

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської роботи
на ступінь вищої освіти бакалавр

на тему: **«РОЗРОБКА ГРИ У ЖАНРІ ACTION-RPG “BRAVE ARCHER”
НА ПЛАТФОРМУ ANDROID З ВИКОРИСТАННЯМ ДВИГУНА
UNITY»**

Виконав: студент 4 курсу, групи ПД-41
спеціальності
121 Інженерія програмного забезпечення
(шифр і назва спеціальності/спеціалізації)

_____ Ковальов Д.Д.
(прізвище та ініціали)

Керівник _____ Дібрівний О.А.
(прізвище та ініціали)

Рецензент

_____ (прізвище та ініціали)

Київ –2022

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення
Ступінь вищої освіти - «Бакалавр»
Спеціальність підготовки – 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри
Інженерії програмного
забезпечення
Негоденко О.В.
“ _____ ” _____ 2022 року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТА

КОВАЛЬОВУ ДЕНИСУ ДМИТРОВИЧУ

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка гри у жанрі Action-RPG “Brave Archer” на платформу Android з використанням двигуна Unity»

Керівник роботи: Дібрівний О.А., доктор філософії

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом вищого навчального закладу від «18» лютого 2022 року №22.

2. Строк подання студентом роботи 03.06.2022

3. Вхідні дані до роботи

Алгоритм дії ігрових застосунків, Розробка мобільної гри для платформи Android. Тестування розробленого програмного забезпечення. Науково-технічна література, пов'язана з розробкою програмного забезпечення для платформи Android

4. Зміст розрахунково-пояснювальної записки(перелік питань, які потрібно розробити).

4.1 Аналіз предметної області.

4.2 Вимоги та оцінка якості систем.

4.3 Вибір засобів реалізації.

4.4 Реалізація та тестування програмного забезпечення.

5. Дата видачі завдання 11.04.2022

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	11.04-15.04	Виконано
2	Аналіз існуючих ігор на ринку	16.04-18.04	Виконано
3	Дослідження засобів реалізації програмного забезпечення	19.04-20.04	Виконано
4	Підготовка об'єкту проектування	21.04-24.04	Виконано
5	Розробка функціоналу програмного забезпечення	25.04-14.05	Виконано
6	Розробка презентації	15.05-30.05	Виконано
7	Попередній захист роботи	31.05	Виконано
8	Здача роботи	03.06	

Студент _____
(підпис)

(прізвище та ініціали)

Керівник роботи _____
(підпис)

(прізвище та ініціали)

РЕФЕРАТ

Текстова частина магістерської роботи 60 с., 31 рис., 11 джерел.

UNITY, VISUAL STUDIO, C#, ACTION-RPG, ANDROID, ГРА,
МОБІЛЬНІ ІГРИ, ВІДЕОІГРИ

Об'єкт дослідження – ігровий процес гри «Brave Archer» на платформу Android з використанням двигуна Unity.

Предмет дослідження – жанр Action-RPG в іграх

Мета роботи – вдосконалення ігрового процесу гри в жанрі Action-RPG для платформи Android.

Методи дослідження – методи розпізнавання та групування інформації, проведення аналогії, аналіз отриманих результатів

Для досягнення поставленої мети в даній роботі:

1. Створено функціонал для реалізації гри «Brave Archer» у жанрі Action-RPG на платформу Android з використанням двигуна Unity.
2. Виявлення того, що використання ігрового двигуна Unity та середовища розробки програмного забезпечення Visual Studio – це правильне рішення.
3. Надано докази того, що ігровий двигун Unity надає змогу зручно та ефективно використовувати його при розробці проєктів, порівняно з іншими ігровими двигунами.
4. На основі виконаного аналізу розроблено гру «Brave Archer» у жанрі Action-RPG на платформу Android з використанням ігрового двигуна Unity.

Під час виконання роботи був виконаний аналіз існуючих ігор та середовищ розробки програмного забезпечення. В результаті аналізу було визначено переваги та недоліки застосунків.

Розроблено функціонал програмного забезпечення та загальну інформацію для користувачів.

Галузь використання – будь-яка людина, яка забажає пограти в гру «Brave Archer».

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	8
ВСТУП	9
1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ.....	11
1.1 Поняття про ігри	11
1.1.1 Комп'ютерна гра	11
1.1.2 Історія розвитку ігор.....	12
1.1.3 Розробка ігор.....	13
1.2 Жанр Action-RPG	14
1.3 Ігри на мобільні платформи	17
2 ВИБІР ІНСТРУМЕНТІВ РОЗРОБКИ.....	19
2.1 Огляд ігрових двигунів.....	19
2.1.1 Unity.....	23
2.1.2 Інструментарій Unity	25
2.2 Огляд середовищ розробки	30
2.2.1 Visual Studio	34
2.3 Мова програмування C#	36
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	38
3.1 Основний функціонал.....	38
3.2 Меню гри.....	44
3.3 Тестування програмного забезпечення.....	48
ВИСНОВКИ	50
ПЕРЕЛІК ПОСИЛАНЬ	52
Додаток А.....	54

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

Engine – основне ядро гри, на основі якого будуються складові гри та спрощують роботу для розробника.

Asset – частина ігрового контенту, яка оброблюється грою для відображення її на сцені.

Player – основний персонаж гри яким керує гравець.

GUI (Graphical User Interface) – інтерфейс за допомогою якого користувач має можливість взаємодіяти із програмою через візуальні елементи управління.

IDE (Integrated Drive Electronics) – інтегрована середовище розробки, яка використовується програмістами для розробки програмного забезпечення.

Скрипт – це послідовність дій, описаних за допомогою мови програмування (JavaScript, PHP, Perl, Python та ін.) для автоматичного виконання певних завдань.

ВСТУП

Обґрунтування вибору теми та її актуальність: створити комп'ютерну гру не так просто, як здається. Щоб створити дійсно цікаву та захоплюючу гру потрібно доволі багато людей, які будуть працювати над сценарієм, механіками, зовнішнім виглядом та ще багато чим. Кожен працівник несе свій вклад в розробку для того, щоб зацікавити гравця.

Нажаль деякі студії розробляють ігри лише для того, щоб заробити якомога більше грошей, не задумуючись над тим, чи буде цікаво грати в їх гру.

Створення дійсно цікавої та розвиваючої гри наразі є актуальною темою.

Ступінь вивчення проблеми: на розробці ігор дійсно можна заробити великі гроші, але щоб це сталося, треба залучити велику аудиторію гравців. Для того, щоб у гру грало багато людей – їх треба зацікавити. Є доволі багато способів це зробити, але не всі розробники їх використовують. Існують студії, які роблять багато схожих ігор із нудним сценарієм та великою кількістю реклами.

Нажаль ще однією з проблем ігор є те, що в більшості випадків діти доволі багато проводять за ними часу, таким чином забуваючи про навчання та інші важливі речі.

Об'єктом дослідження ігровий процес гри «Brave Archer» на платформу Android з використанням двигуна Unity.

Предметом роботи є жанр Action-RPG в іграх.

Метою роботи є вдосконалення ігрового процесу гри в жанрі Action-RPG для платформи Android.

Завданням роботи є розробка гри «Brave Archer» з використанням ігрового двигуна Unity на платформу Android.

Методика дослідження: методи розпізнавання та групування інформації, проведення аналогії, аналіз отриманих результатів.

Практична значущість результатів: створена гра залучить гравців будь-якого віку та буде їм цікава. Гра допоможе розвивати рефлексорні та стратегічні навички.

1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття про ігри

1.1.1 Комп'ютерна гра

Комп'ютерна гра – це програма для створення ігрового процесу, під час якого, гравець взаємодіє з ігровим інтерфейсом за допомогою пристроїв, таких як клавіатура, мишка або геймпад.

Ігри бувають різні і кожен зможе знайти для себе улюблену. Доволі багато людей використовують ігри, щоб відволіктись від якихось справ або просто добре провести час. Хтось грає для того, щоб кинути собі виклик та побачити, чи зможе він пройти декілька складних рівнів. Інша ж частина гравців полюбляє ігри з великим та цікавим сюжетом. Перелічувати можна ще багато чого, але можна виділити основні відмінності, це жанр та платформа.

Жанри відеоігор:

- Екшн;
- Стратегія;
- Симулятор;
- Пригоди;
- Головоломки;
- Рольові;

Ігрові платформи:

- Персональні комп'ютери
- Домашні консолі
- Мобільні платформи
- Веб-платформи
- Віртуальна/доповнена реальність

1.1.2 Історія розвитку ігор

Хоча сьогодні відеоігри можна знайти в будинках по всьому світу, вони насправді зародилися в дослідницьких лабораторіях вчених. У 1952 році британський професор А. С. Дуглас створив ОХО, також відому як хрестиків-нулики, в рамках своєї докторської дисертації в Кембриджському університеті. А у 1958 році Вільям Хігінботам створив «Теніс для двох» на великому аналоговому комп'ютері та підключив екран осцилографа для щорічного дня відвідувачів у Брукхейвенській національній лабораторії в Аптоні, штат Нью-Йорк.

Це були лише перші кроки в світі ігор. Ігри почали еволюціонувати. Стала покращуватися графіка, функціонал та почали з'являтися різноманітні жанри. Але щоб гра змогла увімкнутися треба було використовувати комп'ютери потужніші за своїх попередників. Тому зі стрибком ігор, почали еволюціонувати і пристрої, на яких вони запускалися. Так почали з'являтися ігрові автомати та домашні консолі. У 1995 році Sega випустила в Північній Америці свою систему Saturn, першу 32-розрядну консоль, яка запускала ігри на компакт-дисках, а не на картриджах.

У 2000-ні роки відеоігри ставали все реалістичнішими в плані графіки, розвинулися он-лайн ігри та мережеві режими, що дозволили багатьом гравцям взаємодіяти в реальному часі. Наприкінці десятиліття відбулася інтеграція з ігровими сервісами та соціальними мережами. Ігри на фізичних носіях стали відходити у минуле, поступаючись цифровій дистрибуції. Це обумовило поширення інді-ігор, ігор від незалежних розробників чи навіть окремих людей. Гравці стали активними учасниками створення й модифікації відеоігор шляхом моддингу, конструювання інтерактивних творів у спеціальних редакторах. Типовим стало випускати завантажувані доповнення до ігор, у тому числі платні, впроваджувати мікротранзакції, покликані за реальні гроші полегшувати гру.

1.1.3 Розробка ігор

Так, ігри зараз дивують кожного хто хоч раз побачить їх на екрані. Але чи все так просто, як може здатися на перший погляд? За вашими емоціями, здивуваннями та відчуттями, які ви отримали в процесі гри, стоїть велика праця багатьох людей. Щоб створити дійсно якісну гру знадобиться велика група людей. І кожен буде важливий, бо від нього залежить, як саме буде виглядати головний герой, який сюжет буде в цієї гри або чи зможе гравець літати на літаках. Деякі дуже великі компанії починають створювати свої студії, щоб одночасно розроблялося декілька ігор. Наприклад в компанії Electronic Arts в 2021 році працювало більш ніж 11 тисяч людей.

Все починається з ідеї. Спочатку геймдизайнер повинен придумати концепцію гри, її основні механіки та риси, усе базове, що стосується гри. Потім програміст власне пише всі необхідні механіки та функціонал, створює файлову архітектуру. Художник малює концептуальні рисунки, щоб дизайнеру рівнів було на що спиратися. Також в процес створення гри підключені звукорежисери, сценаристи, тестери та багато інших людей. Необхідні ролі для створення гри:

- Геймдизайнер;
- Сценарист;
- Програміст;
- Художник;
- Дизайнер рівнів;
- Звукорежисер;
- Тестер;
- Піар-менеджер;
- Актор озвучування;

Розробка гри може зайняти доволі багато часу. Наприклад гра «Відьмак 3» розроблялась 3,5 роки. Гра наповнена великою кількістю контенту, цікавих

історій та різноманітних дій. Також гра мала дуже добру рекламу. В сукупності усіх цих частин гра виявилася неймовірною, через що загальні продажі склали більш ніж 30 мільйонів, а гра отримала величезну кількість нагород.

1.2 Жанр Action-RPG

В іграх такого жанру необхідно використовувати рефлексивність та швидкість реакції для подолання ігрових обставин. Це один із базових жанрів і водночас найпоширеніший. Як правило екшн-ігри пов'язані із агресивними діями щодо противників або оточення. Персонаж гравця повинен битися, стріляти, переслідувати ціль чи самому уникати переслідування.

Цей жанр поділяється на велику кількість піджанрів, серед яких основними є:

- Файтнігни;
- Шутери;
- Платформери;
- Лабіринти;

Від звичайної рольової гри рольовий екшн відрізняється насамперед тим, що швидкість дій гравця має таке ж важливе значення, як і характеристики персонажа, у той час як у RPG з покроковим керуванням персонажем такий аспект відсутній. В Action/RPG бої йдуть у реальному часі, і ваш успіх залежить не тільки від характеристик героя, а й від того, наскільки швидко ви натиснете кнопку. Цей жанр добре підходить для тих гравців, хто хоче швидкої динаміки, розвиток головного персонажа та відчуття суперництва. Причини його популярності очевидні: ігри Action RPG поєднують у собі елементи жанрів екшн та RPG, що дозволяє розробникам комбінувати різні механіки та дозволяє гравцям насолоджуватися динамічним геймплеєм, не вникаючи у складності розвитку персонажа, властиві CRPG. У той час як класичні рольові ігри вимагають ретельного вибору функцій та навичок, покладаючись на

прихильність системи, що відповідає за кидок віртуальних кісток, екшн-рольові ігри дають геймерам можливість зробити свій внесок у бойовий процес за рахунок своєї швидкості рефлексів, зменшуючи жорстку залежність від гри для досягнення мети. накачування.

Наприклад, *Beyond Oasis* зазвичай відносять до action/RPG, хоча гра виглядає як аркадна *Streets of Rage* або *Final Fight*, просто трохи інша проекція. Тим часом, *Diablo*, незважаючи на систему зброї, частіше називають *hack'n'slash* або *rogue* (аркадна гра з елементами RPG, слабким сюжетом і рівнями, що випадково генеруються), а не action/RPG. Крім того, ще одна різниця між екшн/рольовими іграми полягає в тому, що в таких іграх завжди є певна кількість головоломок, хоч і не така, як в іграх типу квестів. Вони допомагають применшити бойовики, спокійно шукаючи вирішення простих, але не завжди очевидних проблем. Ось чому ентузіасти ПК часто називають екшн/рольові ігри «пригодами», хоча класичні пригоди все ж таки трохи відрізняються.

