МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК, СЕКЦІЇ ІКТ

ВИПУСКНА РОБОТА

на тему:

«Гра на платформі Unity»

Завідувач випускаючої кафедри **Довбиш А.С.**

Керівник роботи **Бабій М.С.**

Студента групи ІН-64-8 **Зарицький Б.В.**

СУМИ 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК, СЕКЦІЇ ІКТ

Затверджую_____

Зав. кафедрою Довбиш А.С.

“_____” _____2020 р.

ЗАВДАННЯ
до випускної роботи

Студента четвертого курсу, групи ІН-64-8 спеціальності “Інформатика” денної форми навчання Зарицького Богдана Віталійовича.

Тема: “ Гра на платформі Unity ”

Затверджена наказом по СумДУ

№_____ от _____2020 р.

Зміст пояснювальної записки: 1) огляд існуючих рішень; 2) вибір методу рішення; 3) програмна реалізація.

Дата видачі завдання “_____”_____2020 р.

Керівник випускної роботи _____ Бабій М.С.

Завдання прийняв до виконання _____ Зарицький Б.В.

РЕФЕРАТ

Записка: 33 сторінок, 26 рисунків, 6 джерел.

Об’єкт дослідження — Гра на платформі Unity.

Мета роботи — Реалізація 3D гри на основі Unity.

Методи дослідження — аналіз предметної галузі, постановка задачі, тестування розробленої програми.

Результати — Результатом виконання даної роботи є створена гра на платформі Unity та розроблені моделі для неї. Даний ресурс має зручний функціонал та задовольняє всім вимогам, які ставилися на етапі постановки завдання.

UNITY, INDI GAME, GAME ENGINE, 3D PLATFORMER, DEVELOPER, UNITY GAME.

ЗМІСТ

ВСТУП

Unity / Game Developer-це розробник, який створює ігри, а також бізнес-додатки, широко використовувані для рекламних кампаній. Game-розробники зайняті в повному циклі життя гри: створення, тестування, доопрацювання, підтримка, оновлення, модифікація і т.д. за допомогою Unity все частіше створюються великі онлайн-ігри. Завдання розробників полягає в тому, щоб грамотно створити клієнтську частину гри.Одним з основних переваг Unity є можливість створення кроссплатформенних продуктів. Такі величезні компанії, як Intel і Microsoft з випуском своїх нових пристроїв також випускають бібліотеки під Unity для того, щоб зробити розробку під пристрої цих компаній більш простою і зручною.Для того щоб стати розробником ігор на Unity, то вам необхідно добре знання мови C#. Якщо ваша мета-створення простих ігор, то досить буде і базових знань програмування. Але при розробці мережевих складних ігор від розробників потрібно також знання патернів проектування і розуміння того, яким чином буде працювати клієнт з сервером. Величезним плюсом є знання 3D редакторів, таких як 3DMax, Blender, Maya або інших.Розробкою ігор може займатися як одна людина, так і колектив розробників. Часто розробка фінансується компанією-видавцем, яка також бере на себе завдання піару і маркетингу. Розробка великобюджетних ігор зазвичай має на увазі велику чисельність команд розробників і тривалі терміни реалізації проекту. Навички роботи в команді і комунікабельність для розробника не менш важливі, ніж теоретичні знання і його навички.

1. Огляд існуючих рішень

Існує багато компаній гігантів які хочуть від розробки ігор лише прибуток, вони володіють стратегією росту популярності , яку розробляли багато років, персоналом, який може враховувати тисячі співробітників і все ж є люди, які одним своїм бажанням піднятися на рівень до них розробляють ігри на безкоштовних рушіях, які майже такого ж рівня як і популярні ігри валиких компаній, деякі з них виходять на великий ринок.

1.1 Існуючі рішення

Superhot

Оригінальний інді-шутер з видом від 1-ї особи, створений однойменною командою Superhot Team. Будучи невеликою демонстрацією для конкурсу 7 Day FPS Challenge, гра переросла в демоверсію гри для браузерів, а потім перебралася на Kickstarter, де отримала необхідне фінансування і згодом стала повноцінною грою.

Рисунок 1.1 - Superhot

HearthStone

Мабуть, найвідоміша ККІ в світі, по якій вже давно проводяться чемпіонати і змагання зі значними грошовими призами. Карткова битва між героями знаменитої всесвіту, створеної компанією Blizzard, тягне мільйони користувачів, і на даний момент ігрова база акаунтів становить близько декількох десятків мільйонів учасників

Рисунок 1.2 - HearthStone

Ori and the Blind forest

Це 2D платформер з приголомшливою графікою і дуже зворушливим сценарієм. Проект здобув любов геймерів і створив цілу армію шанувальників для майбутніх проектів компанії Moon Studios, які вже зараз працюють не тільки над продовженням пригод забавного лісового духу, але і планують запустити кілька ігор на Unity3D, відмінних від флагмана організації.

Рисунок 1.3 – Ori And the Blind Forest

INSIDE

Ще один платформер, цього разу злегка лякає. Маленький хлопчик бродить по чарівному світу і бореться з якимось злом, бажаючим попрацювати людей. Компанія Playdead вже запустила в мережу вельми успішну знамениту гру під назвою «Limbo», успіх якої дозволив розробникам зайнятися розвитком власного виробництва і задіяти доступне обладнання для представлення шанувальникам ще більш опрацьованих синглів.

Рисунок 1.4 - Inside

RUST

Проект Rust довго знаходився в ранньому доступі (з 2013 року), що по суті зараховувало її в проклятий стан «альфа-ігор», яким ніколи не судилося вийти. Тим не менш, гра вже так давно оновлюється і обросла такою кількістю контенту, що в ранньому доступі вона перебувала швидше за примхою розробників, ніж через технічні причин. Нарешті, 8 лютого 2018 гра все ж вийшла офіційно.

Рисунок 1.5 - Rust

FIREWATCH

Гра на Unity3D під назвою "Firewatch" – це дітище колишніх співробітників компаній Telltale, 2k і Double Fine, які створили новий бренд-незалежну студію Campo Santo. Дивно, але один з перших проектів компанії завоював величезну популярність, зібрав купу нагород і титулів, завдяки дивовижній графіці, придатному сюжеті і гранично зрозумілому інтерфейсі.

Рисунок 1.6 - Firewatch

POKEMON GO

Проект, який влаштував по собі вакханалію "хайпа" в 2016 році, яка захлеснула весь інтернет, проникла на телебачення і в газети. Pokemon Go є безкоштовною грою для мобільних пристроїв, в якій гравцеві необхідно колекціонувати відомих персонажів-монстрів з всесвіту Pokemon.

