - Інструменти редагування terrain'a

За винятком інструменту розміщення дерев і панелі налаштувань, всі інші інструменти на панелі інструментів забезпечують набір "кистей" і налаштувань для розміру і прозорості кисті. Те що, ці кисті схожі на кисті з графічних редакторів - не випадково, тому що саме так створюються деталі у terrain'a, за допомогою їх "малювання" на ландшафті. Якщо виберати перший зліва інструмент на панелі інструментів (Raise / Lower Terrain) і пересунути миш над terrain'ом у вікні Scene, ми побачимо, що курсор нагадує прожектор на поверхні. Клікнувши кнопку миші, можна намалювати поступові зміни в висоті ландшафту в точці розташування мишки. Можна змінювати форму кисті в панелі Brushes. Опції Brush Size і Opacity впливають на область кисті і силу "натиску" відповідно.

Перші три інструменти на панелі інструментів інспектора terrain'a використовуються для малювання змін висоти на terrain.

Рисунок 4.18 - Інструменти для змін висоти

Починаючи зліва, перша кнопка активує інструмент Raise / Lower Height (підвищити / знизити висоту). Коли малюємо використовуючи цей інструмент, висота буде збільшуватися поки керуємо мишкою з затиснутою лівою кнопкою по terrain'у. Висота буде підсумовуватися, якщо будемо утримувати курсор в одному місці, аналогічно ефекту аерографії в графічних редакторах. Якщо затиснути кнопку shift, висота буде знижуватися. Можна використовувати різні кисті для створення різних ефектів. Наприклад, можна створити горбисту місцевість за допомогою збільшення висоти пензлем з м'якими краями і потім вирізаючи круті скелі і рівнини за допомогою зниження висоти пензлем з гострими краями.

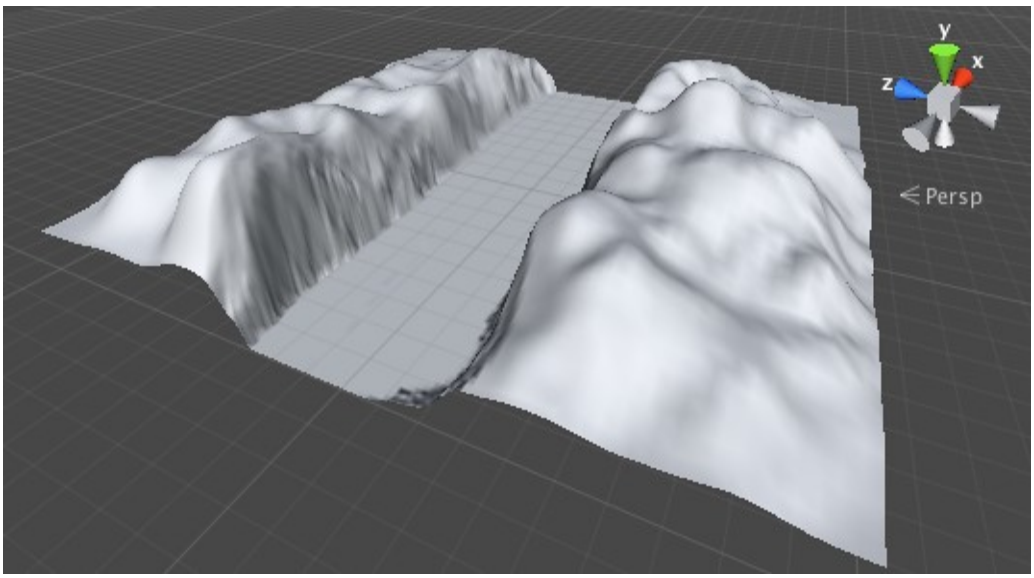

Рисунок 4.19 - Горбиста місцевість з різко виділеною рівнинною

Другий зліва інструмент, Paint Height, аналогічний інструменту Raise / Lower, тільки цього разу є додаткова властивість для установки цільової висоти. Коли малюємо на об'єкті, terrain занижуватиметься в областях вище цієї висоти і завищуватиметься в областях нижче цільової висоти. Можна використовувати слайдер властивості Height для установки висоти вручну, або клікати по terrain'у з затиснутою клавішею shift для взяття зразка висоти в поточній позиції мишки (аналогічно "піпетке" в графічних редакторах). Поряд з потрібними Height є кнопка Flatten, яка просто встановлює висоту у всього terrain'а в задане значення. Це зручно для підняття рівня землі, наприклад, якщо бажаємо щоб ландшафт включав як пагорби вище цього рівня, так і рівнини нижче нього. Paint Height зручний для створення плато в сцені і також для додавання штучних елементів, таких як дороги, платформи або ступені.