Серія ігор *Dark Souls* підкорила доволі багато гравців. Ворожий світ наповнений великою кількістю монстрів, дружелюбних NPC зовсім мало та немає можливості зробити збереження у будь-якій точці карти. Виглядає це все не дуже привітливо, але це і є фішка цієї гри. Гравцям кидають виклик, чи зможуть вони пройти через всі ці перешкоди, перемогти усіх босів та дійти до кінця. Звісно ж у грі присутній великий вибір кастомізації екіпірування, навичок та зброї, що також зацікавлює гравців. Весь ігровий світ виглядає фантастичним та похмурим, великі квіткові поля, зруйновані замки та темні печери. Усі ці локації комплексно надають грі своєї атмосфери та повноцінного вигляду.



Рисунок 1.1 – Гра «Dark Souls 3»

Cyberpunk 2077 молода гра в жанрі Action/RPG. Вона має свою базу прихильників та критиків. Так, гра має свої мінуси у вигляді помилок чи критичному завершенні процесу. Але складно сперечатися з тим, що гра дуже цікава та неймовірно гарна. Цю гру розробляли приблизно 8 років і за коментарями багатьох гравців, вона коштувала такого тривалого очікування. В грі Cyberpunk 2077 досить багато цікавих персонажів, безліч неймовірних завдань, варіативність проходження гри та велика гілка прокачування здібностей головного героя. Великий відкритий світ надає змогу відволіктись від головної гілки завдань та виконати пару побічних завдань або просто покататися та відкрити для себе нові цікаві локації.

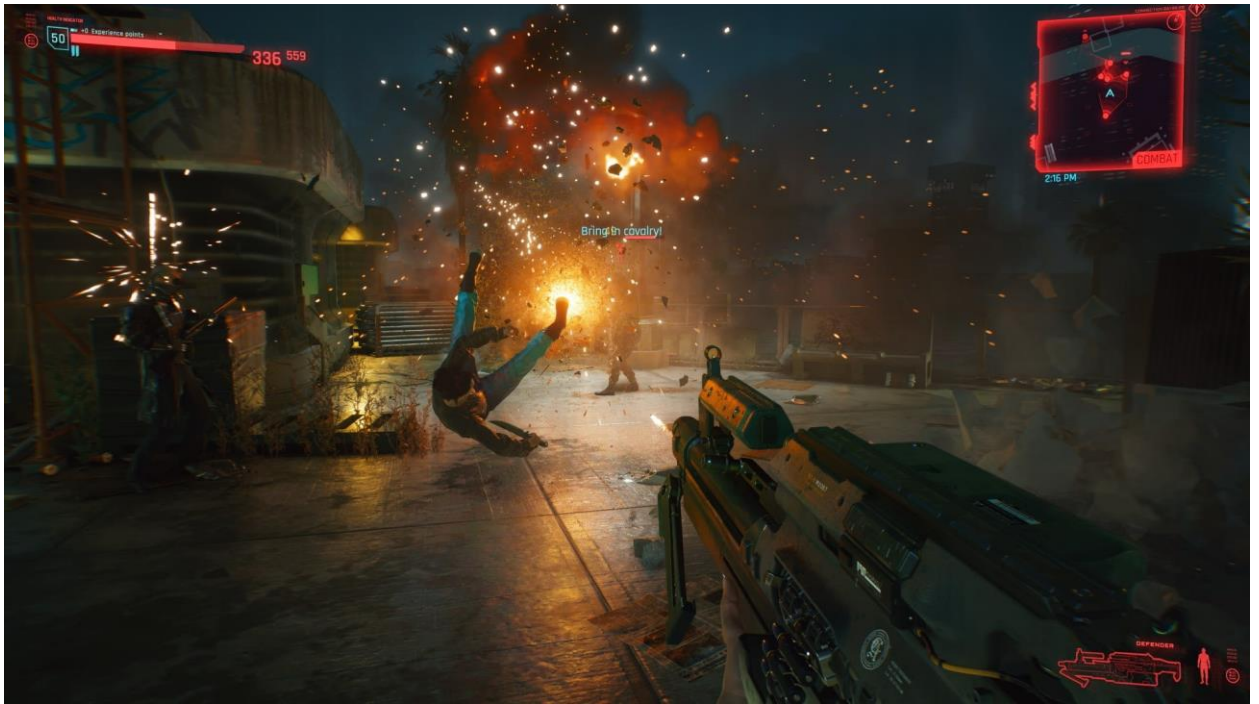


Рисунок 1.2 – Гра «Cyberpunk 2077»

1.3 Ігри на мобільні платформи

Мобільні ігри – це відеоігри, які розроблені саме на мобільні пристрої. Це може бути, як телефон, так і планшет або портативна ігрова консоль. Пристрій повинен мати доступ до мережі, але не завжди.

Гарним прикладом найвідомішої мобільної гри є тетріс, який був зроблений у 1994 році. Також доволі відомою грою є змійка, яка з'явилася у 1997 році. Гра настільки сподобалась користувачам, що була встановлена на пристрої Nokia понад 350 мільйонів разів. Версій цієї гри дуже багато, з різноманітними варіантами графічного оформлення та навіть з додаванням додаткових функцій, наприклад для двох гравців.

Сьогодні мобільні ігри доволі популярні, бо в них можна зіграти будь-де, для цього потрібен лише телефон. Ігри в такому варіанті гарний варіант у тривалу подорож або ж у перерві між справами.

Мобільні ігри зазвичай завантажуються з магазину додатків як і з порталів мобільного оператора, але в деяких випадках також попередньо

завантажується в портативні пристрої OEM або оператором мобільного зв'язку при придбанні через інфрачервоне з'єднання, Bluetooth, або картки пам'яті, або збоку завантажених на трубку за допомогою кабелю.

Завантажувальні мобільні ігри були вперше комерціалізовані в Японії близько запуску NTT DoCoMo І-режим платформи в 1999 році, а на початку 2000-х були доступні на різних платформах по всій Азії, Європі, Північній Америці і, зрештою, на більшості територій, де сучасні мережі перевізників та мобільні телефони були доступні до середини 2000-х. Однак мобільні ігри розповсюджують мобільні оператори та сторонні портали (канали, спочатку розроблені для монетизації завантажувальних мелодій дзвінка, шпалер та інших дрібних фрагментів вмісту з використанням преміум-SMS або прямих платежів оператора як платіжного механізму), залишалися маргінальною формою розповсюдження ігор до Apple Store, який був запусканий в 2008 році. Як перший ринок мобільного контенту, яким безпосередньо керує компанія мобільний тримач платформи, App Store змінив поведінку споживачів та швидко розширив ринок мобільних ігор, оскільки майже кожен власник смартфона почав завантажувати мобільні програми.

З кожним роком фани чекають продовження або виходу нової гри, їх випускають так багато, що можна зробити тільки один висновок: на мобільні ігри є попит.

Найбільшим ринковим гравцем є компанії Playtika, яка сьогодні посідає 16-те місце у світі з річним оборотом \$2,37 млрд. Компанія примітна тим, що за останні 10 років її було продано двічі. Вперше ще на етапі стартапу за \$160 млн, а вдруге вже у 2016 році – китайській компанії за \$4,4 млрд. Незважаючи на те, що Playtika була заснована двома ізраїльськими бізнесменами, спочатку вона піднімалася силами українських розробників. Тобто нинішня корпорація багато в чому завдячує українським програмістам – і справа не в грошах.

Сьогодні є доволі багато різних мобільних платформ, але найпопулярнішими вважаються наступні платформи для мобільних пристроїв:

- Android;

- Windows Phone;
- iOS;
- Symbian;

2 ВИБІР ІНСТРУМЕНТІВ РОЗРОБКИ

2.1 Огляд ігрових двигунів

Ігровий двигун (game engine) – це її основне ядро, базове програмне забезпечення, на основі якого будуються всі інші складові гри. Програмний код, який може використовуватися для створення варіацій гри або навіть зовсім нового ігрового світу.

Вперше визначення з'явилося в середині 90-х років, коли почали з'являтися ігри, схожі на головний шутер того часу – Doom. У той же час у вільному доступі почали з'являтися ігрові двигуни, на основі яких і сторонні розробники, і звичайні користувачі могли пробувати писати власні ігри.

З тих пір ігрові двигуни ставали все більш складними технічно, довгими і насиченими за своїм програмним кодом. Але при цьому, як і на початку свого існування, вони містять у собі жорстко фіксовані дані:

- Ігрову логіку;
- Фізику об'єктів;
- Правила відтворення об'єктів;
- Геймплей в цілому;

Сьогодні існує доволі багато різних ігрових двигунів. В кожного з них є свої переваги та недоліки. Якісь компанії використовують найпопулярніші двигуни та ті, які підходять їм по функціоналу та можливостям. Але є й такі компанії, які розробляють власний ігровий двигун і роблять ігри завдяки ньому. Найбільш використовуваними є такі ігрові двигуни:

- Unity;

- Unreal Engine 4;
- Cry Engine;
- RPG Maker;
- Frostbite;

Unity поширений ігровий двигун, який використовують багато компаній. Є одним із лідерів серед ігрових двигунів. Зручний для багатьох інтерфейс одразу дає розуміння, що та де знаходиться, які функції має та як саме їх можна використати. Саме за це Unity люблять ті розробники, які тільки починають свій шлях в сфері розробки ігор.

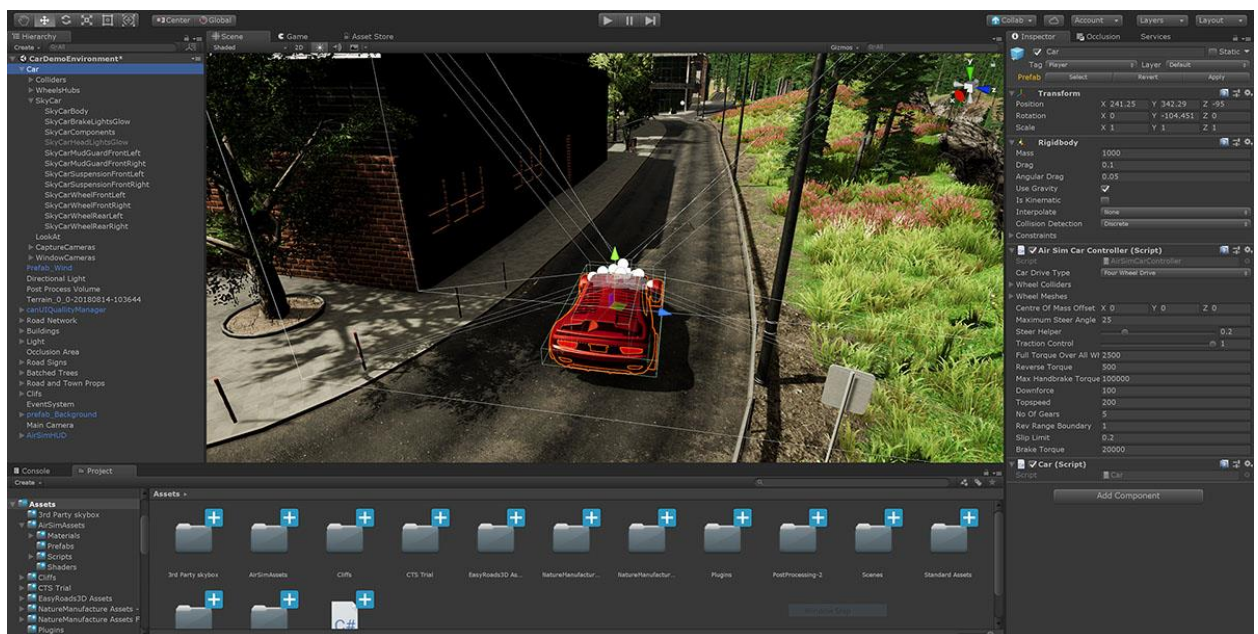


Рисунок 2.1 – Ігровий двигун Unity

Unreal Engine один із найпопулярніших, відомих та зручних ігрових двигунів. Він має дуже великий функціонал та список різноманітних інструментів. Має змогу підключити сторонні плагіни, що тільки поширює його та покращує процес розробки. Також його широко використовують великі компанії, які готуються створити велику та привабливу гру, бо можливість Unreal Engine дійсно велика.