Ключова особливість гри будується на технології доповненої реальності, яка за допомогою Інтернет-з'єднання і відеокамери визначає місце розташування користувача і проектує на екрані самих покемонів, які в цей час нібито знаходяться в реальності.

Рисунок 1.7 – Pokemon GO

1.2 Постановка задачі

Метою роботи є реалізація 3D гри на основі Unity, що буде виконувати такий список функцій:

-Функціонування та відображення меню. Переключення між пунктами меню.

-Створення грального рівня.

-Пересування персонажа по рівню. Функція дозволяє вільно ходити, бігати, пригати та присідати.

-Unity автоматично додає Collider до своїх об’єктів Функція обробки зіткнень з іншими об’єктами, функція пересування по гральному полю.

-Відображення усіх об’єктів на полі. У верхній частині вікна виводиться статистика про стан гри.

-Реалізація функцій для ведення статистики. Функцію підрахунку кількості балів гравця, яка обчислює кількість балів, яку отримує він за сбір гемів , а також перехід на наступний рівень.

Надання можливості користувачу зберегти поточний стан гри при виході в головне меню.

2. ВИБІР МЕТОДУ РІШЕННЯ

На Unity зроблено багато чудових ігор, головною перевагою Unity перед іншими двигунами є його простота для одиночної розробки. Не потрібно мати цілу компанію девелоперів, щоб зробити хорошу гру. Якщо ти один або маєш невелику команду і хочеш зробити хорошу гру без претензій на AAA, то Unity стане кращим вибором. Проте, навіть великі корпорації часто вибирають для своїх ігор саме Unity.

Найсильнішими сторонами є в Unity3D:

- простота розробки,

- зручний інструментарій,

- кроссплатформенность,

- багата документація,

- величезна спільнота.

Для створення гри нам даються на вибір 2 мови програмування-C # Unit UnityScript.

Движок підтримує безліч популярних форматів, таких як:

1) .3ds, .максимум, .параметр obj,.у FBX, .Де, Ма,...мегабайт, .суміш для тривимірних моделей;
2) .MP3,.Огг, .найближчим часом, .wav, .мода .це, .см3 для звукових файлів ;
3) .ПСД .формат JPG, .формат PNG, .файл GIF, .формат BMP, .ТГА,.незлагода, .МКФ, .пикт, .dds для зображень;
4) .мова,.Аві, .АЧС,.миль на галон,.mpeg,. mp4 для відеофайлів.
5) .формат txt, .htm, .формат HTML, .XML, .байт для тексту

Жанр, який можливо вибрати обмежується лише фантазією. Можна створювати і РПГ, і стратегії, і Слешери.

Unity підтримує наступні ОС Windows, Лінукс, Макос, на SteamOS, Android, прошивкою, Windows телефон, ігрову приставку PlayStation4, Xbox один, WebGL і Oculus Ріфт і багато інших. Повний список можна знайти на офіційному сайті. Таким чином, працюють ігри на Unity на десктопах, на

смартфонах, планшетах, приставках, в браузерах, ВР-окулярах і деяких інших системах.

Рисунок 2.1 – Unity

Також Вам надаються Kit'и.

Кіт - це набір скриптів і префабів, а часто і графічних елементів для гри. Вони покликані полегшити розробку гри певного жанру і як правило розбиті за жанрами (action-RPG Starter Kit, RTS Starter Kit, 3D Shooter Starter Kit, Space Game Starter Kit, VR Starter Kit і так далі). Також бувають стартер кити різних ігрових елементів, не пов'язаних з геймплеєм (Nature Starter Kit з додатковими природними об'єктами, Medieval Starter Kit з середньовічними об'єктами і так далі). По суті, стартер кити виконують в розробці гри ту ж роль, що і фреймворки в програмуванні. Однак варто відзначити, що використання геймплейного стартер кита примушує розробника вивчати велику кількість чужого коду і чужої структури для внесення своїх змін і повноцінного використання. У зв'язку з цим більшість розробників вважає за краще писати майже все з нуля, отримуючи повне розуміння роботи своєї гри.

Хочу звернути увагу на декількох моментах, а саме язик програмування, Кіти та программи для моделювання.

Щоб писати скрипти, необхідний редактор коду. У комплекті з Unity йде MonoDevelop, так що його не потрібно встановлювати окремо. Інший варіант-використовувати Visual Studio, я використовував другий варіант. Програмування в Unity називається скриптингом.

Скриптинг говорить нашим об'єктам GameObjects, як себе вести; ігровий процес створюють скрпіти і компоненти, прикріплені до GameObjects, а також їх взаємодія один з одним. Скриптинг в Unity відрізняється від чистого програмування. Якщо ви займалися чистим програмуванням, наприклад створювали працюючий додаток, вам слід зрозуміти, що в Unity вам не потрібно писати програмний код, що виконується додатком, так як Unity робить це за вас. Замість цієї рутинної роботи ви фокусуєтеся на ігровому процесі, що задається скриптами. До Об'єкта GameObject, що знаходиться в сцені, повинен бути приписаний скрипт, щоб викликатися Unity. Скрипти пишуться на спеціальній мові, зрозумілій движку Unity. На цій мові ми можемо взаємодіяти з движком і віддавати йому свої команди.

Мова, що використовується в Unity, є C# . Всі мови, з якими Unity має справу, є об'єктно-орієнтованими скриптовими мовами. Подібно до будь-якої мови, скриптові мови мають синтаксис (як би частини мови), і первинні елементи цієї мови називаються змінними, функціями і класами.

Asset Store - доступний великий вибір ассетів-текстури, моделі та анімації, приклади проектів, підручники та розширення для редактора. Асети доступні через простий інтерфейс, вбудований в редактор єдності, через нього можна завантажити і імпортувати безпосередньо в ваш проект.

Рисунок 2.1 – Asset Store

Программи для моделювання , які я використовував такі як Blender, Zbrush.

Blender - професійне вільне і відкрите програмне забезпечення для створення тривимірної комп'ютерної графіки, що включає в себе засоби моделювання, скульптингу, анімації, симуляції, рендеринга, постобробки і монтажу відео зі звуком, компонування за допомогою «вузлів» (Node Compositing), а також створення 2D-анімацій. В даний час користується великою популярністю серед безкоштовних 3D-редакторів у зв'язку з його швидким стабільним розвитком і технічною підтримкою.