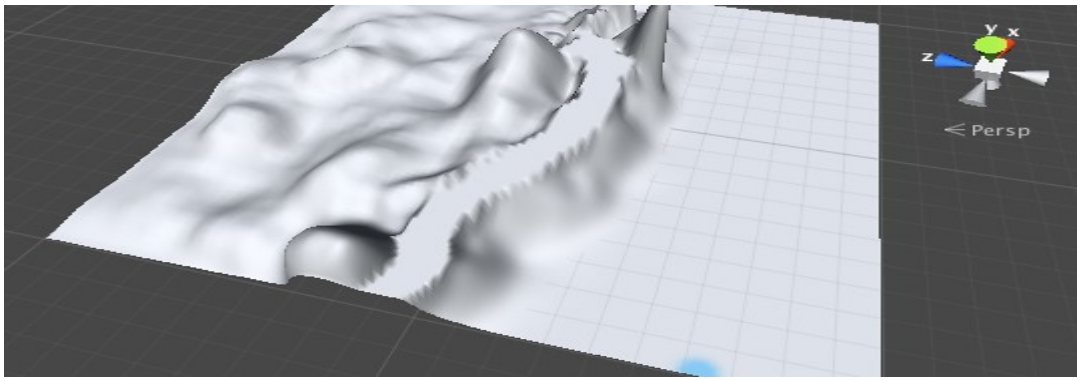

Рисунок 4.20 - Схил пагорба з плоскою дорогою

Третій зліва інструмент, Smooth Height, не піднімає або опускає значно висоту terrain'a, а скоріше, усереднює сусідні області. Це пом'якшує ландшафт і зменшує поява різких змін, щось на зразок інструменту розмиття в графічному редакторі.

4.6 Текстури

Можна додавати текстури на поверхню terrain'a для створення забарвлення і дрібних деталей. Так як terrain'и досить великі об'єкти, до них зазвичай застосовують текстури, які можна бесшовно стикувати, щоб замостити ними поверхню (повтор зазвичай не помітний з точи зору персонажа, який знаходиться близько до землі). Одна текстура буде служити як "фонова" картинка по всьому ландшафту, але також можна малювати області, використовуючи інші текстури, щоб симулювати різні покриття, такі як трава, пустеля і сніг. Намальовані текстури можуть бути застосовані з різною прозорістю, так, щоб вийшов плавний перехід між трав'янистої місцевістю і піщаним пляжем, наприклад.