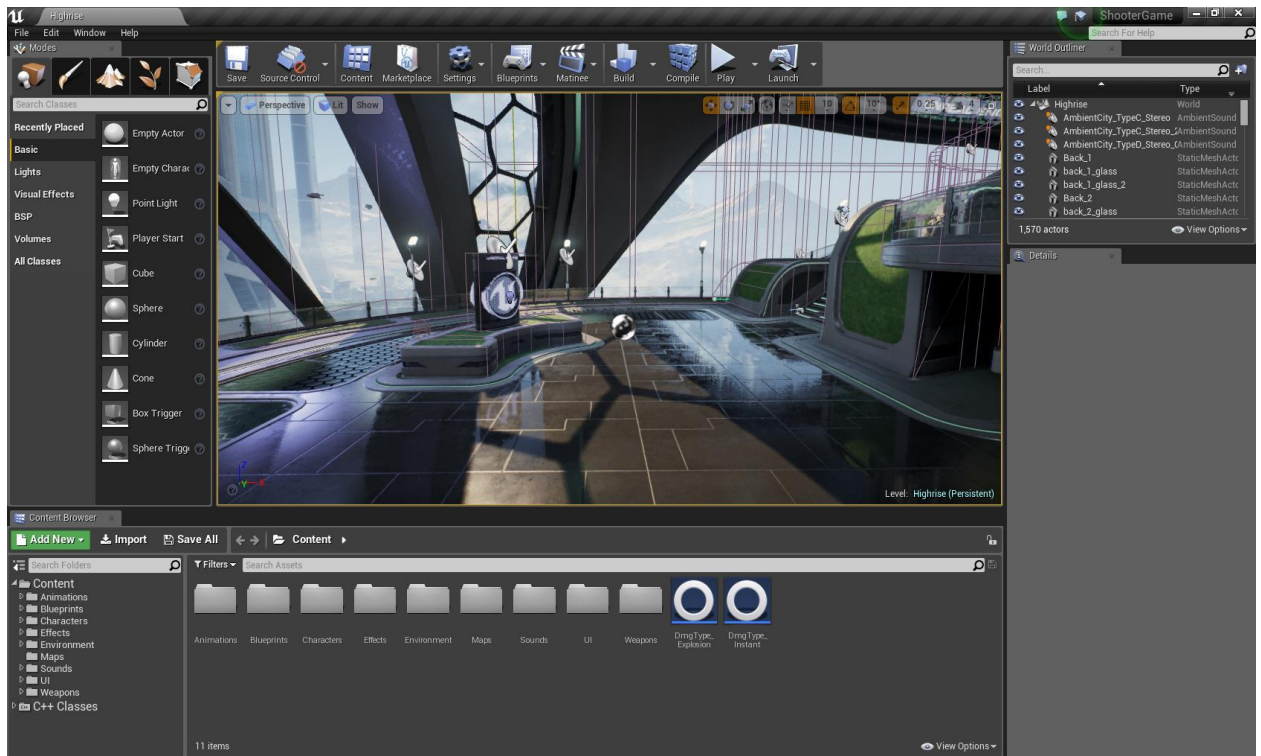


Рисунок 2.2 – Ігровий двигун Unreal Engine

CryEngine прославився тим, що за допомогою нього були створенні відомі ігри, такі як серія FarCry. Цей ігровий двигун забезпечує відображення величезних локацій та має відмінну імітацію нетвердих об'єктів. Для налаштувань деяких параметрів, CryEngine використовує штучний інтелект для покращення фізики та зовнішнього вигляду об'єктів. Відмінна платформа для освідчених розробників та початківців.

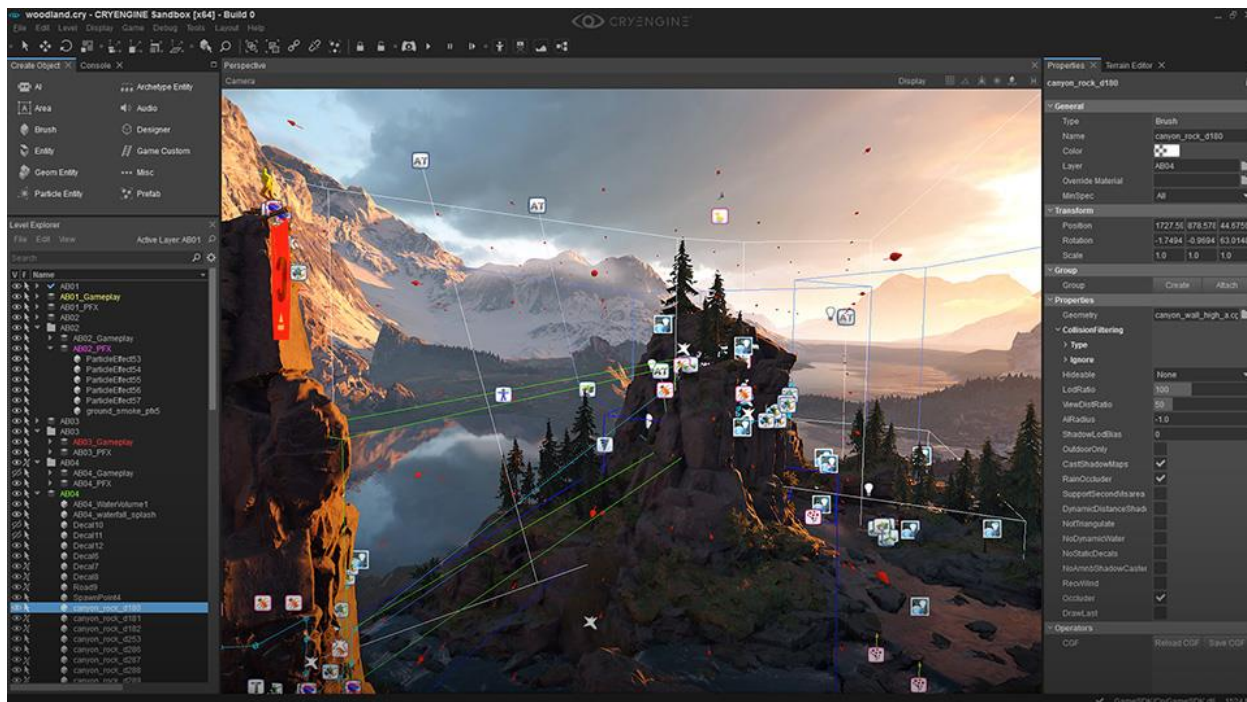


Рисунок 2.3 – Ігровий двигун CryEngine

RPG Maker добре підходить для тих, хто тільки хоче спробувати себе в розробці. Ігровий двигун допоможе створити гру від початку і до кінця, якщо користувач навіть нічого не знаю про програмування ігор. Він дозволяє налаштувати кожен аспект вашої гри за допомогою простого у користуванні інтерфейсу, що робить його ідеальним для початківців.

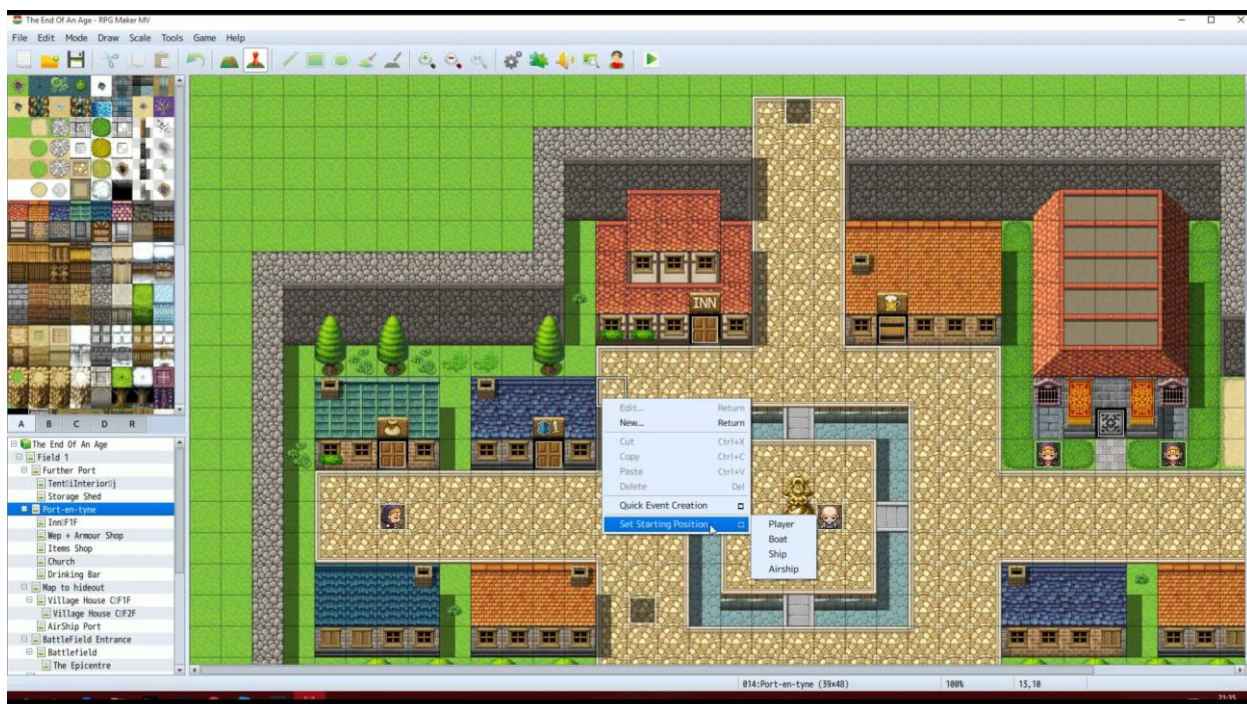


Рисунок 2.4 – Ігровий двигун RPG Maker

Frostbite це ігровий двигун про який знають всі фанати ігор від Electronic Arts. Він використовується в багатьох іграх цієї студії, включаючи Battlefield, FIFA, DragonAge. Простіше кажучи, це платформа-універсал, на основі якої написані РПГ, гонки, спортивні симулятори і т. д. Інший дивовижний момент криється в тому, що двигун має багато переваг: безліч пост-ефектів, відмінна руйнівність об'єктів, тривимірні і двомірні спецефекти, реалістичні текстури і навіть окремий ігровий редактор для роботи з шейдерами і дрібними деталями.

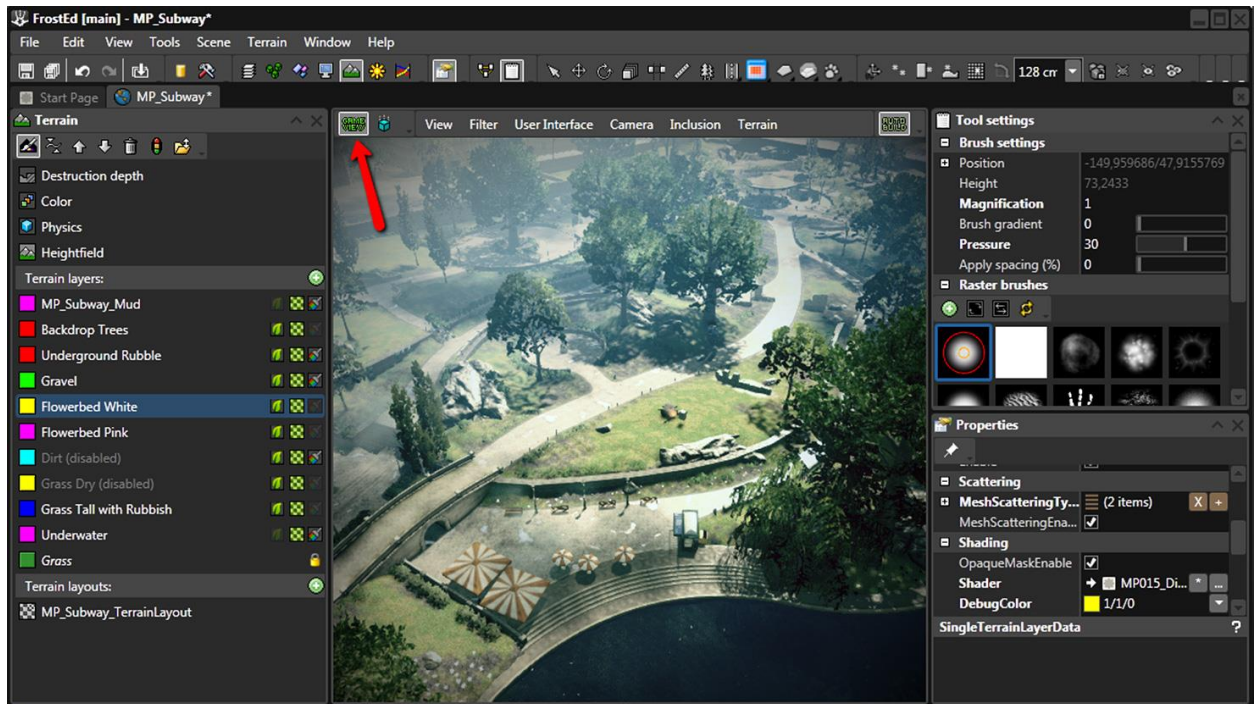


Рисунок 2.5 – Ігровий двигун Frostbite

2.1.1 Unity

Unity – це ігровий двигун, який надає змогу розробляти ігри на велику кількість різних платформ. Цей ігровий двигун був розроблений у 2005 році американською компанією Unity Technologies. Головними перевагами цієї програми є те, що вона має візуальне середовище розробки, модульну систему компонентів та міжплатформену систему підтримки. На Unity написані тисячі ігор та додатків, які доступні на багатьох платформах та охоплюють велику кількість жанрів. Цей ігровий двигун досить функціональний та зручний, що

ним користуються не тільки великі компанії розробників але й маленькі студії, початківці та навіть студенти.

Unity дозволяє розробляти ігри усіх можливих типів:

- Мобільні, комп'ютерні та для консолей;
- 2D та 3D ігри;
- Класичні, адаптовані для віртуальної та доповненої реальності;

Цей ігровий двигун має маленький поріг входу, тому новачкам, які тільки ознайомилися з цією програмою не складно почати розробляти ігри або додатки. Функціонал в Unity доволі великий, тому за допомогою нього можна створити дійсно щось цікаве та оригінальне.

- Переваги ігрового двигуна Unity:
- Тривимірне та технічне моделювання;
- Сучасна графіка;
- Готова фізика та взаємодія об'єктів;
- Скрипти на C# та JavaScript;
- Мультиплатформеність;
- Unity є безкоштовною;

У Unity скрипти можна використовувати розробки практично будь-якого елемента гри чи інтерактивного контенту з графікою реального часу. Unity підтримує скрипти на C#, створені відповідно до одного з двох основних підходів: традиційного та широко використовуваного об'єктно-орієнтованого підходу та інформаційно-орієнтованого підходу.

Одне з найсильніших місць Unity — скрипти на C#. Вам не потрібно вивчати вбудовану мову скриптування з її логікою та обмеженнями – ви просто пишете, що вам потрібно. На скрипти можна повісити будь-що: поява ворогів з-за кутка, вміст скрині зі скарбами або відповіді другорядного героя. Якщо ви вмієте програмувати, то зможете зробити таку складну і цікаву гру, як захочете самі.