Рисунок 2.3 – Blender

ZBrush - програма для 3D моделювання, створена компанією Pixologic. Відмінною особливістю даного ПЗ є імітація процесу "ліплення" тривимірної скульптури, посиленого движком тривимірного рендеринга в реальному часі, що істотно спрощує процедуру створення необхідного тривимірного об'єкта. Кожна точка (називається піксель) містить інформацію не тільки про свої координати XY і значеннях кольору, але також і глибині Z, орієнтації і матеріалі. Це означає, що ви не тільки можете «ліпити» тривимірний об'єкт, але і «розфарбувати» його, малюючи штрихами з глибиною. Але вам не доведеться малювати тіні і відблиски, щоб вони виглядали натурально — ZBrush це зробить автоматично. Також швидко працює зі стандартними 3D об'єктами, використовуючи Кисті для модифікації геометрії матеріалів і текстур. Дозволяє домогтися інтерактивності при великій кількості полігонів. Використовуючи спеціальні методи, можна підняти деталізацію до десятків мільйонів полігонів. Також є безліч модулів, що підключаються (робота з текстурами, геометрією, безліч нових кистей, швидка інтеграція з професійними пакетами 2D графіки і багато іншого)

3. ПРОГРАМНА РЕАЛІЗАЦІЯ

Перед тим як починати реалізацію з програмування, потрібно визначитись з концептом гри.

Концепція ігрового проекту - це документ, що описує цілі, завдання, основні особливості проекту, дослідження ринку і цільової аудиторії, умови його виконання. Також, так як проект ігровий, обов'язково опис ігрової механіки, ігрових понять, приблизний сценарій і концепт-арт. Якщо ви ще розраховуєте робити проект не поодинці (що досить ймовірно), то знадобиться ще технічне завдання (ТЗ) – документ, що містить опис необхідних робіт, терміни та умови.

Після осмислення концепції йде етап контента. Під контентом розуміється весь вміст гри, з яким взаємодіє Користувач. Це графіка (растрова, векторна, 3D), музичний і звуковий супровід, відеоряд, сценарій і текст. Також сюди слід додати медіаматеріали, використовувані для просування гри (реклама, банери та інші). Зазвичай контент для гри розробляють під час програмування для того щоб не затримувати процес розробки.

Як не дивно, створення програмного коду для ігор не є найскладнішим завданням, але в той же час і не є простою.

Для початку я визначився з концептом, це має бути гра без насильства, поріг входження має бути низьким. На думку пришла гра платформер де гравцю давали виклик нелегкий левел дизайн та різноманітні рівні.

Canvas - це область, в якій повинні знаходитися всі елементи інтерфейсу користувача. Canvas - це ігровий об'єкт з компонентом полотна на ньому, і всі елементи інтерфейсу користувача повинні бути дочірніми елементами такого полотна.

Створення нового елемента інтерфейсу користувача, наприклад зображення за допомогою меню GameObject > UI > Image, автоматично створює канвас, якщо в сцені ще немає канваса. Елемент інтерфейсу користувача створюється як дочірній елемент цього полотна.

Область канвас відображається у вигляді прямокутника у вигляді сцени. Це дозволяє легко позиціонувати елементи користувальницького інтерфейсу без необхідності мати вигляд гри видимим в будь-який час.

Canvas використовує Об'єкт EventSystem для допомоги системі обміну повідомленнями.

У меню наявна можливість почати гру та вийти з неї(Рисунок 3.1)

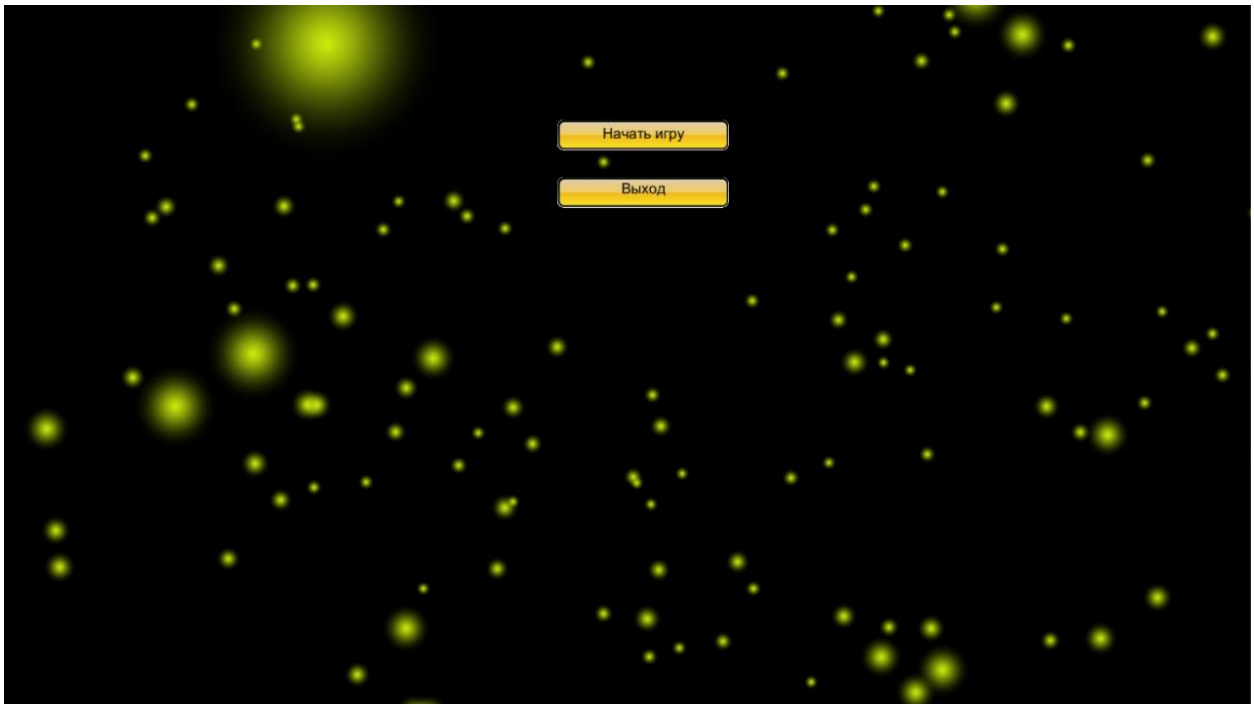

Рисунок 3.1 – Головне меню

При заході в гру вас зустрічає ваш персонаж та виклик зібрати всі сфери та дійти рівень до кінця (Рисунок 3.2).