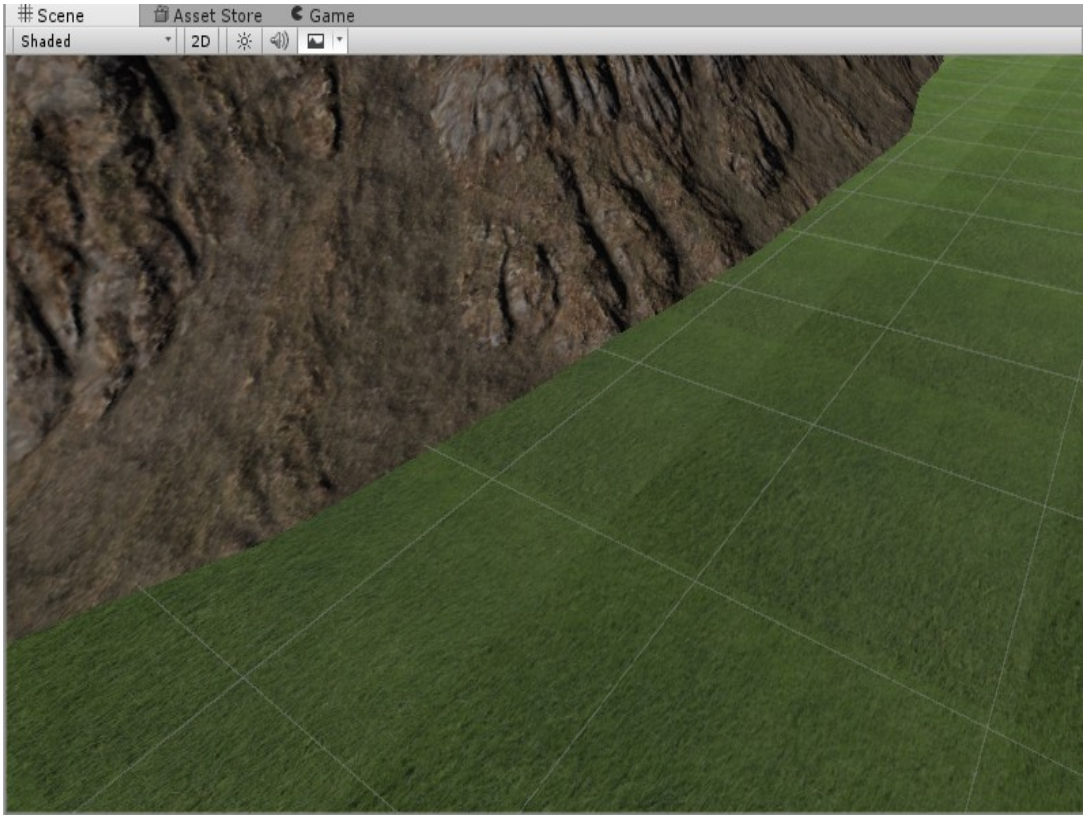

Рисунок 4.21 - Terrain с зеленою текстурою на фоні гір

Спочатку у terrain'a НЕ буде текстур, призначених для малювання. Якщо ви клацніть кнопку Edit Textures і з випав меню виберіть Add Texture, ви побачите вікно, в якому ви можете вказати текстуру і її властивості.

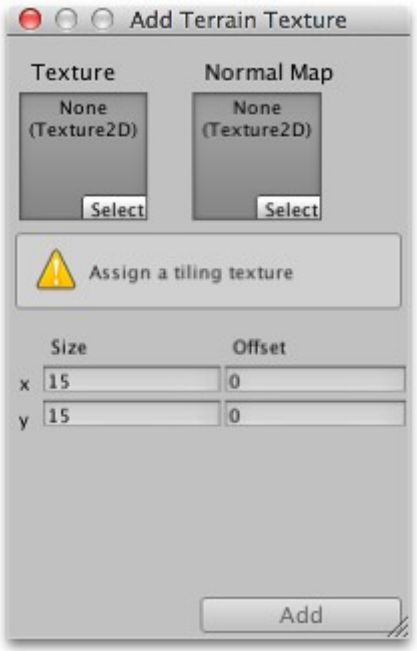

Рисунок 4.22 - Вікно Add Texture

4.7 Створення дерев

Terrain'и в Unity можуть бути доповнені деревами. Острівці з деревами можна малювати прямо на terrain'е, аналогічно малювання карт висот і текстур, але дерева - повноцінні 3D об'єкти, які ростуть з поверхні. Unity використовує оптимізації (наприклад, перетворює вилучені дерева в спрайт) для збереження гарної продуктивності рендеринга, так що можна мати щільно засаджені ліси з тисячами дерев, і зберігати при цьому прийнятну частоту кадрів.

У Unity є свій інструмент для створення дерев (Tree creator), який можна використовувати для створення нових Ассет дерев, але також можна використовувати і стандартні програми для 3D моделювання для цього завдання. Меш дерева повинен мати щонайменше 2000 трикутників (для підвищення продуктивності) і центр обертання повинен бути розташований прямо в підставі дерева, там, де воно стикується з землею. Меш завжди повинен мати рівно два матеріали, один для стовбура і гілок, інший для листя.

Дерева повинні використовувати шейдери Nature / Soft Occlusion Leaves і Nature / Soft Occlusion Bark. Щоб використовувати ці шейдери, потрібно помістити дерево в спеціальну папку, яка містить в імені "Ambient-Occlusion". Коли помістімо модель в цю папку і повторно імпортуємо її, Unity прорахує м'яке розсіяне затінення спеціально створеним для дерев способом. Шейдери "Nature / Soft Occlusion" спираються на угоду про іменування папок, і дерево не отмалюється коректно, якщо не будемо його дотримуватися.

Після збереження Ассет дерева з пакета для 3D моделювання, буде потрібно натиснути на кнопку Refresh (відображається в інспектора, коли обраний інструмент малювання дерев) для того, щоб побачити оновлені дерева на terrain'е.