Unity підтримує багато різних платформ, таких як Linux, IOS, Android, Windows тощо. в цілому майже 30 платформ. Платформи, які підтримує Unity:

- Персональні комп'ютери (Windows, Linux, MacOS);
- Мобільні платформи (Android, iOS, tvOS);
- Домашні консолі (PS4, PS5, Xbox One, Xbox Series X/S, Nintendo Switch);
- Віртуальна/доповнена реальність (Oculus Rift, Steam VR, Apple ARKit, Google ARCore, Playstation VR, Microsoft HoloLens);
- Веб-платформа WebGL;

Для мережевих і багатокористувацьких ігор реального часу Unity - найкращий варіант. Він забезпечує сервери для спільної роботи в режимі реального часу. Ефективність двигуна Unity надзвичайна. Багатопотокова обчислювальна система підходить для інтенсивних сценаріїв. Двигун Unity має вдосконалені інструменти профілювання, які визначають, чи пов'язана ваша гра з процесором, а також розповідає, як її оптимізувати з кращим досвідом. Unity має одну з найкращих систем візуалізації графіки. Це дозволяє створювати такі середовища, як світлий день, яскраве сяйво неонових вивісок, слабо освітлені опівночі та тіньові тунелі. Він створює дивовижні візуальні ефекти. Хоча Unity підтримує багатоплатформенність, але все ж намагається охопити графічний API низького рівня для кожної платформи для кращих результатів та полегшення роботи користувачів.

2.1.2 Інструментарій Unity

В ігровому двигуні Unity доволі багато функцій та інструментів. Але з усього великого списку, можна виділити основні, а саме:

- Сцена;
- Ігрове вікно;
- Проект;
- Консоль;
- Ієрархія;

- Панель інструментів;
- Оглядач;

Сцена необхідна для того, щоб розмістити на ній об'єкти. Це може бути сфера, площина, куб, елемент інтерфейсу або спрайт. Також за допомогою сцени з'являється можливість налаштувати об'єкт, наприклад додати до об'єкту скрипт або змінити його розмір та позицію на рівні.

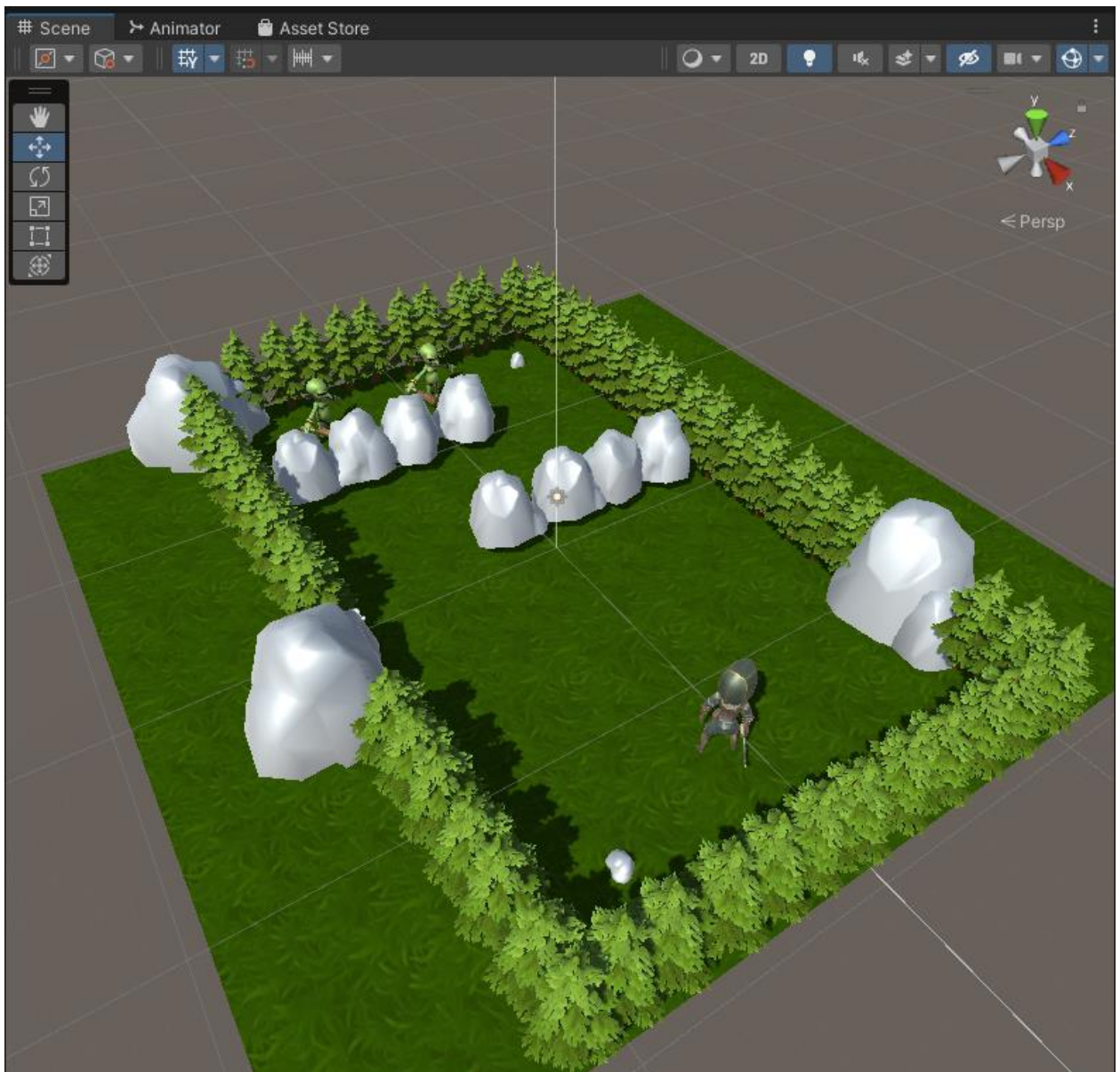


Рисунок 2.6 – Сцена

Ігрове вікно показує, як буде виглядати рівень, після того, як ви запусите гру. Почнуть працювати скрипти, які розробник встановив на об'єкти та можна буде побачити їх роботу.

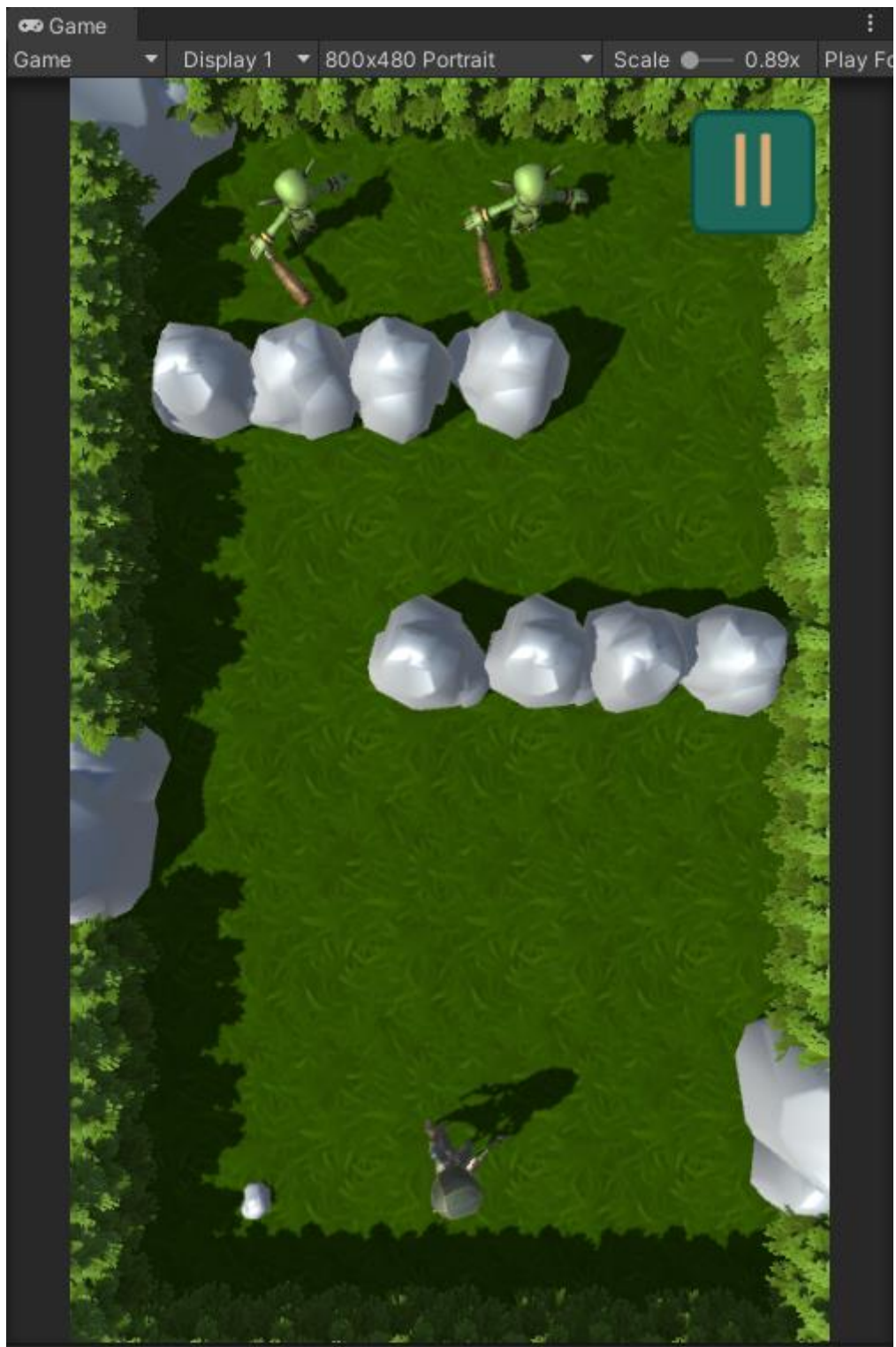


Рисунок 2.7 – Ігрове вікно

Вікно «проект» показує, усі файли які знаходяться безпосередньо в самому проекті. Для організації проекту, можна створювати папки, змінювати їм колір та швидко переміщатися між ними за допомогою бокової панелі.

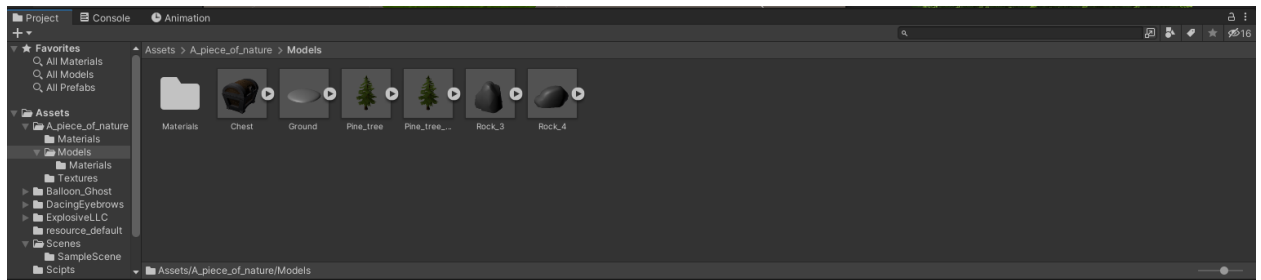


Рисунок 2.8 – Вікно «Проект»

Консоль досить корисна річ, бо вона може виводити помилки та дає змогу побачити, де саме вона знаходиться. Також консоль може повідомляти про те, що якийсь об'єкт не використовується на сцені, а просто займає пам'ять.

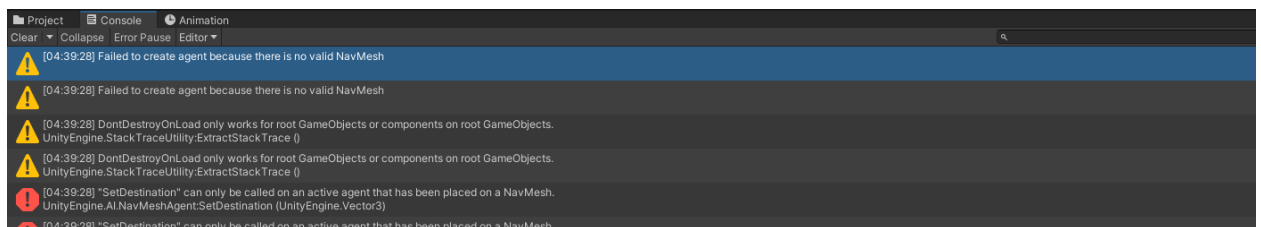


Рисунок 2.9 – Консоль

Ієрархія доволі корисна тим, що дає змогу побачити увесь список об'єктів, які знаходяться на сцені. За допомогою неї, можна вибрати певний об'єкт не шукаючи його на сцені, а зі списку, та одразу перейти до його налаштувань. Також з'являється можливість виключити або виключити відображення об'єкту на сцені.

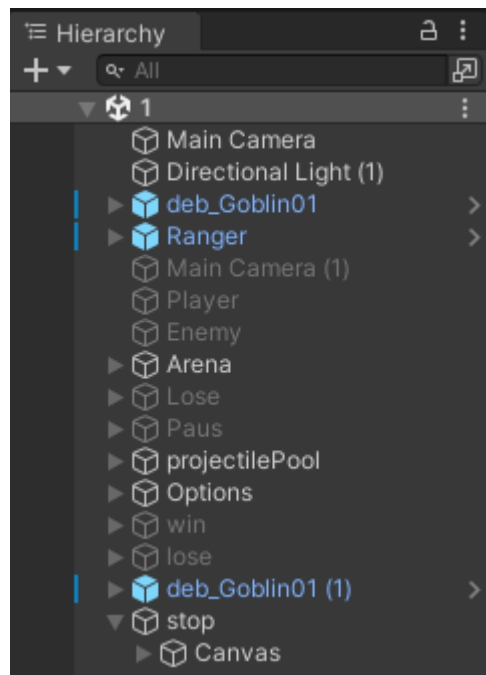


Рисунок 2.10 – Ієрархія

Панель інструментів дає швидкий доступ до налаштувань параметрів об'єкту, наприклад розмір, кут нахилу чи переміщення його на сцені.

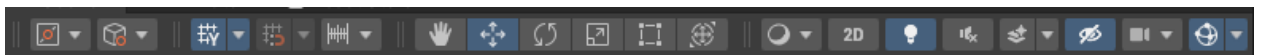


Рисунок 2.11 – Панель інструментів

Оглядач є одним із найважливіших вікон в Unity. Він дає змогу до повного налаштування та параметрів об'єкту. Якщо треба змінити розмір, додати скрипт або властивості, це можна зробити за допомогою оглядача.