Рисунок 3.2 – Початок рівня

Контролер аніматора дозволяє організувати і підтримувати набір анімаційних кліпів і пов'язаних з ними анімаційних переходів для персонажа або об'єкта. У більшості випадків це нормально мати кілька анімацій і перемикатися між ними, коли виникають певні умови гри. Наприклад, ви можете перемикатися з анімаційного кліпу walk на анімаційний кліп jump щоразу, коли натискається пробіл. Однак навіть якщо у вас є тільки один анімаційний кліп вам все одно потрібно помістити його в контролер аніматора щоб використовувати його на GameObject. (Рисунок 3.4)

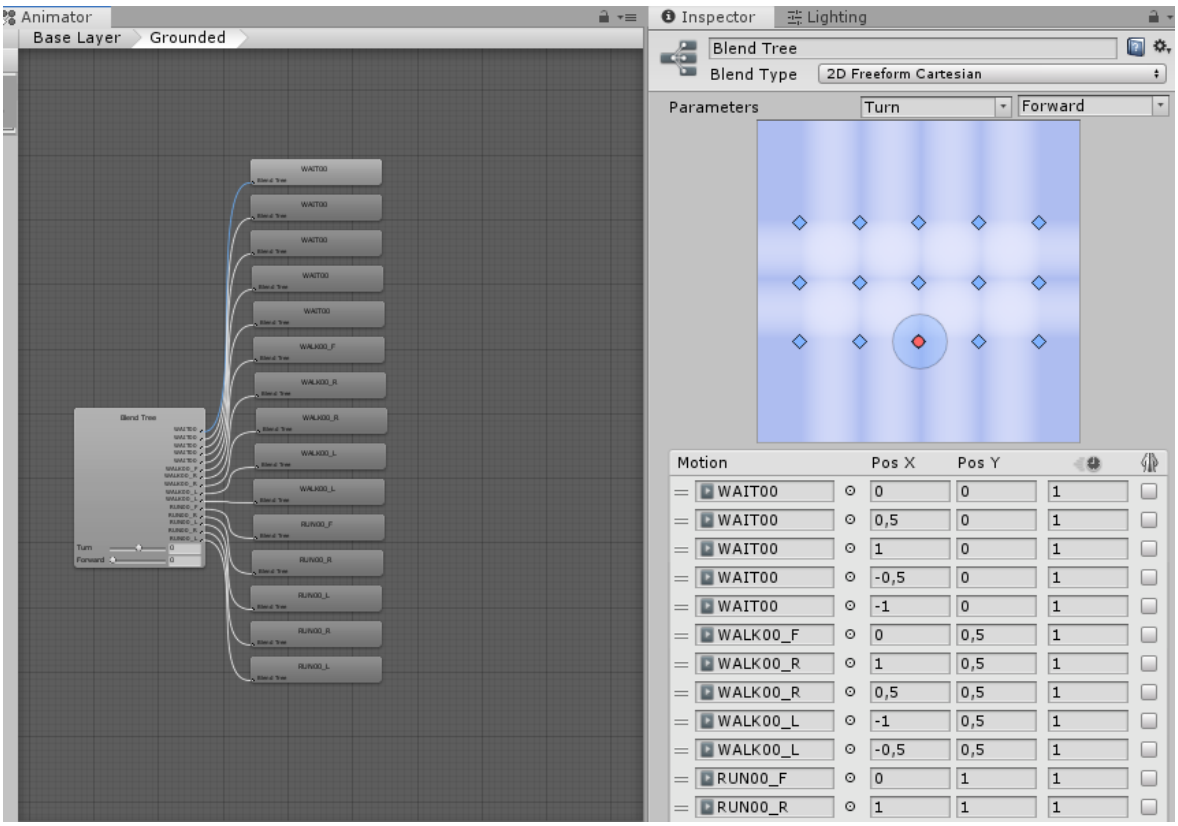

Рисунок 3.3– Анімації гравця

Audio Source (Джерело звуку) відтворює Audio Clip в сцені. Якщо Audio Clip є 3D кліпом, джерело програється в заданому положенні в просторі і буде приглушуватися в залежності від відстані. Аудіо може бути розподілено по колонках (наприклад, з стерео в 7.1) за допомогою властивості Spread і трансформуватися між 3D і 2D за допомогою властивості PanLevel. Можна контролювати залежність цих ефектів від відстані за допомогою кривих загасання. Також якщо слухач знаходиться в одній або декількох зонах реверберації, то до джерела застосовуються реверберації (тільки для Unity Pro). Для збагачення аудіо ряду, до джерела можна застосовувати окремі аудіо фільтри. (Рисунок 3.5)

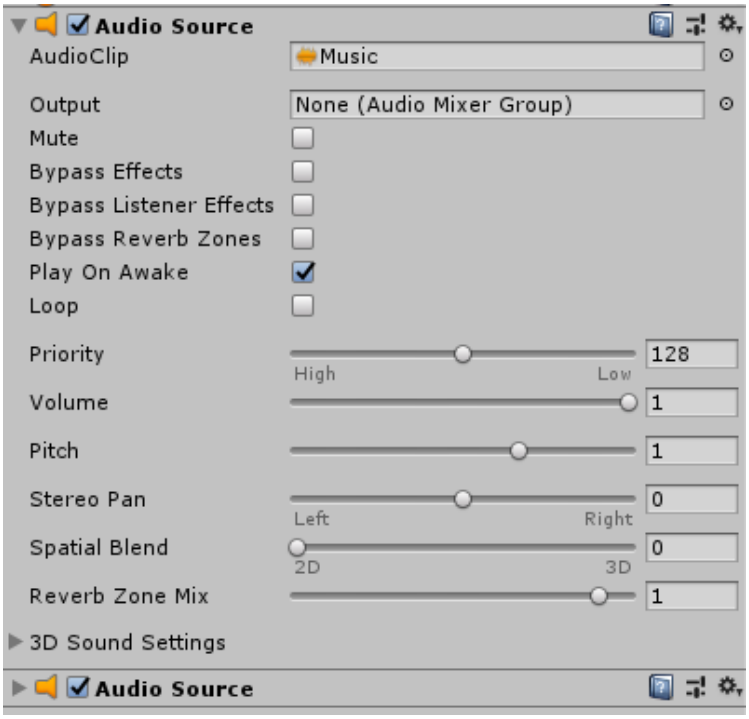

Рисунок 3.4– Джерело звука

Платформи були створені за допомогою додатка ProBuilder.