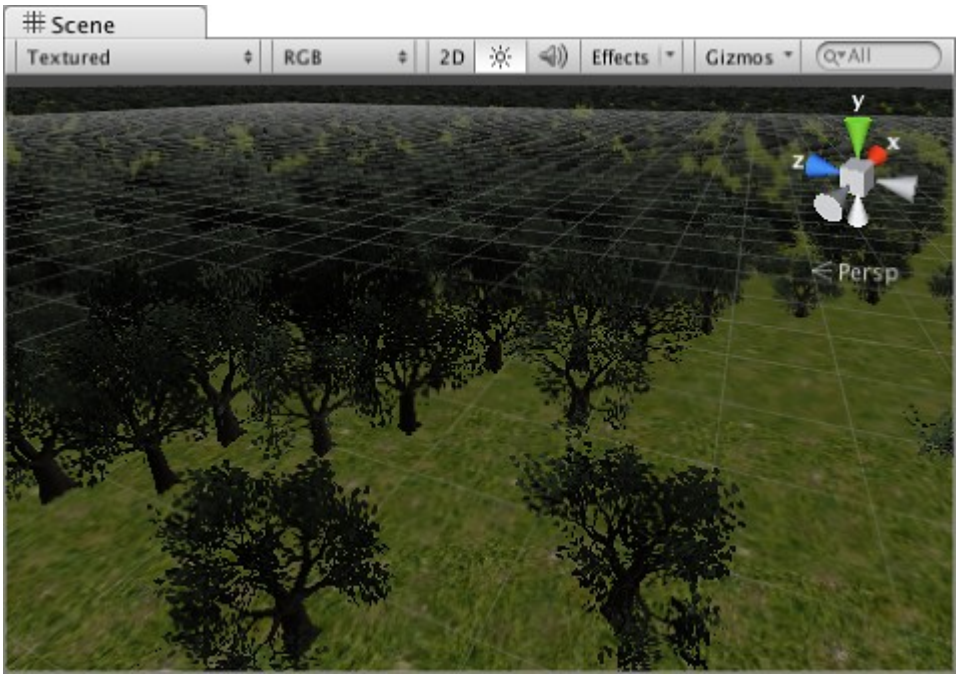

Рисунок 4.23 - Terrain з деревами

ВИСНОВКИ

Дослідження предметної області, безумовно, є важливим, особливо в тому випадку, якщо предметну область потрібно проаналізувати та виявити недоліки та переваги.

Unity 3D виконує великий обсяг роботи. В основному це налаштування об'єктів на сцені та іх взаємодія між собою.

Була створена "Survival Horror" гра для реалізації якоі був виконаний аналіз існуючих ігрових движків та графічних 3D редакторів.

За допомогою графічного редактору Blender 3D було створено більш ніж 20 ігрових об' єктів власноруч, для використання їх у проекті.

Основною розробкою гри була сцена в Unity 3D яка скалається з більш ніж 100 ігрових об'єктів створених у Blender і взаємодією між ними.

ПЕРЕЛІК ПОСИЛАНЬ

1. Большаков Д. І. 3D моделювання. / Д. І. Большаков - Техатека, 2011. - 34 с.

2. Бочков М. Д. Основи 3D-моделювання. / М. Д. Бочков - СПб. : Пітер, 2003. - 106 с.

3. Гембейк Е. В. Ігрова індустрія. Створення ігор. / Е. В. Гембейк - М. : 3DNs, 2007. - 412 с.

4. Дацко М. А. Моделювання складних об'єктів. / М. А. Дацко - М. : Максімас, 2015. - 111 с.

5. Хокінг Д. М. Unity в дії. Мультиплатформенна розробка на. / Д. М. Хокінг, 2016. - 336 с.

6. Кроніестер Д. Б. Основи Blender навчальний посібник 4-е видання. / Д. Б. Кроніестер - Blender Basics 2.6, 2012. - 416 с.

7. Максимов А. П. Створення найпростіших моделей, побудова сцени. / А. П. Максмов - М. : Мітра, 2011. - 38 с.

8. Мейкісон Л. К. Моделювання - це легко. / Л. К. Мейкісон - М. : 3DNs, 2013. - 313 с.

9. Петренко. С. М. Вивчаємо Blender 3D. / С. М. Петренко Blend. - М. : 3DNs, 2009. - 542 с.