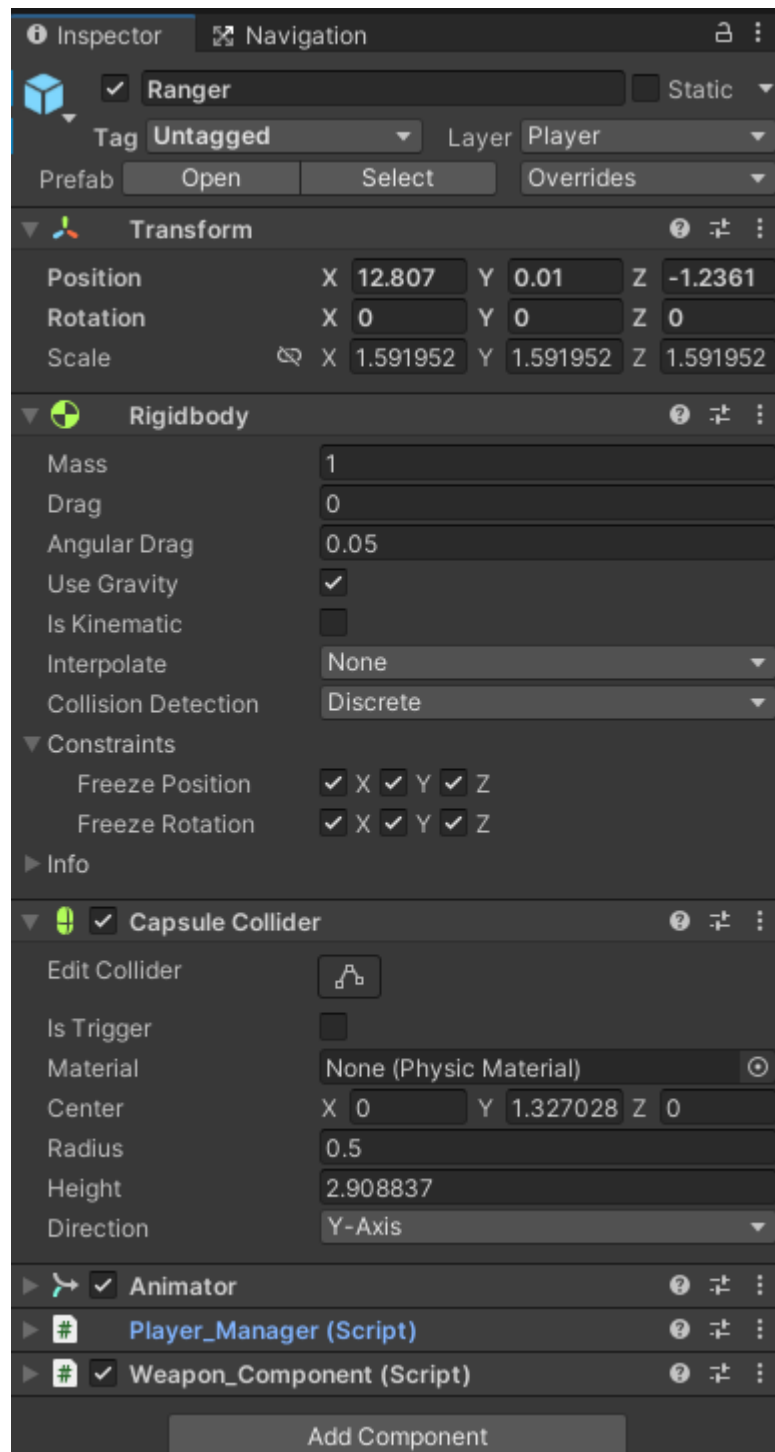


Рисунок 2.12 – Оглядач

2.2 Огляд середовищ розробки

IDE(Integrated development environment) або інтегроване середовище розробки – це програма, призначена для допомоги програмістам та розробникам у створенні програмного забезпечення. Інтегровані середовища

розробки створені для того, щоб максимізувати продуктивність програміста, надавши йому пов'язані інструменти розробки зі схожими інтерфейсами як одну програму, в якій відбуватиметься весь процес розробки й яка надає необхідні функції для модифікації. Одним із завдань IDE є зменшення часу, необхідного на конфігурацію різноманітних інструментів розробки, натомість пропонуючи той самий набір, як єдине ціле. Це може збільшити продуктивність розробника, у випадку, коли навчання тому, як працює інтегроване середовище розробки є швидшим, ніж освоєння всіх інструментів зокрема. Крім того, більша інтеграція між вбудованими інструментами потенційно може сприяти додатковому збільшенню продуктивності. Наприклад, синтаксичний аналіз коду може відбуватися безпосередньо під час його редагування, тим самим виявляючи помилки ще до трансляції коду. Більшість IDE включають:

- Редактор вихідного коду;
- Компілятор;
- Налаштовувач;
- Засоби автоматизації побудови;

Наразі існує доволі багато різних інтегрованих середовищ розробки для написання коду на мові програмування C#. Кожна з них має свої переваги та недоліки. Але серед всіх існуючих IDE можна виділити найпопулярніші, а саме:

- Visual Studio;
- Project Rider;
- Eclipse;
- Mono Developer;

Visual Studio визнана найкращою IDE для C#. Справа в тому, що обидва продукти належать корпорації Microsoft. Тому вони ідеально підходять для роботи один з одним. Середовище містить багато інструментів, які дуже добре працюють з C#. У цього середовища також є можливість зберігання даних у хмарі та підключати сторонні бібліотеки та розширення. Інтерфейс Visual

Studio доволі зрозумілий та гнучкий, що одразу дає розуміння у навігації в програмі.

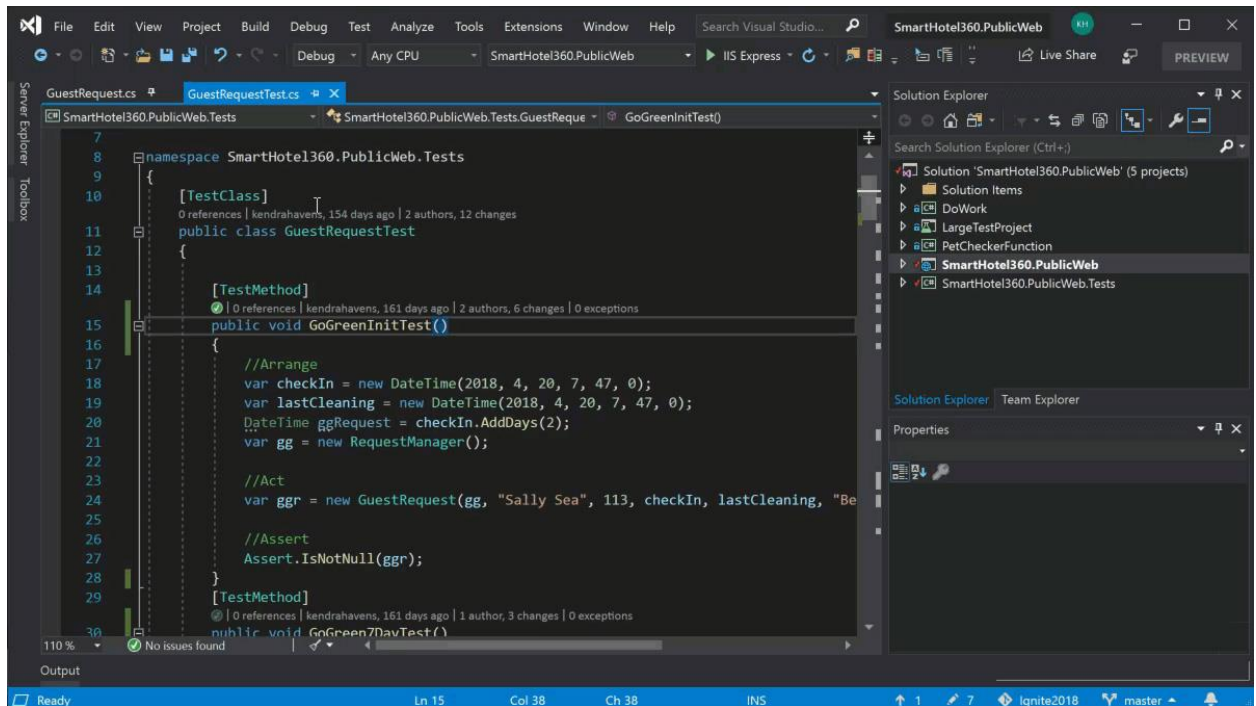


Рисунок 2.6 – Інтерфейс IDE Visual Studio

Project Rider є кросплатформовою .Net IDE. Вона підходить для використання під Windows, Linux, Mac OS X. Продукт базується на IntelliJ IDEA та ReSharper. В цій програмі чудово реалізована підтримка інтелектуальних сполучень клавіш. Також це середовище розробки має потужну підтримку навігації та рефакторингу. Project Rider має інтеграцію з Visual Studio та Unity, що надає більше можливостей при розробці ігор.

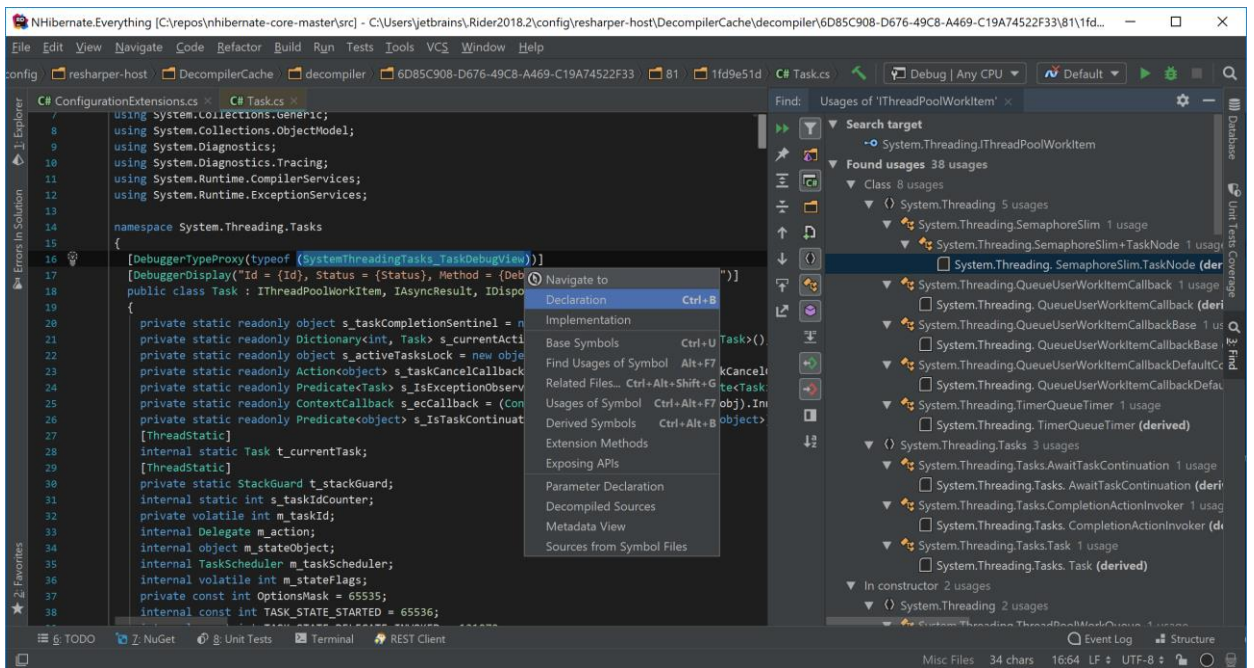


Рисунок 2.7 – Інтерфейс IDE Project Rider

Еclipse – це одне із найбільш функціональних IDE з відкритим кодом. Графічний інтерфейс доволі зрозумілий та гнучкий. Eclipse надає об’ємні можливості аналізу коду та інтеграцією з іншими програмами контролю версій. Раніше ця програма використовувалася для розробки Java, але сьогодні підтримується більш великий список мов. Це середовище розробки доступне для Windows, Linux та MacOS.

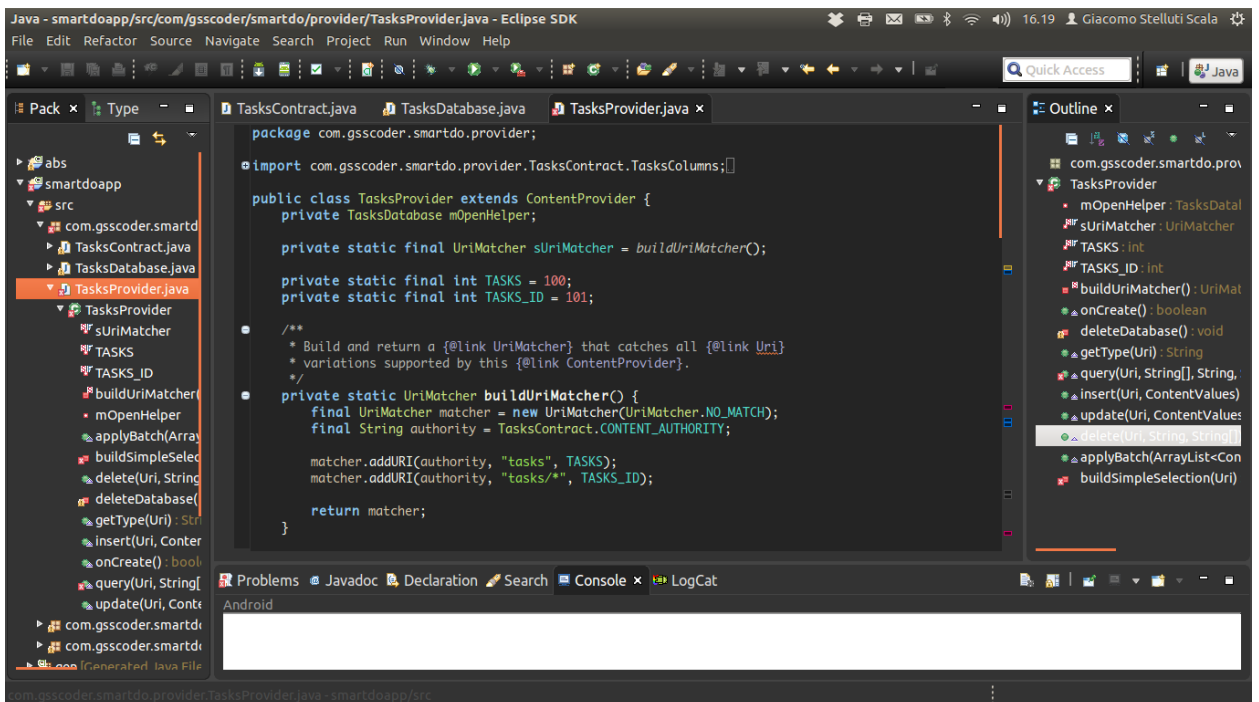


Рисунок 2.8 – Інтерфейс IDE Eclipse

MonoDeveloper добре підходить для швидкої розробки настільних та web-додатків. IDE дозволяє переносити програми .Net, написані у Visual Studio, на Linux та Mac OS, оскільки підтримує єдину базу коду платформ. Основними перевагами MonoDeveloper є мультиплатформеність, можливість налаштування продукту під кожного розробника, наявність великого списку інструментів та підтримка платформи Unity.