ProBuilder - це унікальний гібрид інструментів 3D-моделювання та проектування рівнів, оптимізований для побудови простої геометрії, але здатний до детального редагування та розгортання в міру необхідності. ProBuilder дозволяє швидко створювати прототипи структур, складних об'єктів рельєфу, транспортних засобів і зброї; або дозволяє створювати користувальницькі геометрії зіткнень, тригерні зони або навігаційні сітки. (Рисунок 3.5)

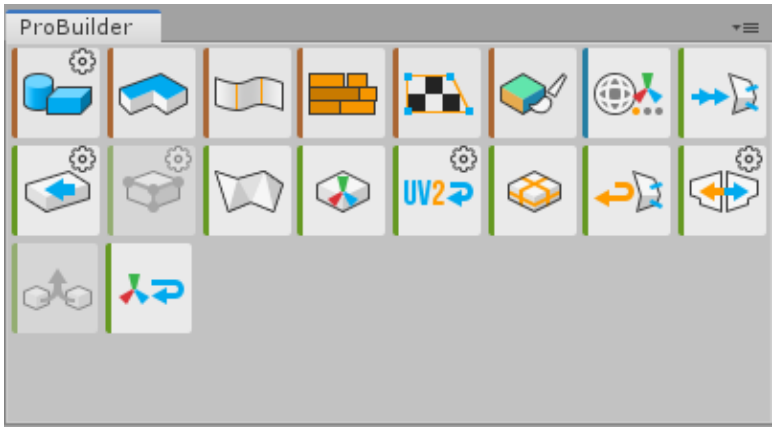

Рисунок 3.5 – Панель ProBuilder

Підказки у вигляді тексту в повітрі створені за допомогою Text Mest та зникаючі за допомогою маніпуляціям з Box Collider(Рисунок 3.6)

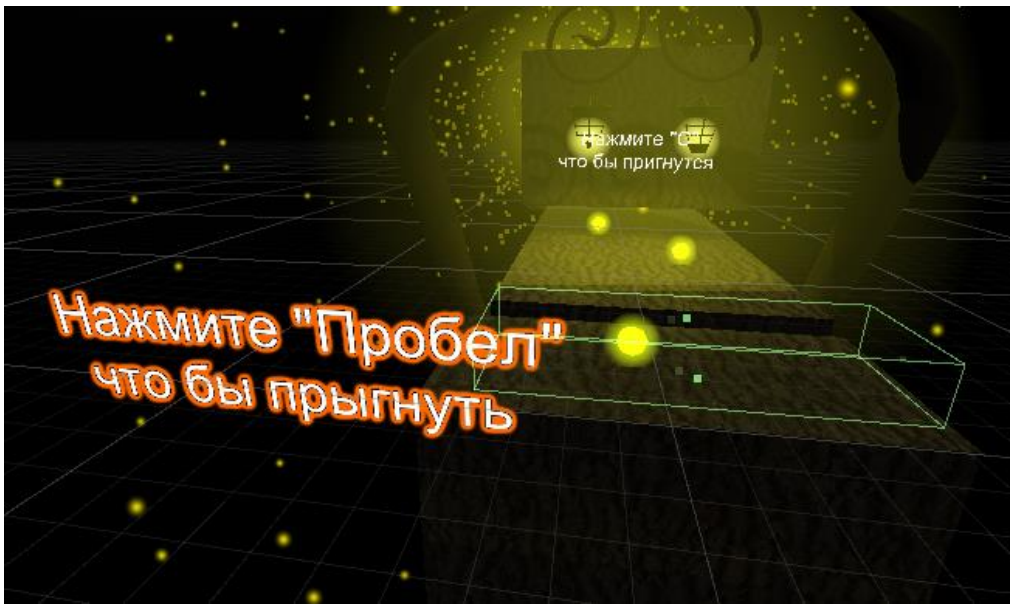

Рисунок 3.6- Підказки

Матеріали використовуються в поєднанні з сітчастими візуалізаторами , система частинок та інших способів.(Рисунок 3.7)

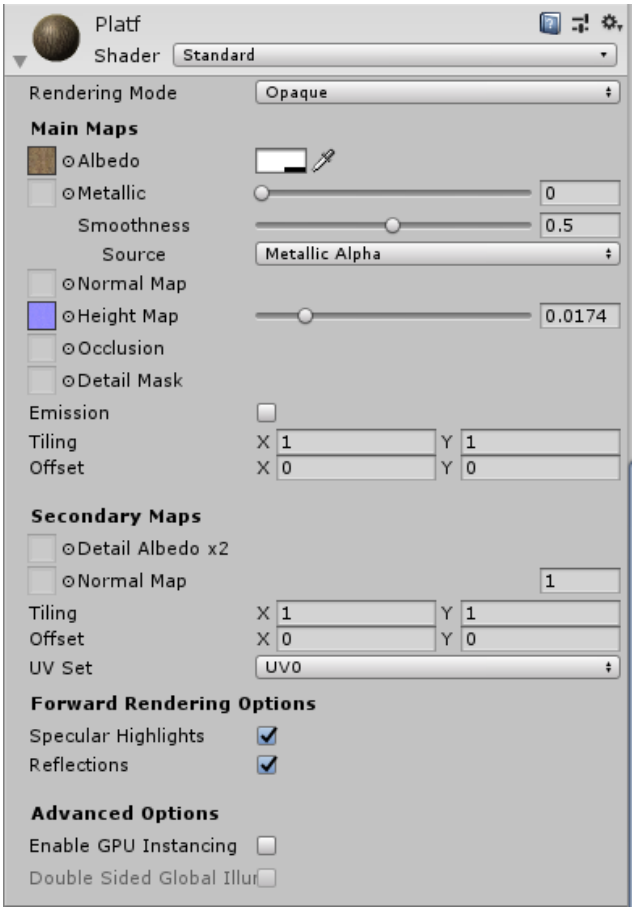

Рисунок 3.7 – Налаштування матеріала

Властивості, якими володіє інспектор матеріалу визначається шейдером що використовує матеріал. **Шейдер** - це спеціалізована графічна програма, яка

визначає, як текстура і світлова інформація об'єднуються для створення пікселів побудованого об'єкта GameObject на екрані.

Компоненти, що використовуються в Unity відіграють важливу роль у визначенні способу відображення вашого об'єкта.