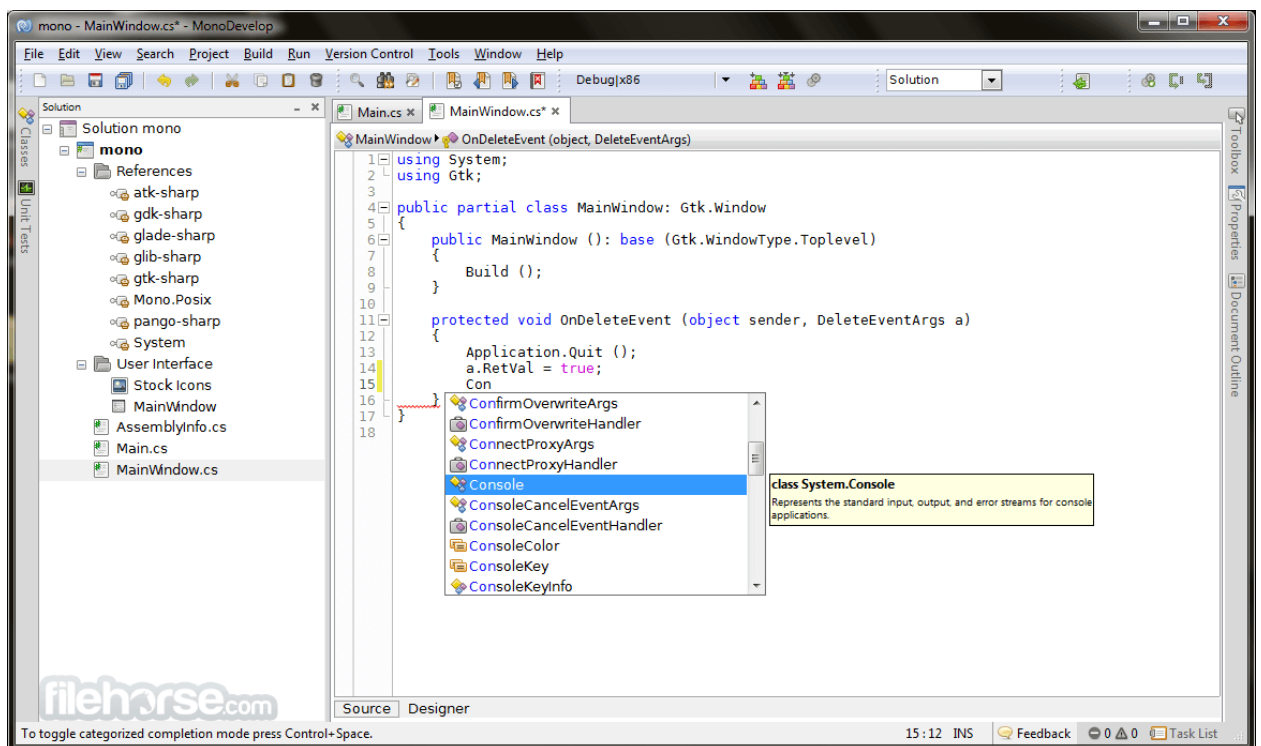


Рисунок 2.9 – Інтерфейс IDE MonoDeveloper

2.2.1 Visual Studio

Це багатofункціональна програма, яка підтримує багато аспектів розробки програмного забезпечення. Інтегроване середовище розробки Visual Studio – це стартовий майданчик для написання, налагодження та складання коду, а також подальшої публікації програм. На додаток до стандартного редактора та відладчика, які є в більшості IDE, Visual Studio включає

компілятор, автодоповнення коду, графічний конструктор і багато інших функцій, які покращують процес розробки.

Переваги Visual Studio:

- Гнучкість;
- Виробничість;
- Сумісна робота;
- Не дуже великий поріг входу;
- Безпека коду та швидкодія;

Доволі багато досвідних розробників користується саме Visual Studio. Саме це середовище розробки дозволяє зручно та швидко написати програму. Зручний та гнучкий інтерфейс, можливість підключити додаткові бібліотеки, додати систему контролю версій приваблює багатьох розробників. Visual Studio побудована в архітектурі, що підтримує можливість використання доповнень - плагінів від сторонніх розробників, що дає змогу розширювати можливості середовища розробки. Деякі з популярних доповнень:

- DevPartner Studio;
- Visual Assist;
- ReSharper;
- IncrediBuild;
- Workspace Whiz;
- Viva64;

Також Visual Studio має три версії, Community, Professional та Enterprise. Версія Visual Studio Community є абсолютно безкоштовною для учнів, студентів та розробників програм з відкритим програмним кодом. Версії Professional та Enterprise є платними. Відрізняються вони від безкоштовної Community версії тим, що мають кращий функціонал кросплатформеної розробки, інструменти тестування, розширене налагодження та діагностику.

Редактор вихідного коду Visual Studio за допомогою технології IntelliSense дозволяє дуже зручно відредагувати вихідний код і при цьому не втратити нитку змін.

Найпотужніший налагоджувач коду може працювати як з машинним, так і вихідним кодами. Також Visual Studio надає безліч інших інструментів для роботи: багатфункціональний редактор форм для створення графічного інтерфейсу, потужний веб-редактор, інструмент дизайну класів і інструмент дизайну схем баз даних.

2.3 Мова програмування C#

C# - проста, потужна, статично типізована, об'єктно орієнтована мова програмування від компанії Microsoft. C# входить до сімейства мов програмування C, синтаксис мови здається знайомим програмістам, які працювали з C, C++, Java та JavaScript. Перша версія мови C# була створена в 1998-2001 роках, групою інженерів Microsoft під керівництвом Андреса Хейлсберга і Скотта Вільтаумота як основна мова програмування платформи Microsoft .Net. C# увібрав у собі найкращі властивості мов C, C++, Modula, Object Pascal, спираючись на досвід їх практичного використання.

.Net дозволяє розробляти програмне забезпечення під різні ОС: Windows, MacOS, Linux, Android, iOS. Mono – реалізація .Net для Linux. Серед розробки: Visual Studio, Visual Studio Code, Unity3D, Xamarin, SharpDevelop, MonoDevelop покривають всі операційні системи та технології, і дозволяють робити мобільні, настільні (desktop), Web, вбудовані програми та сервіси.

- Особливості мови C#:
- Портативність;
- Типи даних;
- Мета програмування;
- Методи та функції;

Властивості;

C# розроблялась як мова програмування прикладного рівня для CLR (Common Language Runtime) і тому вона залежить, перш за все, від можливостей самої CLR. Це стосується, перш за все, системи типів C#. Присутність або відсутність тих або інших виразних особливостей мови диктується тим, чи може конкретна мовна особливість бути трансльована у відповідні конструкції CLR. CLR надає C#, як і всім іншим .NET-орієнтованим мовам, багато можливостей, яких позбавлені «класичні» мови програмування. Наприклад, збірка сміття не реалізована в самому C#, а проводиться CLR для програм, написаних на C# точно так, як і це робиться для програм на VB.NET, J# тощо.

Синтаксис мови програмування C# близький до C++. Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML.

C# у порівнянні з Java легше взаємодіє, з кодом програм, написаних на інших мовах. І саме на C# часто пишуться розширення для інших мов програмування, які використовуються у якості прошарку між бібліотекою C# і мовою, можливості якої під конкретні цілі планується розширювати.

Галузі використання мови програмування C#:

Web Development;

Client Application;

Game Development;

Internet of Things;

Enterprise;

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Основний функціонал

До основного об'єкту «Player», яким буде керувати гравець, підключені наступні компоненти:

- Скрипти
- Collider
- Rigidbody
- Transform
- Animator

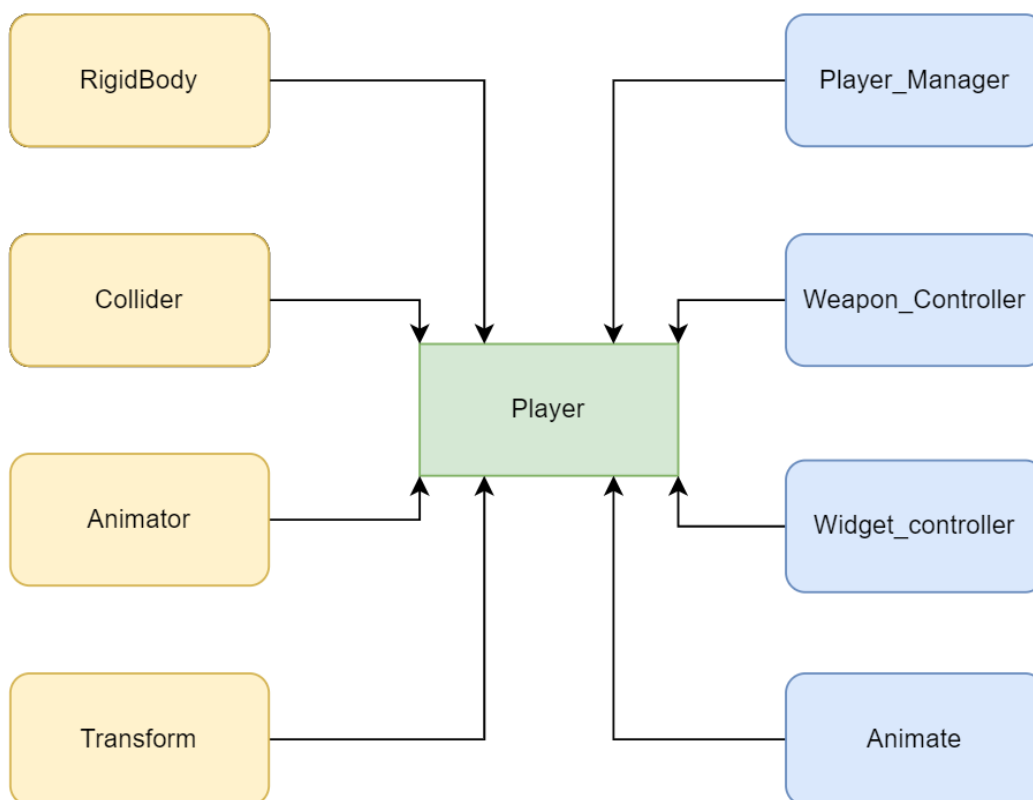


Рисунок 3.1 – Компоненти об'єкту «Player»

Rigidbody — це компоненти, які дозволяють об'єкту реагувати на фізику в реальному часі. Сюди входять реакції на сили і гравітацію, масу, опір і імпульс.

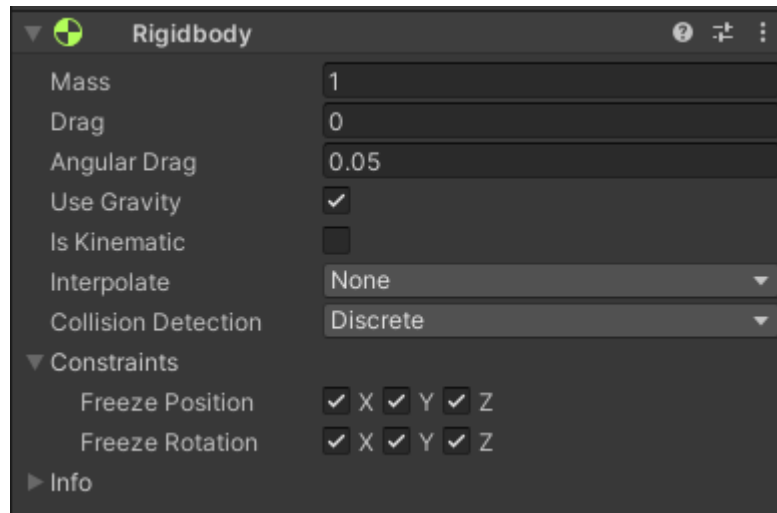


Рисунок 3.2 – Властивості Rigidbody

Collider це область, яка може взаємодіяти з іншими об'єктами фізично або викликати певні дії. Існує три види коллайдерів – Box Collider, Capsule Collider та Sphere Collider. Це компоненти, вбудовані в Unity, які забезпечують виявлення зіткнень з використанням різних «обмежуючих рамок».