Unity UI - це інструментарій користувальницького інтерфейсу для розробки користувальницьких інтерфейсів для ігор і додатків. Це система інтерфейсу користувача на основі GameObject, яка використовує компоненти та представлення гри для організації, позиціонування та стилю користувацьких інтерфейсів.

Якшо підібрати сферу то в лівому верхньому куті екрана ви побачете UI рахунок, який змінився з 0 на 1 (Рисунок 3.8).

Рисунок 3.8 – Рахунок

Якщо гравець уже грав в гру то рахунок продовжиться з останнього місця збереження (Рисунок 3.9).

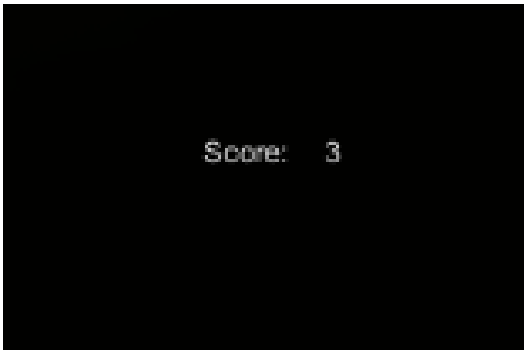

Рисунок 3.9 – Повторний запуск

Впавши за платформи ваш персонаж буде переміщений на початок рівня але дібравшись до пункту збереження ви скоротаєте час проходження рівня

(результат на рисунку 3.10).

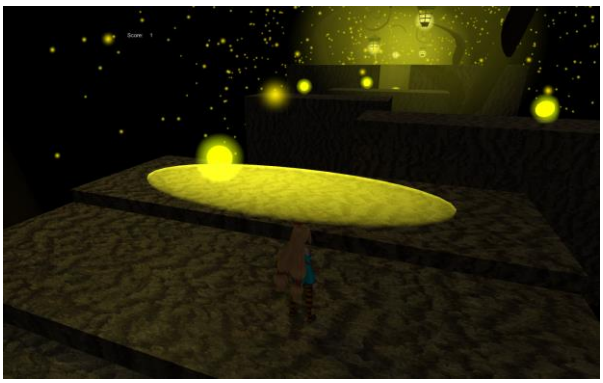

Рисунок 3.10 – Місце збереження позиції

Дібравшись кінця рівня на вас буде чекати перехід на наступний рівень (результат на рисунку 3.11).

Рисунок 3.11 – Місце завершення рівня

Кожен наступний рівень відрізнятиметься від попереднього. Зіткнувшись з об’єктом пастка бали з вашого рахунку віднімуться , а ви повернетесь на початок рівня (результат на рисунку 3.12).

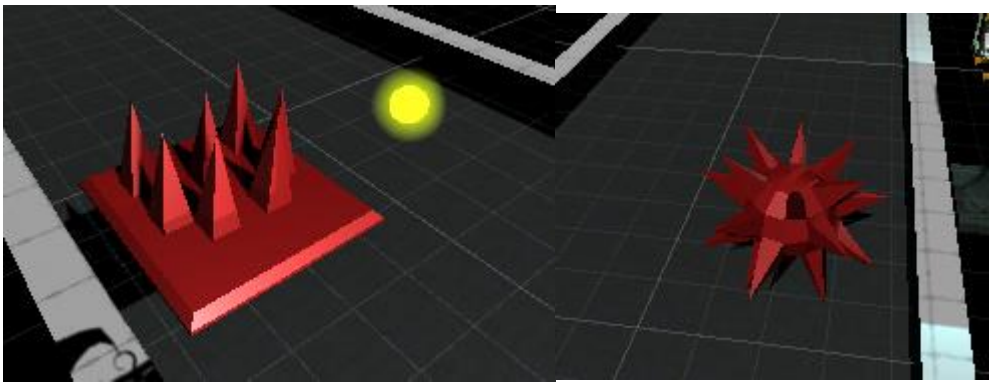

Рисунок 3.12 – Пастка

У 3D грі більшість контенту (персонажі, оточення, об'єкти) представлені у вигляді мешей, в той час як в 2D грі для цього використовуються спрайти. Меш

і спрайт ідеально підходять для відображення" цілісних " об'єктів з чітко визначеною формою. Однак, в іграх зустрічаються і інші елементи, які за своєю природою не мають чіткої форми і змінюються в реальному часі, тому їх важко відобразити за допомогою мешів і спрайтів. Для ефектів у вигляді поточних рідин, диму, хмар, полум'я і магічних заклинань слід застосовувати інший підхід, відомий як системи частинок, які дозволяють відобразити всі мінливість і енергетику таких ефектів. Система частинок зазвичай випускає частинки в випадкових положеннях в заздалегідь визначеному просторі, яке може мати форму сфери або конуса. Система сама визначає час життя частинки, і коли цей час закінчується, система знищує частинку.

Одна хороша річ про системи частинок полягає в тому, що вони є компонентами, які ви можете додати до будь-якого ігрового об'єкта в сцені. Хочете, щоб ваші акули випромінювали лазери зі своїх очей? Просто додайте систему частинок в ігровий об'єкт shark eye, і Ваші акули будуть Нестримні(Рисунок 3.13)

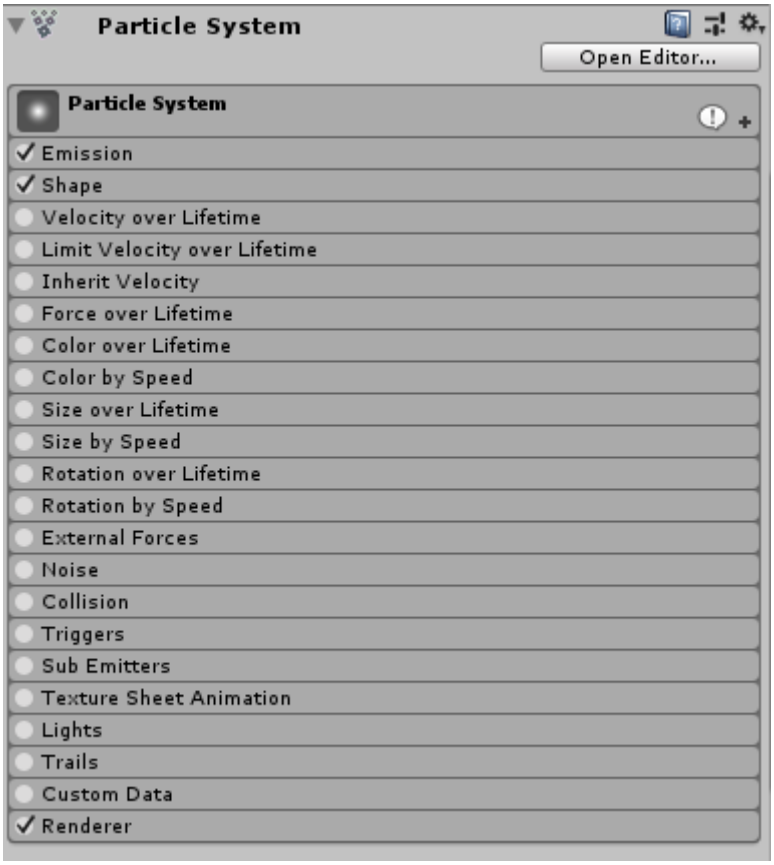

Рисунок 3.13 – Система часток

4. СТВОРЕННЯ БІЛДА UNITY

Вікно налаштування збірки містить всі налаштування і параметри, необхідні для створення білда на різні платформи. З цього вікна можна створити білд розробника (development build) щоб протестувати ваш додаток, а також зробити фінальний білд який можна буде публікувати в магазини додатків. (Рисунок 4.1)

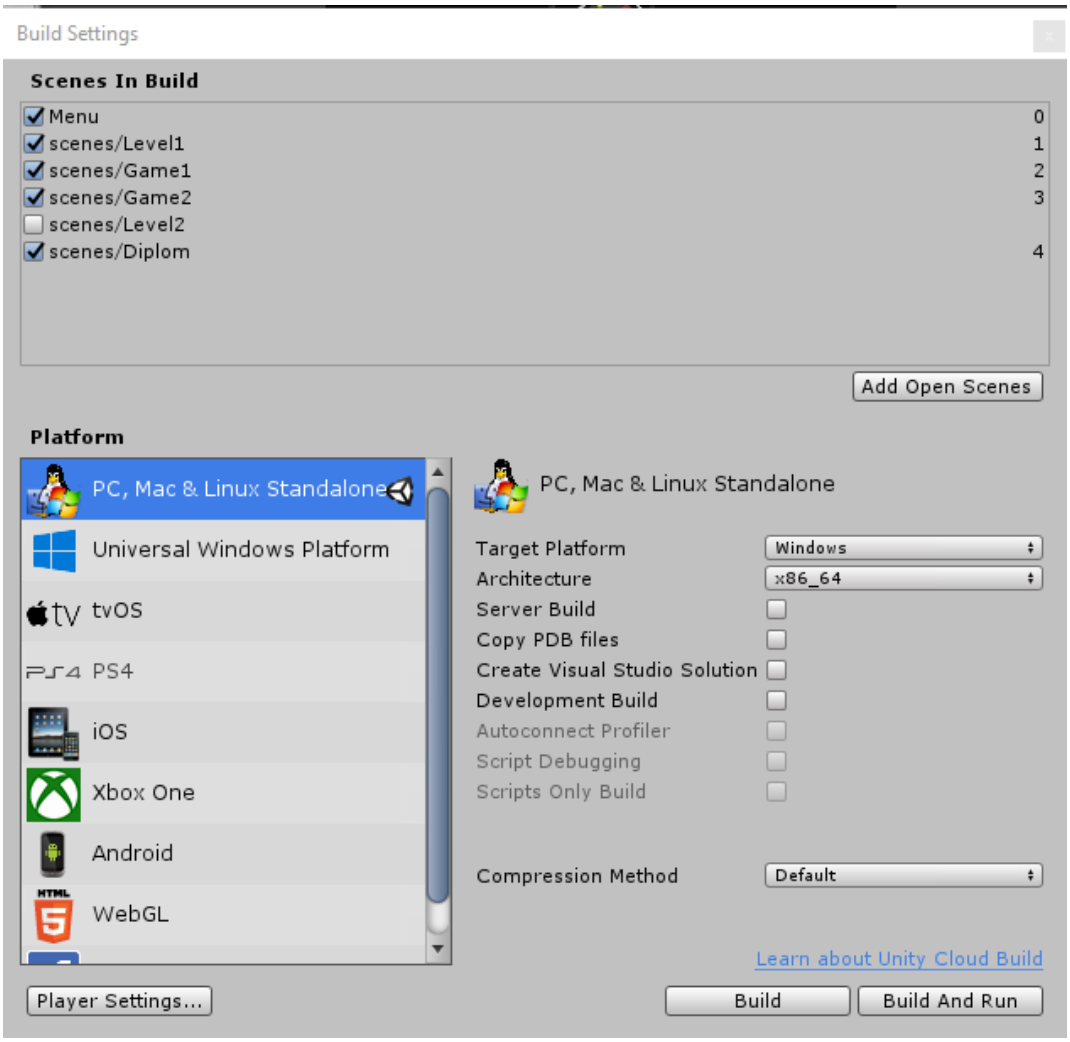

Рисунок 4.1 – Настройки білда

Є ряд параметрів для створення білда :

1) **Target Platform** - це платформа на яку буде зроблений білд

Architecture - як Ви знаєте існують 32х розрядні і 64х розрядні процесори, так ось за допомогою цієї настройки ми можемо вибрати на яку розрядність робити білд

2) **Server Build** - при перемиканні даного чекбокса Unity створить білд для сервера, який не має графіки, а також буде використовувати платформозалежний визначник UNITY_SERVER для коду. Якщо ми з такою галочкою зробимо білд для Windows він буде працювати як консольне додаток.
3) **Copy PDB files** - при включенні даного чекбокса Unity при білді на ПК включить в нього PDB файли. Файли PDB містять інформацію про налагодження програми, яка корисна для налагодження, але при цьому вона збільшує розмір вашого білда.
4) **Create Visual Studio Solution**-даний чекбокс при включенні створить файли рішення для вашого проекту, щоб ви могли створити остаточний виконуваний файл в Visual Studio.
5) **Development Build**-даний чекбокс при включенні активує налагодження і профілювання в білді. Використовуйте цей параметр, коли хочете протестувати свій додаток.
6) **Autoconnect Profiler** - цей чекбокс можна включити тільки при включеному Development Build. Коли цей параметр активований профайлер Unity (про нього ще буде стаття, якщо коротко то потрібен для перегляду і налагодження роботи програми) буде автоматично підключатися до білду і профілювати його.
7) **Deep Profiling** - цей чекбокс також можна включити тільки при активному Development Build. Він автоматично підключає глибоке профілювання в профайлері. Ця функція дозволяє більш глибоко досліджувати ваш додаток, але при цьому робота всіх скриптів і всього додатка буде уповільнена.