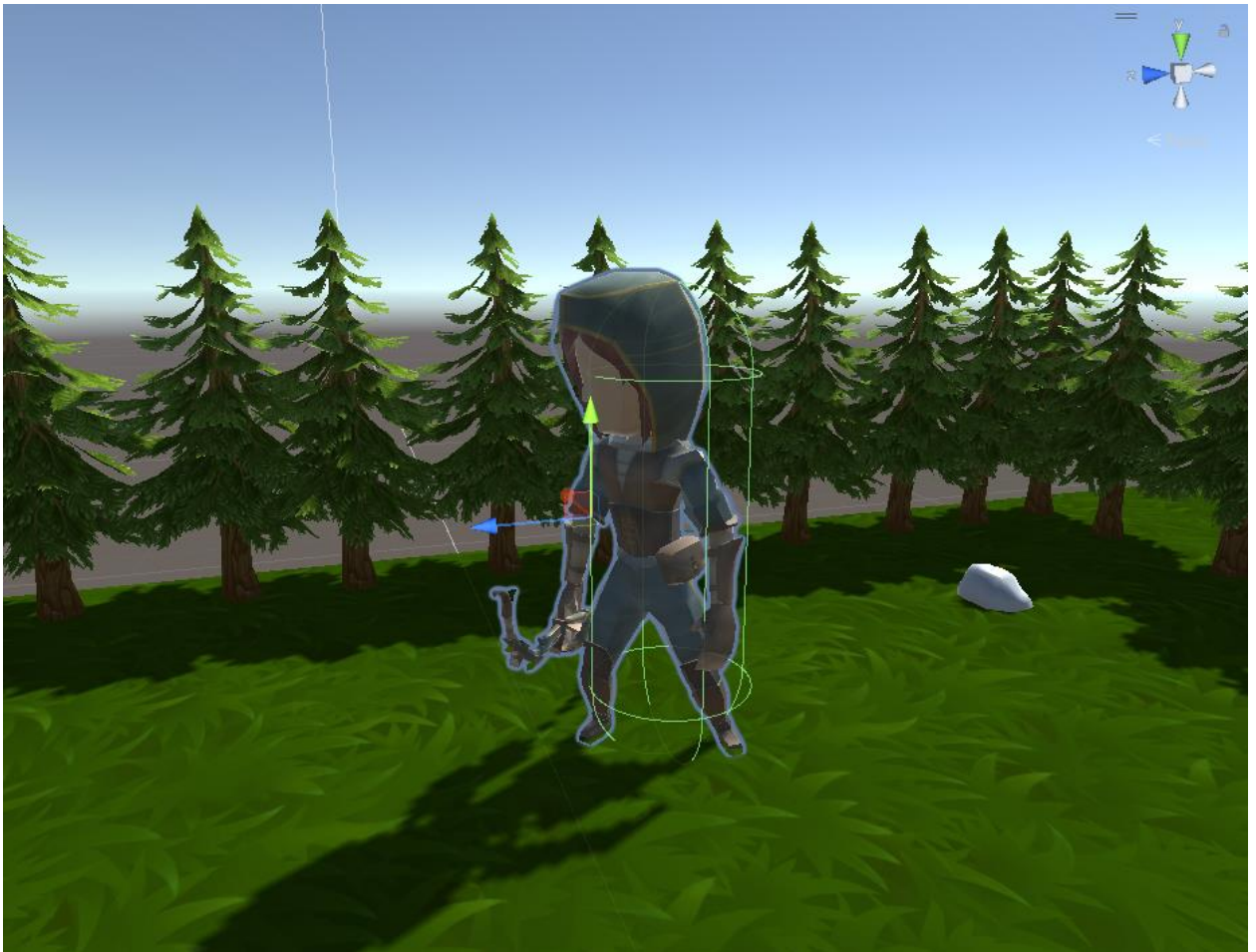


Рисунок 3.3 - Компонент Capsule Collider на об'єкті «Player»

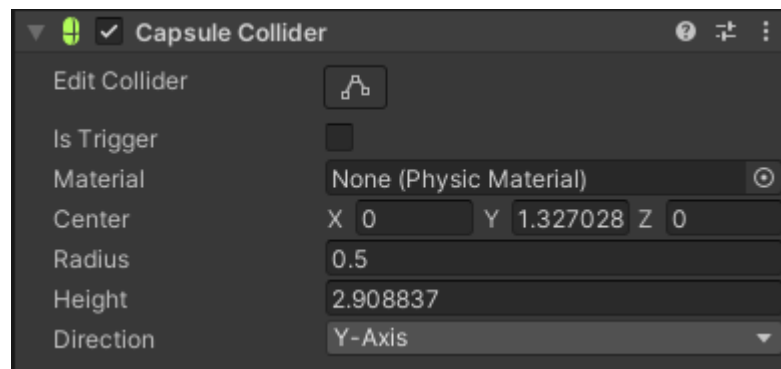


Рисунок 3.3 – Властивості Capsule Collider

Animator — це ресурс Unity, який керує логікою анімованого GameObject. У контролері Animator є стани та машини субстанів, які пов'язані між собою за допомогою переходів. Стани — це представлення анімаційних кліпів у Animator.

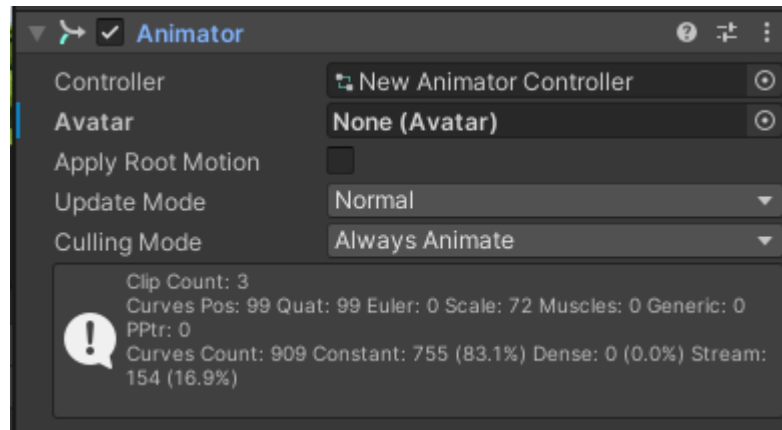


Рисунок 3.4 – Властивості Animator

У Unity компонент Transform має три видимі властивості — положення, поворот і масштаб. Кожна з цих властивостей має три значення для трьох осей. Transform використовується для визначення положення, обертання та масштабу кожного об'єкта в сцені. Кожен ігровий об'єкт має трансформацію.

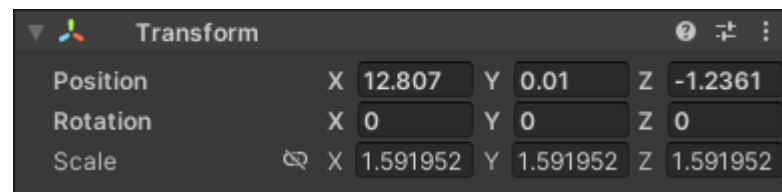


Рисунок 3.5 – Властивості Transform

Скрипт PlayerManager необхідний для того, щоб налаштувати кількість здоров'я в гравця та здатність об'єкту Player отримувати пошкодження.

```
using UnityEngine;

Скрипт Unity (ссылки на ресурсы: 4) | Ссылки: 0
public class Player_Manager : MonoBehaviour
{
    [SerializeField] private float HP;
    [SerializeField] private GameObject loseUI;

    Сообщение Unity | Ссылки: 0
    private void OnCollisionEnter(Collision collision)
    {
        if (collision.gameObject.CompareTag("Enemy"))
        {
            collision.gameObject.GetComponent<Enemy_Component>().PlayAttackAnimation();
            HP -= collision.gameObject.GetComponent<Enemy_Component>().damage;
            Debug.Log(collision.gameObject.GetComponent<Enemy_Component>().damage);

            if (HP <= 0)
            {
                loseUI.SetActive(true);
                Debug.Log("Lose");
                Destroy(gameObject);
            }
        }
    }
}
```

Рисунок 3.6 – Демонстрація скрипту «Player Manager»

За допомогою скрипта Weapon Controller можна налаштувати швидкість снаряду, який запускає гравець та пошкодження, які наносить сам снаряд.

```
using UnityEngine;

Скрипт Unity (ссылки на ресурсы: 4) | Ссылки: 2
public class Weapon_Component : MonoBehaviour
{
    ProjectileSystem m_projectileSystem;

    [SerializeField] private Transform firepoint;
    [SerializeField] float m_projectileSpeed = 3;
    [SerializeField] float damage = 0;

    Сообщение Unity | Ссылки: 0
    private void Start()
    {
        m_projectileSystem = ProjectileSystem.GetInstance();
    }

    ссылка: 1
    public void FireWeapon(Vector3 _target)
    {
        m_projectileSystem.FireProjectile(firepoint.position, _target, m_projectileSpeed, damage);
    }
}
```

Рисунок 3.7 – Демонстрація скрипту «Weapon Controller»

У скрипті Widget Controller основні функції для камери, пересування головного героя, та знаходження місцеположення ворогів на рівні.

```
using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

© Скрипт Unity (ссылка на ресурсы: 3) | Ссылка: 7
public class Widget_Controller : Singleton<Widget_Controller>
{
    float m_maxScale = 100f;

    Vector3 m_latestDirection;
    Vector2 m_touchStartPos = Vector2.zero;
    Vector2 m_touchEndPos = Vector2.zero;

    Camera m_mainCamera;

    © Сообщение Unity | Ссылка: 0
    void Start()
    {
        m_mainCamera = GetComponent<Camera>();
    }

    Ссылка: 4
    internal Vector3 GetDirection() { return m_latestDirection; }

    © Сообщение Unity | Ссылка: 0
    void Update()
    {
        if (Input.touchCount != 0)
        {
            Touch touch = Input.GetTouch(0);
            switch (touch.phase)
            {
                case TouchPhase.Began:
                    m_touchStartPos = touch.position;
                    break;
                case TouchPhase.Stationary:
                case TouchPhase.Moved:
                    m_touchEndPos = touch.position;
                    break;
                case TouchPhase.Canceled:
                case TouchPhase.Ended:
                    m_touchStartPos = Vector3.zero;
                    break;
            }
        }
    }
}
```

Рисунок 3.8 – Демонстрація скрипту «Widget Controller»

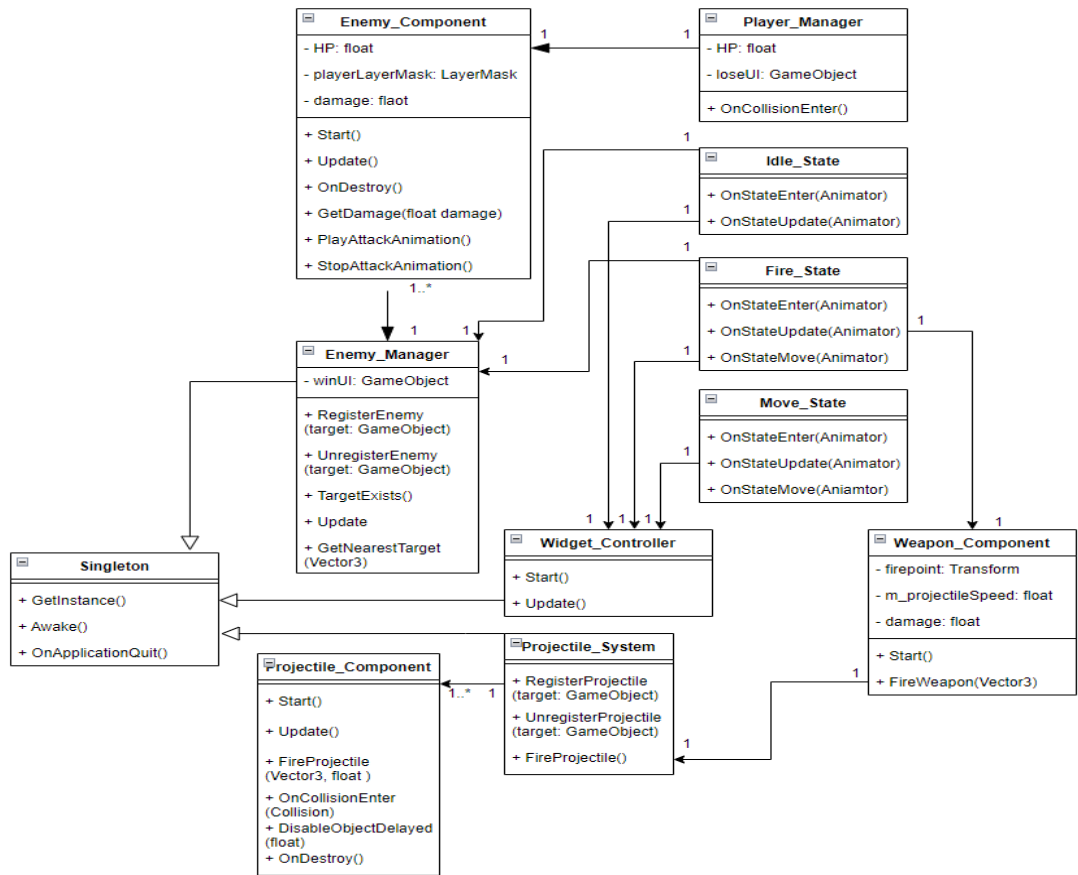


Рисунок 3.9 – Діаграма класів

3.2 Меню гри

Інтерфейс – форма, метод організації взаємодії між окремими системами; сукупність засобів та правил, що забезпечують взаємодію людини та програмного забезпечення.

Типові елементи інтерфейсу:

- Кнопка;
- Мітка;
- Вкладка;
- Список;
- Смуга прокрутки;
- Спливна підказка;
- Вікно;

- Панель;

Інтерфейси потрібні, щоб дві або більше систем могли взаємодіяти між собою. І з точки зору характеру співзалежних систем інтерфейси можна поділити на такі типи:

- Людиноорієнтовані;
- Машинні;
- Програмні;
- Графічні;

Людиноорієнтовані інтерфейси — це інтерфейси, призначені для взаємодії людини з якою-небудь іншою системою. До них відносяться всі ті інтерфейси, про які ми говорили вище: кермо і педалі в машині, дверна ручка, кнопки ліфта, екран банкомату.

Машинні інтерфейси допомагають взаємодіяти між собою машинам і механізмам. Наприклад, велосипедний ланцюг — це інтерфейс взаємодії педалей і колеса.

Бувають ще програмні інтерфейси. З їхньою допомогою програми взаємодіють між собою. Їх часто називають API — від англійського *application programming interface* (програмний інтерфейс застосунку). Наприклад, через API платформа Креативної Практики отримує дані від платіжної системи, коли ви оплачуєте курс.

В сучасних гаджетах і електронних пристроях ми взаємодіємо як правило саме з графічними інтерфейсами. В них не потрібно писати команди та розумітись у тому, як влаштовано комп'ютер з середини. Ми сприймаємо програму як набір зрозумілих для нас елементів — полів вводу, кнопок, перемикачів, іконок тощо. Тобто графічний інтерфейс створює знайомий візуальний образ, і нам відразу стає зрозуміло, як керувати цією системою.