8) **Script Debugging** - цей чекбокс також можна включити тільки при активному Development Build. Це дозволяє вам налагоджувати свій код. Ця функція не доступна на WebGL.
9) **Scripts Only Build**-запустіть збірку білда щоб перевірити чи змінилися скрипти. Для цього проект вже повинен бути зібраний хоча б один раз.
10) **Compression Method** – вище я вже писав про це-дозволяє вибрати метод компресії (стиснення) ігрових ресурсів.

Важливо звести Розмір файлу білда до мінімуму, особливо для мобільних пристроїв, так як зазвичай магазини додатків для мобільних накладають обмеження на розмір білда. Першим кроком у зменшенні розміру є визначення того, які ресурси вносять в нього найбільший внесок, оскільки саме ці ресурси є найбільш вірогідними кандидатами на оптимізацію. Ця інформація доступна в Лозі редактора відразу після виконання збірки білда.

Журнал редактора показаний надає зведення ассетів за категоріями, а також показує яке процентне співвідношення по пам'яті займає той чи інший тип. Після чого він починає в порядку убування перераховувати від найбільшого до найменшого ресурси в грі. Зазвичай текстури / звуки / анімації важать багато в той час як скрипти/рівні/шейдера мало. Також не забувайте, що всі ресурси гри, що зберігаються в папці Resources будуть додані в підсумковий білд.

Журнал редактора допоможе визначити які саме ресурси необхідно оптимізувати або видалити, але перед початком роботи слід пам'ятати про наступне:

Під час складання білда Unity кодує імпортовані ресурси (наприклад, текстури) в свої внутрішні формати, тому для роботи потрібно вибирати формат файлу, з яким зручно працювати(наприклад png).

Unity видаляє більшість невикористовуваних ресурсів під час складання. Єдині ресурси, які Unity видалити не може це скрипти і файли, що зберігаються в папці Resources (тому як Unity не знає які файли з цієї папки будуть використані під час роботи гри, а які ні). Маючи це на увазі ми завжди повинні тримати ресурси в папці Resources мають відношення до самої гри-інші сміливо можна видаляти. Також можна подумати про заміну ассетів (вони ж ресурси) з папки Resources на AssetBundles – це означає що такі асети будуть завантажуватися динамічно, тим самим зменшуючи розмір білда.

Пропозиції щодо зменшення розміру білда.

Зазвичай текстури займають більшу частину розміру. Першим рішенням зменшує їх є компресія (вона налаштовується при імпорті текстур в Інспекторі. Якщо це не зменшує розмір файлу достатньо, спробувати зменшити фізичний розмір текстури (в пікселях). Щоб зробити це без змін вихідного файлу, потрібно вибрати текстуру у вікні проекту і у вікні Інспектора змініть параметр Max Size. Щоб побачити, як це виглядає в грі, потрібно збільшити масштаб ігрового об'єкта, який використовує цю текстуру, а потім відрегулювати Max Size, поки він не почне виглядати у вікні сцени гірше. Зміна параметра Max Size ніяк не впливає на ассет текстури, тільки на його дозвіл в грі.

За замовчуванням Unity стискає всі текстури при імпорті. Для прискорення робочого процесу в редакторі потрібно перейти по шляху Edit \ Preferences і у вкладці General прибрати галочку напроти пункту Compress Assets on Import. Всі текстури стискаються при складанні білда незалежно від цього параметра.

Також можна стискати сітки і кліпи анімації при імпорті (знову ж таки для зменшення ваги підсумкового білда). Щоб включити компресію сітки (Mesh compression),потрібно вибрати у вікні проекту необхідну сітку, і у вікні Інспектора встановити напроти пункту Mesh compression один з трьох параметрів: Low, Medium або High. Потрібно пам'ятати, що стиснення сіток і

анімацій вносить в них неточність, тому з цими параметрами потрібно поекспериментувати щоб підібрати потрібні для себе Значення.

Також потрібно звернути увагу що стиснення сітки зменшує займану їй пам'ять тільки на жорсткому диску, в оперативній пам'яті вона має все той же початковий вага. Також потрібно знати, що скорочення ключових кадрів в анімації зменшує вагу файлу як на жорсткому диску, так і в оперативній пам'яті.

І останнє що хотілося б сказати - це збірки. Net API використовувані для розробки в Unity. Їх існує всього дві-4.x і. Net Standard 2.0, остання має урізаний функціонал, але при цьому і важить менше, що в підсумку допоможе зменшити розмір білда.

При створенні білда іноді потрібно якимось чином змінити хід збірки. Наприклад зробити так, щоб Unity зібрав не тільки білд на ПК, а також додав установник до нього. Щоб зробити це потрібно скористатися скриптом редактора, використовуючи BuildPipeline.BuildPlayer, він потрібен для запуску збірки, а потім запустити код постобробки який потрібен.

ВИСНОВОК

В ході виконання роботи було розглянуті та проаналізовані сучасні технології створення комп’ютерних ігор, проведено аналіз тематичної літератури, вибір програмних засобів для реалізації проекту.

Цей проект буде розвиватися, доки є бажання, та я сподіваюсь, що колись він буде запущений до експлуатації, та реальні гравці зможуть відчути усю атмосферу створеного світу та просто гарно провести час за цією грою разом з іншими гравцями.

При розробці відео гри було використано такі засоби як, мова програмування C# для створення внутрішніх механізмів гри, було ророблено багато складних алоритмів для досягнення реалістичної фізики поводження персонажа та оточуючих його об’єктів. Платформа для реалізації потенціалу гри був використаний ігровий движок Unity, який допоміг мені зв’язати між собою алгоритми та візуальну частину гри. Для створення візуальної частини було викоистано не лише можливості ігрового рушія Unity, а ще такі професійні рішення як програма для моделювання Blender і Zbrush за допомогою них я досягнув естетичного візуального ряду.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Joseph Hocking — Unity in Action. Multiplatform game development in C# with Unity 5, 2015

2. Alan Thorn — Mastering Unity Scripting, 2015

3. Alan Thorn — Unity Animation Essentials, 2015

4. Alan Thorn — How to Cheat in Unity 5: Tips and Tricks for Game Development, 2015

5. Chris Dickinson — Unity 5 Game Optimization, 2015

6. Kenny Lammers — Unity Shaders and Effects Cookbook, 2013