Рисунок 3.10 – Головне меню гри «Brave Archer»



Рисунок 3.11 – Меню паузы

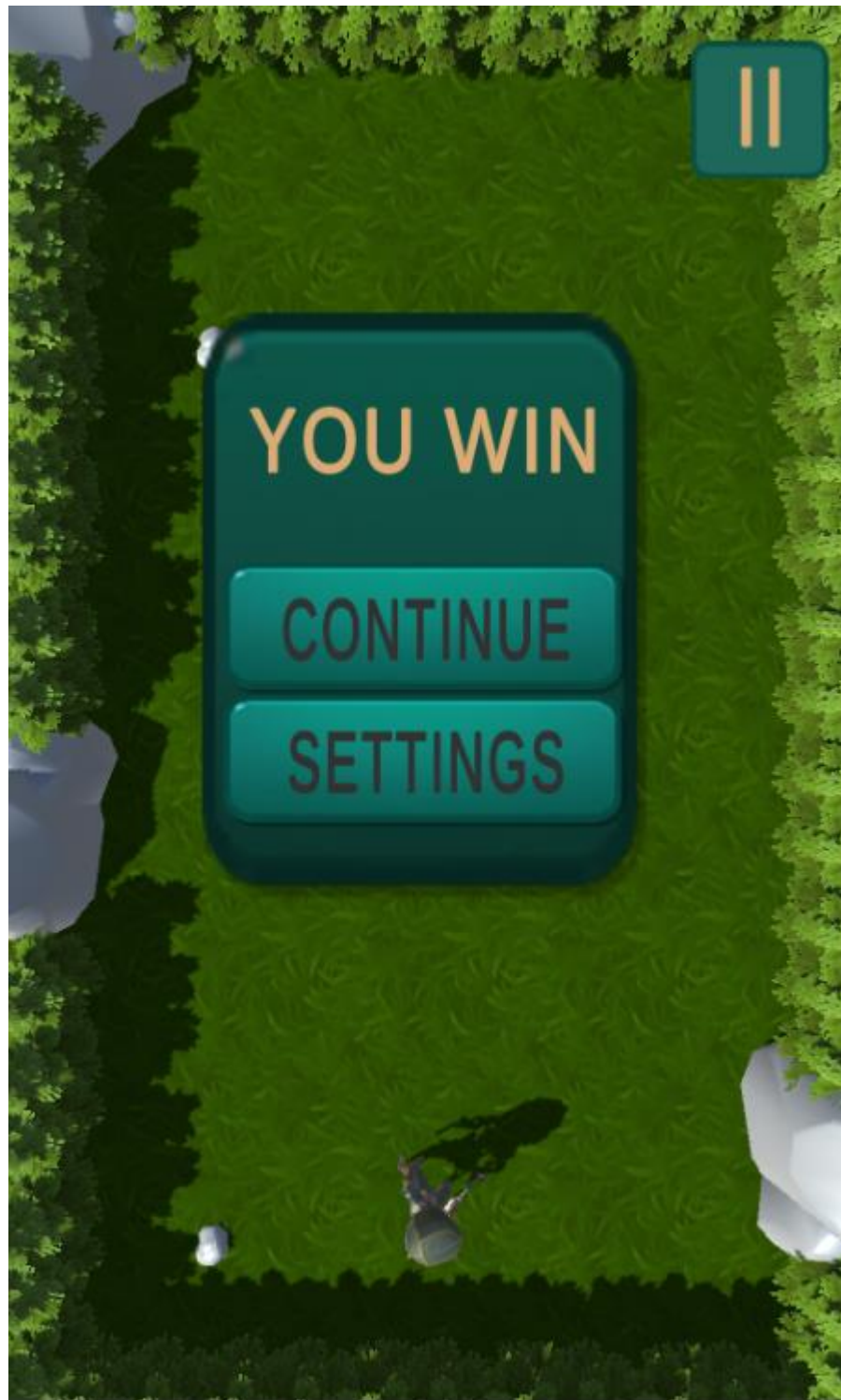


Рисунок 3.12 – Меню гри при перемозі

3.3 Тестування програмного забезпечення

Після проведення тестування, отримано такі результати:

- Меню гри є зрозумілим;

- Керування просте та зрозуміле;
- Гра може запуситися на не потужному пристрої;
- Гру можливо пройти;
- Виконується логіка рівня;;
- Ніяких помилок не виявлено;

ВИСНОВКИ

Під час виконання дипломної роботи було створено гру «Brave Archer» у жанрі Action-RPG на платформу Android з використанням ігрового двигуна Unity.

Для виконання дипломної роботи було обрано ігровий двигун Unity. Цей програмний продукт є у вільному доступі, зручний, має простий поріг входу та увесь необхідний функціонал для виконання проекту.

Для написання функціоналу для дипломної роботи була обрана мова програмування C#. Ця мова широко використовується для написання ігор на ігровому двигуні Unity. C# має великий функціонал, який добре підходить для виконання дипломної роботи.

Також було обрано середовище розробки Visual Studio. Це середовище добре підходить для написання коду на мові програмування C#. Воно зручне, багатофункціональне та має низький поріг входу.

1. Досліджено явище ігор в жанрі Action-RPG. Досліджено особливості та суть ігор в жанрі Action-RPG.
2. Проведено аналіз існуючих ігор на ринку та програмних засобів розробки. Виявлено недоліки та переваги ігрових двигунів, середовищ розробки та обрана мова програмування. Аналіз існуючих ігор в жанрі Action-RPG зміг дати розуміння для створення програмного забезпечення.
3. Розроблено функціонал для головного героя та логіка для ворогів. Написанні скрипти взаємодіють між собою. Працюють правильно та без критичних помилок.
4. Розроблено функціонал пошуку ворогів на рівні. Написанні скрипти виконують свою функцію пошуку ворогів на рівні. Скрипти працюють без помилок.
5. Створено користувацький інтерфейс для гри «Brave Archer». Інтерфейс зручний та інтуїтивно зрозумілий.

Проведені тестування дали наступні результати:

- Меню гри є зрозумілим;
- Керування просте та зрозуміле;
- Гра може запуснитися на не потужному пристрої;
- Гру можливо пройти;
- Виконується логіка рівня;;
- Ніяких помилок не виявлено;

ПЕРЕЛІК ПОСИЛАНЬ

1. Tom Wijman, The Games Market and Beyond in 2021: The Year in Numbers. [Електронний ресурс] – Ресурс доступу: <https://newzoo.com/insights/articles/the-games-market-in-2021-the-year-in-numbers-esports-cloud-gaming/>
2. ІХВТ, 2021. [Електронний ресурс] – Ресурс доступу: <https://www.ixbt.com/news/2021/12/24/v-jetom-godu-vyruchka-igrovaja-industrija-dostigla-odin-rekord-po-vyruchke-1803-mlrd-dollarov-v-jetom-godu.html>
3. Mobile game development market in Ukraine, 2021. [Електронний ресурс] – Ресурс доступу: <https://it.comments.ua/ua/article/games/rinok-rozrobki-mobilnih-igor-v-ukraini-vbivati-chas-uzhe-ne-tak-veselo-yak-ranise-690462.html>
4. What is a game engine, 2018. [Електронний ресурс] – Ресурс доступу: <https://funduk.ua/uk/technoblog/gaming-raznoe/chto-takoe-igrovoy-dvizhok/>
5. Top 10 game engines, 2022. [Електронний ресурс] – Ресурс доступу: <https://ulab.sumdu.edu.ua/uk/10-najkrashhih-igrovih-rushiiv>
6. Evheniy Orlov, Best IDEs for C# Developer, 2020. [Електронний ресурс] – Ресурс доступу: <https://spb.deveducation.com/blog/luchshie-ide-dlya-c-razrabotchika/>
7. What the IDE means and how programmers use it to create web application, 2021. [Електронний ресурс] – Ресурс доступу: <https://tebapit.com/>
8. Getting Started with Unity: Colliders and UnityScript, 2019. [Електронний ресурс] – Ресурс доступу: <https://coderlessons.com/articles/veb-razrabotka-articles/nachalo-raboty-s-unity-kollaidery-i-unityscript>
9. What is the interface. Various interfaces, 2021. [Електронний ресурс] – Ресурс доступу: <https://cases.media/article/sho-take-interfeis-riznovidi-interfeisiv>

10. A tour of the C# language, 2022. [Электронный ресурс] – Ресурс доступу: <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>
11. Unity - Manual: System requirements for Unity 2020 LTS. [Электронный ресурс] – Ресурс доступу: <https://docs.unity3d.com/Manual/system-requirements.html>

Додаток А



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробка гри Brave Archer у жанрі Action-RPG на платформі Android з використанням двигуна Unity

Виконав студент 4 курсу
групи ПД-41
Ковальов Денис Дмитрович
Керівник роботи
Доктор филозофії (PhD)
Дібрівний Олесь Андрійович

Київ – 2022

АНАЛОГИ



Diablo Immortal

Blade 2

Animus Revenant

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи: вдосконалення ігрового процесу гри в жанрі Action-RPG для платформи Android

Об'єкт дослідження: ігровий процес гри «Brave Archer» на платформу Android з використанням двигуна Unity.

Предмет дослідження: жанр Action-RPG в іграх

3

ТЕХНІЧНІ ЗАВДАННЯ

- Зробити функціонал для головного героя.
- Створити логіку поведінки ворогів.
- Розробити логіку пошуку ворогів на рівні.
- Створити інтерфейс для гри.

4

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ

Операційна система Android



Ігровий двигун Unity



Середовище розробки Visual Studio

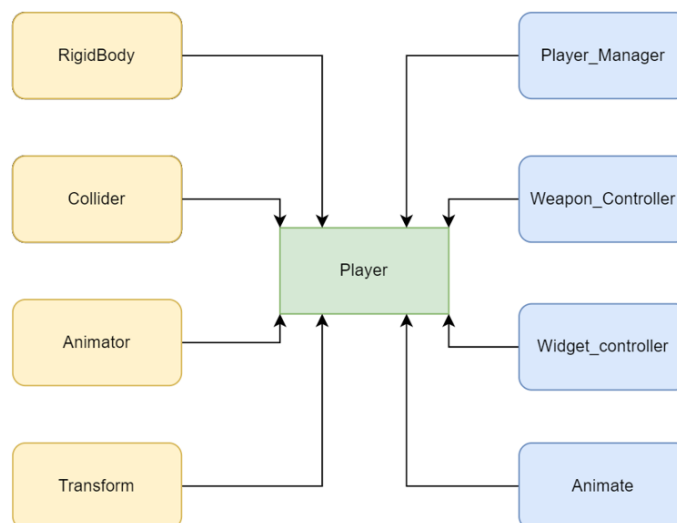


Мова програмування C#



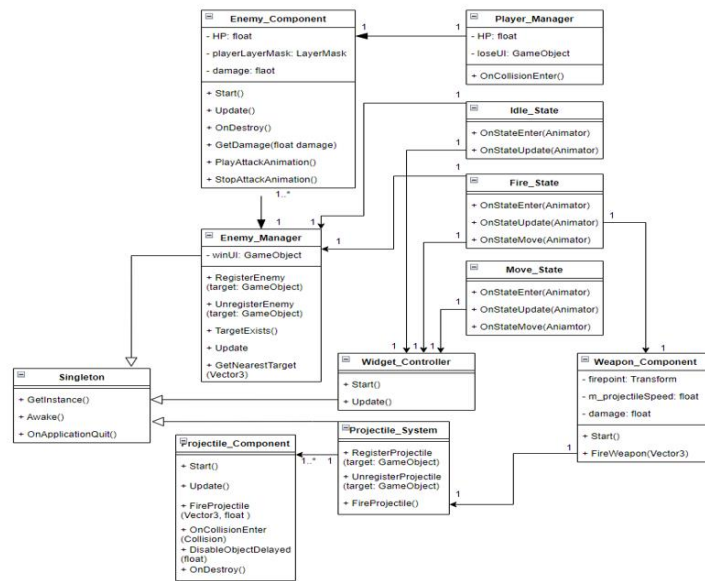
5

Компоненти об'єкту «Player»



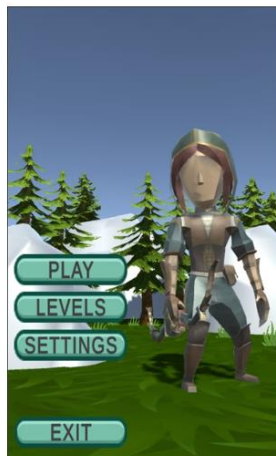
6

Діаграма класів



7

Екранні форми



Головне меню



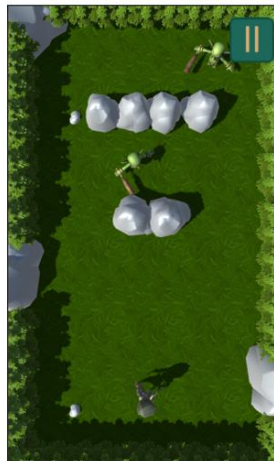
Меню паузи

8

Екранні форми



Рівень 1



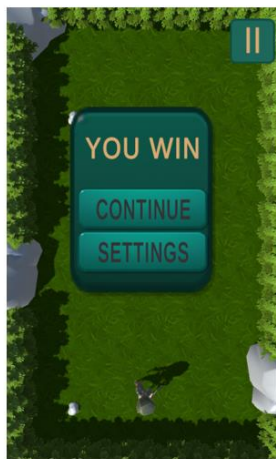
Рівень 2



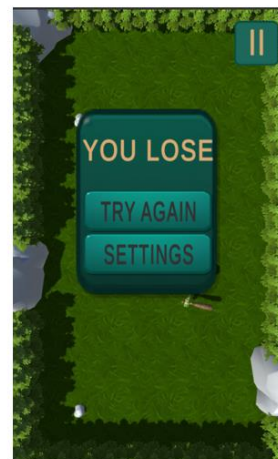
Рівень 3

9

Екранні форми



Меню при перемозі



Меню при поразці

10

Апробація

1. Ковальов Д.Д. Аналіз ринку відеоігор за 2020-2021 роки / О.А. Дібрівний, Д.Д. Ковальов // Науково-технічна конференція «Сучасні інтелектуальні інформаційні технології в науці та освіті». Збірник тез 05.04.2022, ДУТ, м.Київ – К.:ДУТ, 2022. – С. 23-24

11

ВИСНОВКИ

- Досліджено явище ігор в жанрі Action-RPG.
- Розроблено функціонал для головного героя та логіка для ворогів.
- Розроблено функціонал пошуку ворогів на рівні
- Створено користувацький інтерфейс

12

ДЯКУЮ ЗА УВАГУ!